BEYOND HOMOGENEOUS GAUSSIAN MODELING: IS CROSS-REGION GEOMETRY REPRESENTATION THE KEY TO TALKING PORTRAIT GENERATION?

Anonymous authors
Paper under double-blind review

ABSTRACT

3D Gaussian Splatting (3DGS) demonstrates strong representation capability and great potential for portrait generation. However, generated portraits still suffer from significant geometric degradation. Specifically, due to the lack of global constraints, the geometric boundary between motion regions and the static background is difficult to maintain, making the boundary prone to deformation or distortion. Furthermore, structural characteristics in the portrait can lead to motion differences across regions, resulting in structural misalignment or discontinuities. We attribute these issues to the homogeneous representation adopted by existing 3DGS-based methods, which treats regions with different motion characteristics in a unified manner and fails to capture their distinct geometric properties. To address this problem, we propose a cross-region geometric representation framework that aims to characterize the geometric relationships among different motion regions for dynamic portrait generation. The proposed framework consists of two core strategies: global motion depth representation (GMDR) and local motion multi-scale representation (LMMR). GMDR constructs a depth representation of global motion regions to describe the spatial hierarchy between the dynamic foreground and static background, thereby alleviating geometric distortions. LMMR exploits multi-scale representations of local motion regions to capture structural variations in portraits, promoting motion coordination and structural integrity across regions in continuous sequences. Experimental results demonstrate that the proposed method significantly improves the geometric fidelity of portraits in continuous dynamic sequences and outperforms existing state-of-the-art approaches.

1 INTRODUCTION

Talking portrait generation involves synthesizing facial animations from audio signals and has been widely applied in fields such as digital human interaction, content creation, and virtual education Wang et al. (2025c;a); Chatziagapi et al. (2024); Xu et al. (2026). This task involves coordinated motion among the foreground, background, and different facial regions Zhou et al. (2024); Rivero et al. (2025), with the core challenge being to maintain the geometric coherence and integrity of the portrait during dynamic generation.

In recent years, multi-view synthesis techniques, such as NeRF and 3DGS, have made significant progress in talking portrait generation. Specifically, NeRF models volume density and color distribution via an implicit radiance field, enabling fine-grained texture synthesis Li et al. (2023b); Peng et al. (2024); Li et al. (2023a; 2024a). However, under rapid pose variations, local regions are prone to boundary distortions and loss of structural details. 3DGS adopts an explicit representation based on Gaussian point clouds Wang et al. (2025b); Cho et al. (2024); Li et al. (2025); Xie et al. (2025), enabling efficient rendering and high-fidelity appearance reconstruction. Yet, in dynamic scenes, inconsistent motion patterns across regions can lead to local structural discontinuities and geometric instability. Taken together, these methods primarily rely on multi-view portrait frames to synthesize realistic videos. However, in continuous sequences, the motion coupling between the dynamic portrait and the static background, along with subtle deformations in local structures such as the mouth and hair, may cause geometric errors to accumulate over time. This degrades geometric accuracy and visual quality of generated portraits.

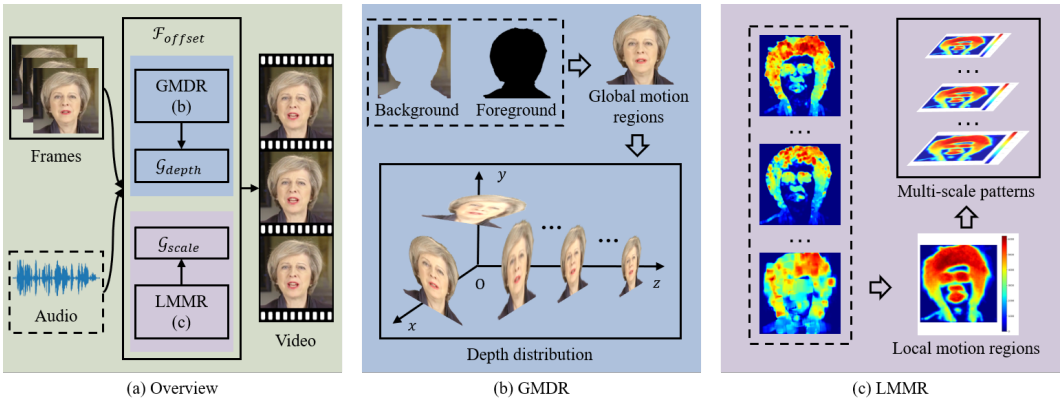
Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

We observe that in real talking videos, the foreground and background often lack clear and rich references or constraints, but their relative spatial hierarchy usually remains stable across consecutive frames Nguyen & Smeulders (2006); Zhou et al. (2020); Lu et al. (2021). Furthermore, although different structural components of the portrait exhibit significant variations during motion Wang et al. (2021); Zhao et al. (2025), the overall structure remains intact. Motivated by these observations, we propose the cross-region geometric representation to characterize the dynamic variations of different regions in continuous sequences, thereby improving the structural stability and integrity of generated portraits. Specifically, we define regions that exhibit motion relative to the static background as global motion regions, and regard the motion variations of different structures within the portrait as local motion regions. To this end, we propose two key strategies: global motion depth representation (GMDR) and local motion multi-scale representation (LMMR). GMDR constructs a depth distribution over global motion regions to represent the spatial hierarchy between the dynamic foreground and static background, and employs a fusion mechanism to achieve smooth transitions of global motion regions across frames, thereby effectively mitigating boundary deformations. LMMR captures multi-scale patterns in local motion regions to represent structural variations of the portrait, and incorporates an alignment mechanism to enhance motion coordination across regions over continuous sequences, thereby preventing structural loss and misalignment.

The main contribution of this work is a talking portrait generation method based on cross-region geometric representation. Specifically, we introduce the global motion depth representation to alleviate geometric distortions in dynamic regions, and design local motion multi-scale representation to reduce structural misalignment across regions. Extensive experiments demonstrate that our method significantly outperforms existing methods in preserving portrait geometry.

2 Overview

The overall framework is illustrated in Figure 1 (a), which includes two core strategies: global motion depth representation (GMDR) and local motion multi-scale representation (LMMR). During training, the input video frame sequence is fed into the GMDR module (Figure 1 (b)) and the LMMR module (Figure 1 (c)). These modules produce the depth distribution representation $\mathcal{G}_{depth}$ and the multi-scale patterns representation $\mathcal{G}_{scale}$, which are used to improve geometric representation consistency. During inference, as indicated by the dashed line in Figure 1 (a), an audio clip is taken as input to generate the corresponding talking video.



(a) Overview (b) GMDR (c) LMMR

Figure 1: The framework of this method consists of two modules: global motion depth representation (GMDR) and local motion multi-scale representation (LMMR). The principle of GMDR is to convert the spatial hierarchy between dynamic foregrounds and static backgrounds (dashed line in Figure 1 (b)) into the depth distribution of global motion regions (solid line in Figure 1 (b)). The core idea of LMMR is to transform portrait structural characteristics over a motion sequence (dashed line in Figure 1 (c)) into multi-scale patterns of local motion regions (solid line in Figure 1 (c)).

2.1 3DGS AND GEOMETRIC OFFSET

In the representation based on 3D Gaussian Splatting, a single Gaussian distribution is defined as:

$$G(x) = \exp\left(-\frac{1}{2}(x-\mu)^{\mathsf{T}}\Sigma^{-1}(x-\mu)\right). \tag{1}$$

where x denotes a spatial point, μ is the Gaussian center, and Σ is the covariance matrix. Each Gaussian point is parameterized by a set G, defined as:

$$G = \{c, s, r, C, \alpha\}. \tag{2}$$

where the geometric properties include the center position c, scale vector s, and rotation parameter r, while the appearance properties include the surface color C and opacity α.

To mitigate inconsistencies in portrait geometry during cross-region motion, we dynamically adjust the geometric properties. The updated parameters are defined as:

$$G_{new} = G + \Delta G. \tag{3}$$

where ΔG denotes the geometric offset for optimizing portrait geometry during cross-region motion.

The geometric offset is defined as:

$$\mathcal{F}_{offset}(\mathcal{G}_{depth}, \mathcal{G}_{scale}) : \Delta G = \{\Delta c, \Delta s, \Delta r\}. \tag{4}$$

where $\mathcal{F}_{offset}$ is a geometric offset prediction module with a branched architecture.

3 METHOD

3.1 GLOBAL MOTION DEPTH REPRESENTATION

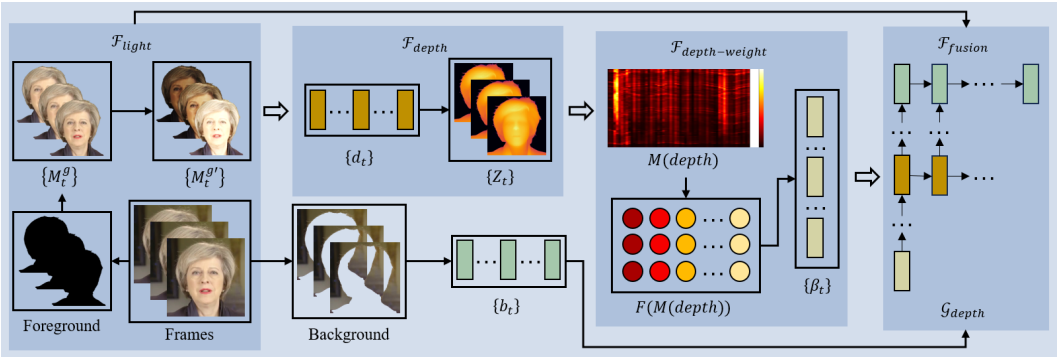


Figure 2: The process of global motion depth representation (GMDR).

The global motion depth representation (GMDR) module is shown in Figure 2. First, we perform frame differencing on the input video frames to obtain a dynamic foreground mask, based on which the global motion region M_t^g is extracted, where t denotes the index of frame I_t, $1 \leq t \leq T$, and T is the total number of frames.

It is worth noting that, since the dynamic foreground obtained via frame differencing is based solely on pixel displacement, the extracted global motion regions struggle to capture lighting changes, shadows, or other dynamic semantic information. To reduce noise in depth estimation, we introduce a lighting parameter L_t to compute the enhanced global motion region $M_t^{g'}$, as follows:

$$\mathcal{F}_{light} : M_t^g \cdot L_t \rightarrow M_t^{g'}. \tag{5}$$

where $\mathcal{F}_{light}$ employs a multi-layer convolutional encoder–decoder architecture to account for illumination and shadow effects, thereby improving the accuracy of depth estimation.

Then, the feature representation d_t of the enhanced global motion region $M_t^{g'}$ is fed into the depth inference network $\mathcal{F}_{depth}$ to obtain the corresponding depth map Z_t, as follows:

$$\mathcal{F}_{depth} : d_t \to Z_t, \quad 1 \leq t \leq T. \tag{6}$$

To ensure the continuity of depth information, we design a loss function for depth estimation:

$$\mathcal{L}_{depth} = \sum_t \left\| \left(1 - \nabla Z_t\right) \circ \left(Z_t - Z_{t-1}\right) \right\|_1. \tag{7}$$

where ∇Z_t denotes the gradient of the depth map at frame I_t, the term $1 - \nabla Z_t$ assigns lower weights to regions with large depth variations, and the depth difference $Z_t - Z_{t-1}$ characterizes the depth variation across consecutive frames.

Next, in the $\mathcal{F}_{depth-weight}$ component, we construct a depth distribution based on depth variations between consecutive frames, defined as follows:

$$|Z_t - Z_{t-1}| \to M(depth). \tag{8}$$

where $Z_t \in \mathbb{R}^{H \times W}$ denotes the depth map at frame I_t, and $M(depth)$ denotes the depth distribution saliency map computed via element-wise absolute differences.

We calculate the depth distribution weight β_t using a multilayer perceptron (MLP), as shown below:

$$\text{MLP} : F\big(M(depth)\big) \to \beta_t. \tag{9}$$

where $F\big(M(depth)\big)$ denotes the feature representation of $M(depth)$. To further regularize the depth distribution weight β_t, we define a weighted smoothness loss as follows:

$$\mathcal{L}_{depth}^{weight} = \sum_t \left\| \exp\big(\|\nabla Z_t\|\big) \cdot \nabla \beta_t \right\|_1. \tag{10}$$

where $\nabla \beta_t$ denotes the gradient of the distribution weight, and $\exp(\cdot)$ is the exponential function. This loss is used to enforce the smoothness of the weight.

Finally, in the $\mathcal{F}_{fusion}$ component, we perform the fusion computation, expressed as:

$$\mathcal{F}_{fusion} : \beta_t \cdot d_t + b_t \to \mathcal{G}_{depth}. \tag{11}$$

where $\mathcal{F}_{fusion}$ is implemented with a lightweight Transformer that concatenates the global motion region feature d_t, the static background feature b_t, and the weight β_t along the channel dimension. It outputs the depth distribution representation $\mathcal{G}_{depth}$.

We further introduce a perceptual consistency loss by feeding the fused feature representation into a lightweight perceptual encoder to measure consistency in a higher-level feature space, as follows:

$$\mathcal{L}_{fusion} = \sum_t \left\| \phi(\mathcal{G}_{depth}^t) - \phi(\mathcal{G}_{depth}^{t-1}) \right\|_1. \tag{12}$$

where ϕ is a lightweight feature extractor used to capture semantic differences Rad et al. (2019).

3.2 Local Motion Multi-scale Representation

The local motion multi-scale representation (LMMR) module is shown in Figure 3. Given a sequence of frames, we construct an attention-based local temporal window $\mathcal{W}_K$ centered at frame I_t to capture pixel-level motion variations, as follows:

$$\mathcal{W}_K : I_t \to \{H_{t-K}, \ldots, H_t, \ldots, H_{t+K}\}. \tag{13}$$

where K denotes the temporal radius of the window. For frames near sequence boundaries, we replicate the nearest valid frame to maintain a consistent window size.

We introduce a lightweight temporal encoding network $\mathcal{F}_{window}$ to model pixel-wise temporal variations and capture motion-related attributes, such as intensity, frequency, and direction-aware motion characteristics. The mapping can be formulated as:

$$\mathcal{F}_{window} : H_t \to M^l(x, y), \quad 1 \leq t \leq T. \tag{14}$$

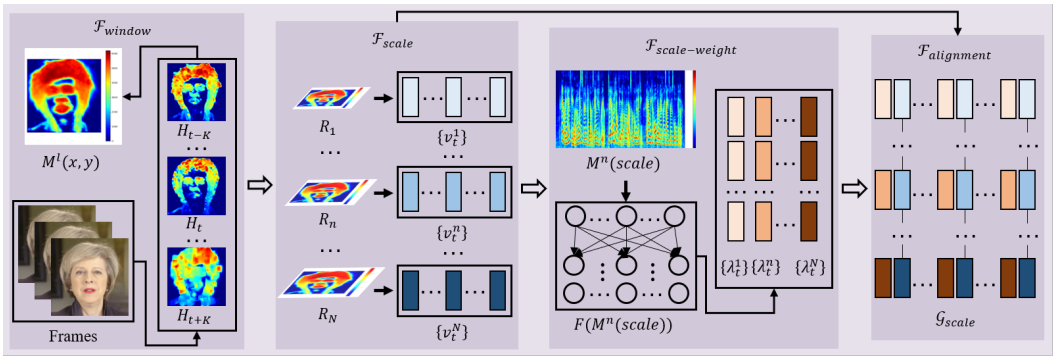


Figure 3: The process of local motion multi-scale representation (LMMR).

where $M^l(x, y)$ denotes the local motion region, reflecting the motion variations of pixel (x, y).

In the $\mathcal{F}_{scale}$ component, we perform local aggregation on $M^l(x, y)$ using different receptive field windows to capture multi-scale motion patterns, as follows:

$$\mathcal{F}_{scale} : \frac{1}{|W_h(i,j)|} \sum_{(x,y)\in W_h(i,j)} M^l(x,y) \rightarrow R_n. \tag{15}$$

where $W_h(i, j)$ denotes an $h \times h$ local window centered at (i, j). For each scale R_n, v_t^n denotes the corresponding feature sequence, where $n \in \{1, \ldots, N\}$ indexes the scale.

To ensure the accuracy and completeness of multi-scale feature information, we introduce a reconstruction loss, as follows:

$$\mathcal{L}_{scale} = \left\| \phi(f_t) - \phi(\hat{f}_t^n) \right\|_1. \tag{16}$$

where f_t denotes the target image of frame I_t, $\hat{f}_t^n$ is the image reconstructed by the decoder based on the scale feature v_t^n, and $\phi(\cdot)$ is the pre-trained encoder.

In the $\mathcal{F}_{scale\text{-}weight}$ component, the multi-scale feature sequence v_t^n is first input into a decoder and transformed into the multi-scale saliency map $M^n(scale)$, expressed as follows:

$$v_t^n \rightarrow M^n(scale), \quad 1 \le n \le N. \tag{17}$$

where $F(M^n(\text{scale}))$ denotes the feature representation of $M^n(\text{scale})$, which is used to predict the multi-scale weights λ_t^n via a fully connected layer, expressed as:

$$F(M^n(scale)) \rightarrow \lambda_t^n. \tag{18}$$

To enforce smooth transitions of multi-scale weights between adjacent scales and reduce abrupt changes, we introduce a scale smoothness loss, as follows:

$$\mathcal{L}_{scale}^{weight} = \frac{1}{T} \sum_{t=1}^{T} \frac{1}{N-1} \sum_{n=1}^{N-1} \left\| \lambda_t^{n+1} - \lambda_t^n \right\|_2. \tag{19}$$

In the $\mathcal{F}_{alignment}$ component, the multi-scale feature representations v_t^n are mapped into a unified feature space for alignment under the guidance of λ_t^n. This process produces the multi-scale patters representation $\mathcal{G}_{scale}$, as shown below:

$$\mathcal{F}_{alignment} : \sum_{n=1}^{N} \lambda_t^n \cdot v_t^n \rightarrow \mathcal{G}_{scale}. \tag{20}$$

To further improve the consistency of motion regions in generated frames, we introduce an alignment loss for region stability to penalize inconsistencies between consecutive frames, as follows:

$$\mathcal{L}_{alignment} = \sum_{t=1}^{T} \left| \|\text{Mask}_t\|_1 - \|\text{Mask}_{t-1}\|_1 \right|. \tag{21}$$

where Mask_t denotes the binary mask of the motion region in the generated image of frame I_t, with 1 denoting the motion region and 0 denoting the background.

4 EXPERIMENT

We first evaluate the generation quality through both qualitative and quantitative experiments, followed by ablation studies, and further investigate the robustness of the proposed method in complex scenarios. The Appendix includes related work (Section A), a lightweight version (Section C) of our method, and additional experiments, including a user study (Section D.1), parameter settings (Section D.2), efficiency evaluation (Section D.3), and limitations (Section D.5).

Implementation Details. Our method is implemented in PyTorch and trained on an NVIDIA RTX 5090 GPU. The training videos are sampled at 25 FPS with a resolution of 512×512, ensuring that all frames contain the speaker, and are primarily sourced from publicly available videos on YouTube. Notably, existing methods typically require 2–5 minutes of training video, whereas our method can generate high-quality results using only about 10 seconds of training video (see Section D.4 *Training with Short Videos* in Appendix).

Comparison Methods. For comparison, we select representative multi-view synthesis methods, including NeRF-based approaches (ER-NeRF Li et al. (2023a), SyncTalk Peng et al. (2024), S^3D-NeRF Li et al. (2024a)) and 3DGS-based approaches (PointTalk Xie et al. (2025), InsTaG Li et al. (2025), GaussianTalker Cho et al. (2024)).

4.1 QUALITATIVE COMPARISON

As shown in Figure 4, we compare our method with other methods and present qualitative results across consecutive frames. Ground Truth refers to the original video corresponding to the input audio. A companion video (*supplementary.mp4*) further visualizes the results.

Among NeRF-based methods, ER-NeRF introduces region-aware computation, but it does not properly account for the interaction between the global moving region and the static background. This can lead to artifacts and jitter under continuous pose changes. SyncTalk leverages large-scale audio-visual pre-training to capture head and facial motion, but it may result in non-smooth transitions and distorted facial features. S^3D-NeRF employs a multi-scale strategy to preserve identity characteristics and drive motion, yet it ignores differences between different motions, leading to blurred contours under large poses. Among 3DGS-based methods, PointTalk designed a dynamic differential encoder to accurately capture lip movements and reflect differences in movement intensity. However, it lacks global structural constraints and tends to smooth transitions while overlooking fine details. InsTaG can quickly adapt to new identities using limited data, but this comes at the expense of visual quality and temporal consistency. GaussianTalker combines spatiotemporal attention to improve lip synchronization, yet it suffers from structural inconsistencies under extreme poses due to insufficient training coverage.

In contrast, our method better preserves the spatial hierarchy between the dynamic foreground and the static background, effectively reducing deformations and distortions at motion boundaries, due to the use of global motion depth representation. Meanwhile, our method better captures structural variations across regions with the proposed local motion multi-scale representation, alleviating structural misalignment and discontinuities. As shown in Figure 4, we zoom in on the hair region and provide a comparison of the depth maps. The results demonstrate that our method produces results closer to the Ground Truth, with depth maps exhibiting clearer hierarchical structure and finer details. Overall, the proposed method can more accurately capture the hierarchical structure of motion regions across consecutive frames, significantly improving the accuracy of portrait geometry reconstruction, while achieving superior performance over existing methods in terms of visual quality and motion coherence.

4.2 QUANTITATIVE COMPARISON

To comprehensively evaluate the proposed method, we conducted quantitative tests in several aspects. For image quality, PSNR, SSIM Wang et al. (2004), and LPIPS Zhang et al. (2018) are used to measure image fidelity. For lip-sync accuracy, SyncNet confidence and LSE-C are employed to assess the synchronization between audio and lip movements Chung & Zisserman (2017); Prajwal et al. (2020). For facial motion, AUE and LMD are used to evaluate the localization errors of action units and keypoints Chen et al. (2018). For identity preservation, FID Heusel et al. (2017) and CSIM

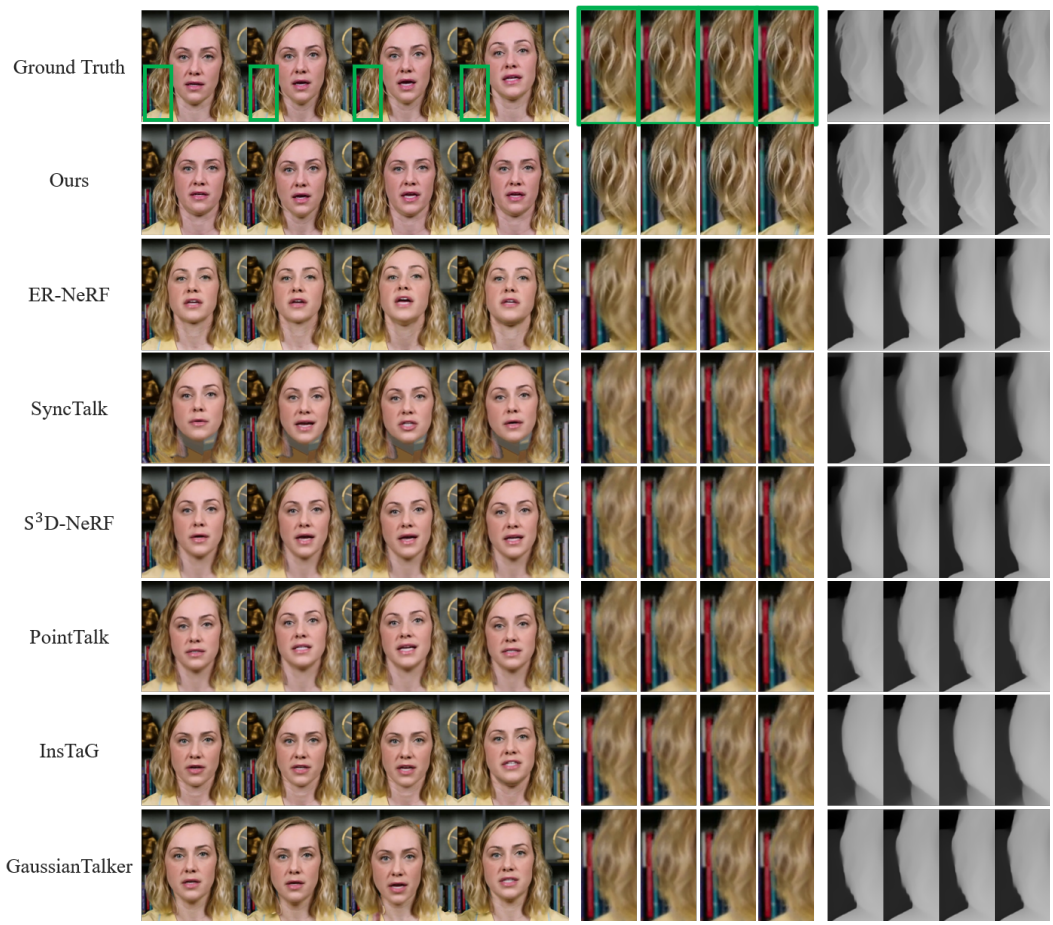


Figure 4: Comparison results. We present visualizations of consecutive frames and corresponding depth maps. As can be seen, our method outperforms other state-of-the-art methods, achieving clear visual performance and well-defined spatial hierarchy.

Huang et al. (2020) are adopted to quantify the consistency of identity. For inference speed, frames per second (FPS) Ye et al. (2023a); Chen et al. (2019) are used to compare the generation efficiency.

As shown in Table 1, NeRF-based methods such as ER-NeRF, SyncTalk, and S^{3}D-NeRF often suffer from blurred head contours and varying degrees of lip-sync errors under large head poses, and they are limited in capturing fine facial motion details. PointTalk, InsTaG, and GaussianTalker, which are based on 3DGS, can produce smoother motion but still suffer from geometric fragmentation or blurriness in subtle dynamic regions such as the mouth and hair. In contrast, our method can more accurately recover geometric structures while demonstrating advantages in image quality, lip synchronization, and detail preservation.

Table 1: Quantitative comparison.

Methods	PSNR ↑	LPIPS ↓	SyncNet ↑	LSE-C ↑	LMD ↓	CSIM ↑	FPS ↑
Ground Truth	N/A	N/A	7.647	7.524	N/A	1	N/A
ER-NeRF	30.145	0.038	6.594	6.894	4.017	0.935	27
SyncTalk	30.446	0.039	6.348	6.784	3.984	0.928	23
S^{3}D-NeRF	30.349	0.038	6.764	6.861	4.152	0.934	9.5
PointTalk	30.484	0.037	6.854	7.021	3.764	0.929	69
InsTaG	30.494	0.038	6.497	7.016	3.814	0.932	72
GaussianTalker	30.816	0.037	6.585	6.946	3.906	0.933	81
Ours	**33.324**	**0.025**	**7.045**	**7.212**	**2.775**	**0.946**	**87**

4.3 Ablation Study

To evaluate the contributions of different components to the generated results, we conducted several ablation experiments. Figure 5 presents representative comparison results from the ablation study, while Table 2 summarizes the quantitative results under different variants.

Ours w/o $\mathcal{F}_{light}$. $\mathcal{F}_{light}$ effectively distinguishes pixel dynamics induced by lighting variations, thereby enhancing the accuracy of motion region representation. Removing this component may result in flickering at boundary regions or jittering in local areas.

Ours w/o $\mathcal{F}_{depth}$. $\mathcal{F}_{depth}$ computes depth from the global motion region, ensuring spatial hierarchy between the foreground and background. Removing this component reduces the accuracy of depth estimation and may cause structural misalignment.

Ours w/o $\mathcal{F}_{depth\text{-}weight}$. $\mathcal{F}_{depth\text{-}weight}$ calculates the weight variations of the depth distribution between consecutive frames to ensure temporal smoothness. Removing this component may lead to inter-frame depth jitter, size inconsistencies, and facial artifacts.

Ours w/o $\mathcal{F}_{fusion}$. $\mathcal{F}_{fusion}$ fuses the dynamic foreground and static background to maintain a consistent spatial hierarchy. Removing this component may result in unnatural head boundaries, causing jitter or occlusion errors.

Ours w/o $\mathcal{F}_{window}$. $\mathcal{F}_{window}$ maps sequences of local changes across consecutive frames into multi-dimensional motion attributes. Removing this component may lead to blurring or jittering artifacts along foreground boundaries.

Ours w/o $\mathcal{F}_{scale}$. The $\mathcal{F}_{scale}$ maps structural characteristics to multi-scale patterns, thereby capturing motion information at different scales. Removing this component reduces the ability to extract motion features, potentially leading to errors in lip-sync.

Ours w/o $\mathcal{F}_{scale\text{-}weight}$. $\mathcal{F}_{scale\text{-}weight}$ computes the motion intensity weight at each scale to quantify the contribution of each scale. Removing this component may cause local motions to be over-smoothed or ignored.

Ours w/o $\mathcal{F}_{alignment}$. $\mathcal{F}_{alignment}$ aligns different local motion regions to maintain structural continuity. Removing this component may result in misaligned regions and broken boundaries.

Based on both visualization results and quantitative metrics, the full model (Ours) achieves results closest to the Ground Truth. In contrast, all ablated variants show noticeable degradation across multiple aspects, highlighting the effectiveness of the proposed components.

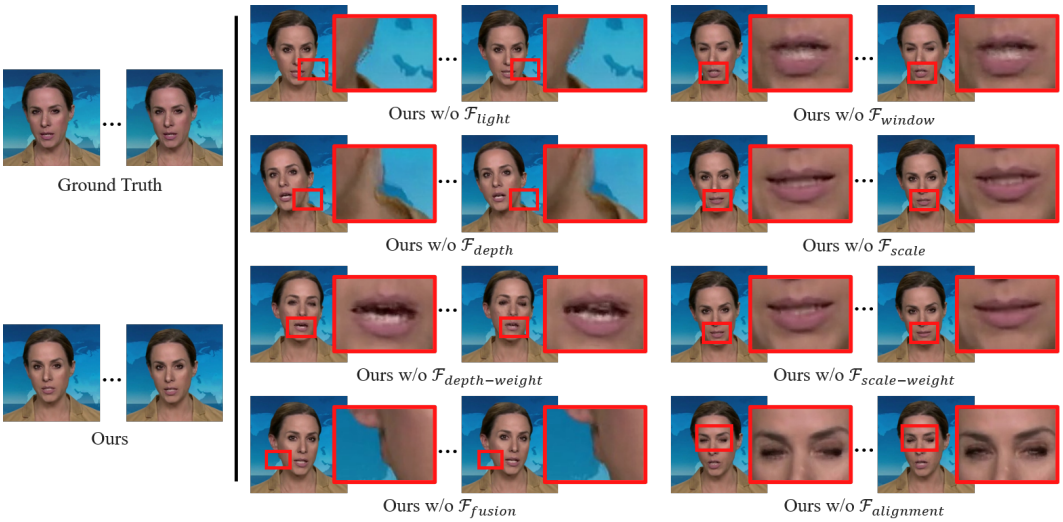


Figure 5: Qualitative ablation study. The red boxes highlight the locations of artifacts, lip sync, etc.

Table 2: Quantitative ablation study.

Methods	PSNR ↑	LPIPS ↓	SyncNet ↑	LSE-C ↑	LMD ↓	CSIM ↑
Ground Truth	N/A	N/A	7.549	7.436	N/A	1
Ours w/o $\mathcal{F}_{light}$	30.412	0.042	6.573	6.814	3.824	0.932
Ours w/o $\mathcal{F}_{depth}$	30.602	0.041	6.416	6.758	4.115	0.928
Ours w/o $\mathcal{F}_{depth-weight}$	30.185	0.040	6.525	6.705	4.069	0.926
Ours w/o $\mathcal{F}_{fusion}$	30.376	0.039	6.621	6.822	3.958	0.929
Ours w/o $\mathcal{F}_{window}$	30.489	0.035	6.518	6.872	3.824	0.932
Ours w/o $\mathcal{F}_{scale}$	30.826	0.038	6.323	6.609	3.782	0.927
Ours w/o $\mathcal{F}_{scale-weight}$	30.638	0.037	6.417	6.725	3.707	0.930
Ours w/o $\mathcal{F}_{alignment}$	30.634	0.036	6.582	6.904	3.659	0.931
Ours	**33.326**	**0.026**	**7.058**	**7.158**	**2.784**	**0.947**

4.4 Discussion

To further validate the robustness of our method in challenging scenarios, we extend the test settings to include more difficult conditions, such as wearing glasses, large head movements, and hand gestures. As shown in Figure 6, we compared our method with ER-NeRF and GaussianTalker, which are also based on multi-view synthesis. The results show that both methods exhibit noticeable finger misalignment or artifacts. In contrast, our method generates structurally complete fingers with clearer visual results. This is because our method models the spatial hierarchy between the foreground and background as global motion regions, while mapping portrait structural variations to local motion regions, thereby ensuring geometric completeness and stability in dynamic generation and enhancing robustness in complex scenarios.

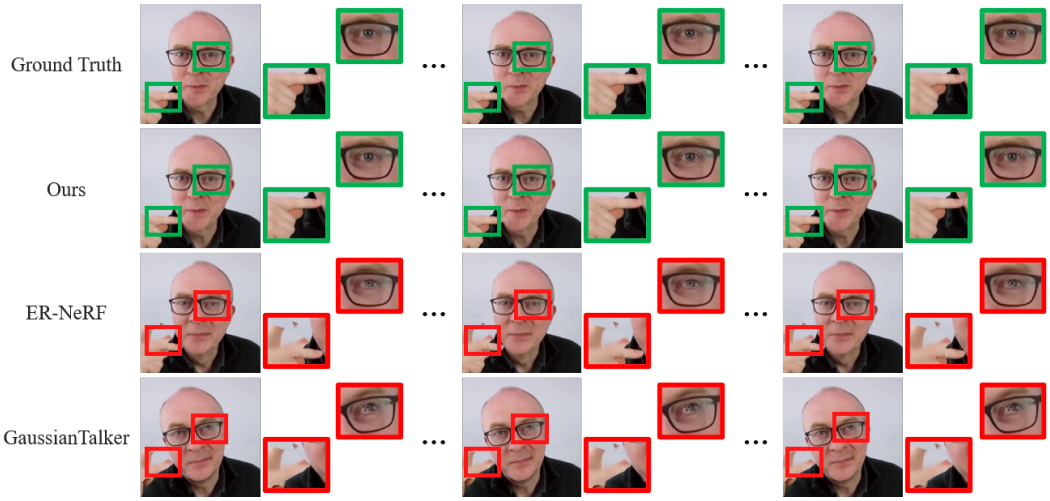


Figure 6: Robustness under complex scenarios. The red boxes indicate structural misalignment of the fingers and artifacts in the regions occluded by eyeglasses.

5 Conclusion

This paper proposes a talking portrait generation method based on cross-regional geometric representation to enhance the geometric fidelity of portraits during dynamic generation. Specifically, we propose the GMDR (global motion depth representation) to characterize the spatial hierarchy of the foreground and background, thereby mitigating geometric distortions. We further introduce the LMMR (local motion multi-scale representation) to capture portrait motion characteristics, thus alleviating structural misalignment or loss. Experimental results demonstrate that the proposed method outperforms existing approaches, achieving higher geometric accuracy and superior visual quality.

AI USE STATEMENT

We used generative AI tools solely to improve the clarity and readability of the English writing. All research ideas, methodology, experiments, results, and interpretations were developed by the authors. All AI-assisted text was reviewed by the authors, who take full responsibility for the final content.

ETHICS STATEMENT

This work does not involve human subjects, sensitive personal data, or other foreseeable ethical concerns. All data used in this study are publicly available, and the experiments were conducted in accordance with applicable research integrity and data usage guidelines.

REFERENCES

Aggelina Chatziagapi, Bindita Chaudhuri, Amit Kumar, Rakesh Ranjan, Dimitris Samaras, and Nikolaos Sarafianos. Talkinnerf: Animatable neural fields for full-body talking humans. In *Proceedings of the European Conference on Computer Vision*, pp. 148–166, 2024.

Jianchuan Chen, Jingchuan Hu, Gaige Wang, Zhonghua Jiang, Tiansong Zhou, Zhiwen Chen, and Chengfei Lv. Taoavatar: Real-time lifelike full-body talking avatars for augmented reality via 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10723–10734, 2025.

Lele Chen, Zhiheng Li, Ross K Maddox, Zhiyao Duan, and Chenliang Xu. Lip movements generation at a glance. In *Proceedings of the European Conference on Computer Vision*, pp. 520–535, 2018.

Lele Chen, Ross K Maddox, Zhiyao Duan, and Chenliang Xu. Hierarchical cross-modal talking face generation with dynamic pixel-wise loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7832–7841, 2019.

Kyusun Cho, Joungbin Lee, Heeji Yoon, Yeobin Hong, Jaehoon Ko, Sangjun Ahn, and Seungryong Kim. Gaussiantalker: Real-time talking head synthesis with 3d gaussian splatting. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 10985–10994, 2024.

Joon Son Chung and Andrew Zisserman. Out of time: Automated lip sync in the wild. In *Computer Vision–ACCV 2016 Workshops: ACCV 2016 International Workshops, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part II 13*, pp. 251–263, 2017.

Helisa Dhamo, Yinyu Nie, Arthur Moreau, Jifei Song, Richard Shaw, Yiren Zhou, and Eduardo Pérez-Pellitero. Headgas: Real-time animatable head avatars via 3d gaussian splatting. In *Proceedings of the European Conference on Computer Vision*, pp. 459–476, 2024.

Biao Dong and Lei Zhang. Talking head generation via viewpoint and lighting simulation based on global representation. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 10258–10267, 2025.

Guy Gafni, Justus Thies, Michael Zollhofer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8649–8658, 2021.

Shengjie Gong, Haojie Li, Jiapeng Tang, Dongming Hu, Shuangping Huang, Hao Chen, Tianshui Chen, and Zhuoman Liu. Monocular and generalizable gaussian talking head animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5523–5534, 2025.

Yudong Guo, Keyu Chen, Sen Liang, Yong-Jin Liu, Hujun Bao, and Juyong Zhang. Ad-nerf: Audio driven neural radiance fields for talking head synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5784–5794, 2021.

Qianyun He, Xinya Ji, Yicheng Gong, Yuanxun Lu, Zhengyu Diao, Linjia Huang, Yao Yao, Siyu Zhu, Zhan Ma, Songcen Xu, et al. Emotalk3d: High-fidelity free-view synthesis of emotional 3d talking head. In *Proceedings of the European Conference on Computer Vision*, pp. 55–72, 2024.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems*, 30, 2017.

Yuge Huang, Yuhan Wang, Ying Tai, Xiaoming Liu, Pengcheng Shen, Shaoxin Li, Jilin Li, and Feiyue Huang. Curricularface: Adaptive curriculum learning loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5901–5910, 2020.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):139–1, 2023.

Gihoon Kim, Kwanggyoon Seo, Sihun Cha, and Junyong Noh. Nerffacespeech: One-shot audio-driven 3d talking head synthesis via generative prior. *arXiv preprint arXiv:2405.05749*, 2024.

Dongze Li, Kang Zhao, Wei Wang, Yifeng Ma, Bo Peng, Yingya Zhang, and Jing Dong. S^3d-nerf: Single-shot speech-driven neural radiance field for high fidelity talking head synthesis. In *Proceedings of the European Conference on Computer Vision*, pp. 365–382, 2024a.

Dongze Li, Kang Zhao, Wei Wang, Bo Peng, Yingya Zhang, Jing Dong, and Tieniu Tan. Ae-nerf: Audio enhanced neural radiance field for few shot talking head synthesis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 3037–3045, 2024b.

Jiahe Li, Jiawei Zhang, Xiao Bai, Jun Zhou, and Lin Gu. Efficient region-aware neural radiance fields for high-fidelity talking portrait synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7568–7578, 2023a.

Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. Talkinggaussian: Structure-persistent 3d talking head synthesis via gaussian splatting. In *Proceedings of the European Conference on Computer Vision*, pp. 127–145, 2024c.

Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Jun Zhou, and Lin Gu. Instag: Learning personalized 3d talking head from few-second video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10690–10700, 2025.

Weichuang Li, Longhao Zhang, Dong Wang, Bin Zhao, Zhigang Wang, Mulin Chen, Bang Zhang, Zhongjian Wang, Liefeng Bo, and Xuelong Li. One-shot high-fidelity talking-head synthesis with deformable neural radiance field. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17969–17978, 2023b.

Chang Liu, Tianjiao Jing, Chengcheng Ma, Xuanqi Zhou, Zhengxuan Lian, Qin Jin, Hongliang Yuan, and Shi-Sheng Huang. Emodifftalk: Emotion-aware diffusion for editable 3d gaussian talking head. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18063–18073, 2026.

Xian Liu, Yinghao Xu, Qianyi Wu, Hang Zhou, Wayne Wu, and Bolei Zhou. Semantic-aware implicit neural audio-driven video portrait generation. In *Proceedings of the European Conference on Computer Vision*, pp. 106–125, 2022.

Yuanxun Lu, Jinxiang Chai, and Xun Cao. Live speech portraits: Real-time photorealistic talking-head animation. *ACM Transactions on Graphics*, 40(6):1–17, 2021.

Zhiyuan Ma, Xiangyu Zhu, Guo-Jun Qi, Zhen Lei, and Lei Zhang. Otavatar : One-shot talking face avatar with controllable tri-plane rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16901–16910, 2023.

Hieu T Nguyen and Arnold WM Smeulders. Robust tracking using foreground-background texture discrimination. *International Journal of Computer Vision*, 69(3):277–293, 2006.

Ziqiao Peng, Wentao Hu, Yue Shi, Xiangyu Zhu, Xiaomei Zhang, Hao Zhao, Jun He, Hongyan Liu, and Zhaoxin Fan. Synctalk: The devil is in the synchronization for talking head synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 666–676, 2024.

KR Prajwal, Rudrabha Mukhopadhyay, Vinay P Namboodiri, and CV 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*, pp. 484–492, 2020.

Shenhan Qian, Tobias Kirschstein, Liam Schoneveld, Davide Davoli, Simon Giebenhain, and Matthias Nießner. Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20299–20309, 2024.

Mohammad Saeed Rad, Behzad Bozorgtabar, Urs-Viktor Marti, Max Basler, Hazim Kemal Ekenel, and Jean-Philippe Thiran. Srobb: Targeted perceptual loss for single image super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2710–2719, 2019.

Alfredo Rivero, ShahRukh Athar, Zhixin Shu, and Dimitris Samaras. Rig3dgs: Creating controllable portraits from casual monocular videos. In *2025 International Conference on 3D Vision*, pp. 1541–1550. IEEE, 2025.

Zhijing Shao, Zhaolong Wang, Zhuang Li, Duotun Wang, Xiangru Lin, Yu Zhang, Mingming Fan, and Zeyu Wang. Splattingavatar: Realistic real-time human avatars with mesh-embedded gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1606–1616, 2024.

Shuai Shen, Wanhua Li, Zheng Zhu, Yueqi Duan, Jie Zhou, and Jiwen Lu. Learning dynamic facial radiance fields for few-shot talking head synthesis. In *Proceedings of the European Conference on Computer Vision*, pp. 666–682, 2022.

Haotian Wang, Yuzhe Weng, Yueyan Li, Zilu Guo, Jun Du, Shutong Niu, Jiefeng Ma, Shan He, Xiaoyan Wu, Qiming Hu, et al. Emotivetalk: Expressive talking head generation through audio information decoupling and emotional video diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26212–26221, 2025a.

Jie Wang, Jiu-Cheng Xie, Xianyan Li, Feng Xu, Chi-Man Pun, and Hao Gao. Gaussianhead: High-fidelity head avatars with learnable gaussian derivation. *IEEE Transactions on Visualization and Computer Graphics*, 2025b.

Suzhen Wang, Lincheng Li, Yu Ding, Changjie Fan, and Xin Yu. Audio2head: Audio-driven one-shot talking-head generation with natural head motion. *arXiv preprint arXiv:2107.09293*, 2021.

Yu Wang, Yunfei Liu, Fa-Ting Hong, Meng Cao, Lijian Lin, and Yu Li. Anytalk: Multi-modal driven multi-domain talking head generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 8105–8113, 2025c.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004.

Yifan Xie, Tao Feng, Xin Zhang, Xiangyang Luo, Zixuan Guo, Weijiang Yu, Heng Chang, Fei Ma, and Fei Richard Yu. Pointtalk: Audio-driven dynamic lip point cloud for 3d gaussian-based talking head synthesis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 8753–8761, 2025.

Haolan Xu, Keli Cheng, Lei Wang, Ning Bi, and Xiaoming Liu. Emotag: Emotion-aware talking head synthesis on gaussian splatting with few-shot personalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10921–10931, 2026.

Zhenhui Ye, Jinzheng He, Ziyue Jiang, Rongjie Huang, Jiawei Huang, Jinglin Liu, Yi Ren, Xiang Yin, Zejun Ma, and Zhou Zhao. Geneface++: Generalized and stable real-time audio-driven 3d talking face generation. *arXiv preprint arXiv:2305.00787*, 2023a.

Zhenhui Ye, Ziyue Jiang, Yi Ren, Jinglin Liu, Jinzheng He, and Zhou Zhao. Geneface: Generalized and high-fidelity audio-driven 3d talking face synthesis. *arXiv preprint arXiv:2301.13430*, 2023b.

Hongyun Yu, Zhan Qu, Qihang Yu, Jianchuan Chen, Zhonghua Jiang, Zhiwen Chen, Shengyu Zhang, Jimin Xu, Fei Wu, Chengfei Lv, et al. Gaussiantalker: Speaker-specific talking head synthesis via 3d gaussian splatting. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 3548–3557, 2024.

Wangbo Yu, Yanbo Fan, Yong Zhang, Xuan Wang, Fei Yin, Yunpeng Bai, Yan-Pei Cao, Ying Shan, Yang Wu, Zhongqian Sun, et al. Nofa: Nerf-based one-shot facial avatar reconstruction. In *ACM SIGGRAPH 2023 Conference Proceedings*, pp. 1–12, 2023.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.

Shuling Zhao, Fa-Ting Hong, Xiaoshui Huang, and Dan Xu. Synergizing motion and appearance: Multi-scale compensatory codebooks for talking head video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26232–26241, 2025.

Yufeng Zheng, Wang Yifan, Gordon Wetzstein, Michael J Black, and Otmar Hilliges. Pointavatar: Deformable point-based head avatars from videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21057–21067, 2023.

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

Zhenglin Zhou, Fan Ma, Hehe Fan, Zongxin Yang, and Yi Yang. Headstudio: Text to animatable head avatars with 3d gaussian splatting. In *Proceedings of the European Conference on Computer Vision*, pp. 145–163, 2024.

A Related Work

A.1 NeRF-Based Talking Portrait Generation

NeRF-based talking portrait methods extend neural radiance fields to facial modeling tasks Gafni et al. (2021); Kim et al. (2024); Yu et al. (2023); Ma et al. (2023), generating synchronized and realistic talking videos driven by audio input Dong & Zhang (2025); Zheng et al. (2023); Liu et al. (2022). Early approaches, such as AD-NeRF Guo et al. (2021), directly generate high-fidelity talking videos from audio. However, they often suffer from lip-sync errors and image blurring when encountering identity changes or large head motions. Subsequently, DFRF Shen et al. (2022) achieved faster training and inference with limited data, but its small model size led to weak generalization. The GeneFace series Ye et al. (2023a;b), trained on large-scale datasets, introduced head–torso joint modeling, which effectively alleviated structural separation issues. However, its rendering granularity remained coarse, resulting in noticeable detail loss. AE-NeRF Li et al. (2024b) improved identity consistency and lip synchronization through audio alignment and region-aware fusion, but its dual-NeRF design limited overall efficiency.

Overall, while recent methods have shown steady progress in rendering quality and efficiency, they still face challenges in maintaining structural stability in dynamic scenes, especially under large motions, which constrains their practical performance.

A.2 3DGS-Based Talking Portrait Generation

3DGS-based talking portrait generation typically achieves natural motion by learning the correspondence between audio, facial expressions, and head movements Kerbl et al. (2023); He et al. (2024); Yu et al. (2024), dynamically controlling the positions and properties of Gaussian points Gong et al. (2025); Li et al. (2024c); Liu et al. (2026). SplattingAvatar Shao et al. (2024) and GaussianAvatars Qian et al. (2024) enhance geometric consistency and animation controllability through explicit

meshes and local coordinate systems. However, they lack the ability to model non-rigid structures such as hair. To further improve expressive detail and real-time performance, HeadGaS Dhamo et al. (2024) introduces linearly fusible latent variables to enhance rendering quality, but relies on a facial tracker, making it vulnerable to pose errors. Meanwhile, TaoAvatar Chen et al. (2025) achieves fine control over facial expressions and body movements, but its application is still limited due to insufficient training data coverage for exaggerated movements.

In summary, for talking portrait tasks where motion regions exhibit complex structures, existing methods still struggle to maintain geometric consistency and temporal stability, which affects realism and limits their ability to generate highly natural and dynamic talking videos.

B More Details of Our Method

This appendix provides the detailed mathematical formulation of the proposed global motion depth representation (GMDR) and local motion multi-scale representation (LMMR). The derivations follow the computational flow of the main method, including dynamic foreground extraction, illumination-aware depth estimation, depth distribution weighting, global feature fusion, local temporal modeling, multi-scale aggregation, adaptive scale weighting, and motion-region alignment.

B.1 3D Gaussian Representation and Geometric Offset

For a spatial point $x \in \mathbb{R}^3$, the Gaussian function is defined by its center μ and covariance matrix Σ:

$$G(x) = \exp\left(-\frac{1}{2}(x-\mu)^{\mathsf{T}}\Sigma^{-1}(x-\mu)\right). \tag{22}$$

The covariance matrix can be parameterized using a scale vector and rotation parameter. Specifically, let $S(s)$ denote the diagonal scale matrix constructed from s, and let $R(r)$ denote the rotation matrix associated with r. Then:

$$\Sigma = R(r)S(s)S(s)^{\mathsf{T}}R(r)^{\mathsf{T}}. \tag{23}$$

The geometric parameters of each Gaussian are therefore jointly represented as:

$$G_{geo} = \{c, s, r\}. \tag{24}$$

The appearance parameters are represented by:

$$G_{app} = \{C, \alpha\}. \tag{25}$$

Thus, the complete Gaussian parameter set is:

$$G = \{c, s, r, C, \alpha\}. \tag{26}$$

The geometric offset predicted from the global depth representation and multi-scale motion representation is:

$$\Delta G = \mathcal{F}_{offset}\left(\mathcal{G}_{depth}, \mathcal{G}_{scale}\right) = \{\Delta c, \Delta s, \Delta r\}. \tag{27}$$

The updated geometric parameters can consequently be written component-wise as:

$$c_{new} = c + \Delta c. \tag{28}$$

$$s_{new} = s + \Delta s. \tag{29}$$

$$r_{new} = r + \Delta r. \tag{30}$$

Therefore, the updated Gaussian is:

$$G_{new} = \{c + \Delta c, s + \Delta s, r + \Delta r, C, \alpha\}. \tag{31}$$

The corresponding updated covariance matrix becomes:

$$\Sigma_{new} = R(r + \Delta r)S(s + \Delta s)S(s + \Delta s)^{\mathsf{T}}R(r + \Delta r)^{\mathsf{T}}. \tag{32}$$

This formulation explicitly separates the geometry update from the appearance parameters, allowing the predicted depth and multi-scale motion representations to primarily regulate the spatial configuration of the Gaussian primitives.

B.2 Dynamic Foreground Extraction

Given an input frame sequence $\{I_t\}_{t=1}^{T}$, frame differencing is used to identify regions with temporal pixel changes. The basic temporal difference between two consecutive frames is:

$$D_t = |I_t - I_{t-1}|. \tag{33}$$

A dynamic foreground mask can then be obtained using a threshold τ:

$$M_t^g(x, y) = \mathbb{I}\left(D_t(x, y) > \tau\right). \tag{34}$$

where $\mathbb{I}(\cdot)$ denotes the indicator function.

The global motion region extracted from the current frame can consequently be represented as:

$$M_t^g = \mathbb{I}\left(|I_t - I_{t-1}| > \tau\right). \tag{35}$$

The corresponding foreground feature can be expressed as:

$$F_t^g = M_t^g \odot I_t. \tag{36}$$

where $\odot$ denotes element-wise multiplication.

This operation isolates regions exhibiting temporal variations while suppressing static background content.

B.3 Illumination-Aware Global Motion Representation

Since pixel differences may also be caused by illumination changes or shadows, the lighting parameter L_t is introduced to provide additional contextual information. The enhanced global motion representation is obtained through:

$$M_t^{g'} = \mathcal{F}_{light}\left(M_t^g \cdot L_t\right). \tag{37}$$

Let the input feature of the lighting encoder be:

$$X_t^{light} = M_t^g \cdot L_t. \tag{38}$$

For an encoder-decoder architecture, the intermediate feature can be represented as:

$$E_t^{light} = \mathcal{E}_{light}\left(X_t^{light}\right). \tag{39}$$

and the enhanced global motion representation is reconstructed as:

$$M_t^{g'} = \mathcal{D}_{light}\left(E_t^{light}\right). \tag{40}$$

The enhanced representation is then encoded into the depth feature:

$$d_t = \mathcal{E}_{depth}\left(M_t^{g'}\right). \tag{41}$$

The depth inference network subsequently predicts the depth map:

$$Z_t = \mathcal{F}_{depth}\left(d_t\right). \tag{42}$$

B.4 Temporal Depth Consistency

The depth variation between two consecutive frames is defined as:

$$\Delta Z_t = Z_t - Z_{t-1}. \tag{43}$$

The spatial gradient of the depth map is:

$$\nabla Z_t = \left[\frac{\partial Z_t}{\partial x}, \frac{\partial Z_t}{\partial y}\right]^{\mathsf{T}}. \tag{44}$$

The magnitude of the depth gradient is given by:

$$\|\nabla Z_t\| = \sqrt{\left(\frac{\partial Z_t}{\partial x}\right)^2 + \left(\frac{\partial Z_t}{\partial y}\right)^2}. \tag{45}$$

The depth continuity loss introduced in the main method can therefore be expressed as:

$$\mathcal{L}_{depth} = \sum_t \left\|(1 - \nabla Z_t) \circ \Delta Z_t\right\|_1. \tag{46}$$

For an individual spatial location (x, y), the corresponding contribution can be written as:

$$\ell_{depth}^t(x, y) = \left|(1 - \nabla Z_t(x, y)) \circ (Z_t(x, y) - Z_{t-1}(x, y))\right|. \tag{47}$$

Thus, the loss measures temporal depth variation while incorporating the spatial depth-gradient information into the weighting process.

B.5 Depth Distribution Construction

The depth distribution saliency map is constructed from the absolute depth difference between consecutive frames:

$$M(depth) = \left|Z_t - Z_{t-1}\right|. \tag{48}$$

For each spatial position (x, y), this can be written as:

$$M(depth)(x, y) = \left|Z_t(x, y) - Z_{t-1}(x, y)\right|. \tag{49}$$

The resulting map therefore characterizes the magnitude of depth variation at each spatial location.

Let the feature representation extracted from the depth distribution be:

$$F(M(depth)) = \mathcal{E}_M\left(M(depth)\right). \tag{50}$$

The depth distribution weight is then predicted by an MLP:

$$\beta_t = \mathrm{MLP}\left(F(M(depth))\right). \tag{51}$$

If the MLP contains L layers, its recursive computation can be written as:

$$h_t^{(l)} = \sigma\left(W^{(l)} h_t^{(l-1)} + b^{(l)}\right), \qquad l = 1, \ldots, L-1. \tag{52}$$

with:

$$\beta_t = W^{(L)} h_t^{(L-1)} + b^{(L)}. \tag{53}$$

where $\sigma(\cdot)$ denotes the nonlinear activation function.

B.6 Depth Distribution Weight Regularization

The spatial gradient of the depth distribution weight is defined as:

$$\nabla\beta_t = \left[\frac{\partial\beta_t}{\partial x}, \frac{\partial\beta_t}{\partial y}\right]^{\mathsf{T}}. \tag{54}$$

The depth-gradient magnitude is:

$$g_t = \|\nabla Z_t\|. \tag{55}$$

The weighted smoothness loss is then:

$$\mathcal{L}_{depth}^{weight} = \sum_t \left\|\exp(g_t) \cdot \nabla\beta_t\right\|_1. \tag{56}$$

For each spatial location, the corresponding penalty can be expressed as:

$$\ell_{depth}^{weight}(x,y) = \exp\left(\|\nabla Z_t(x,y)\|\right) \left|\nabla\beta_t(x,y)\right|. \tag{57}$$

The exponential weighting introduces a spatially varying regularization coefficient:

$$w_t^{depth}(x,y) = \exp\left(\|\nabla Z_t(x,y)\|\right). \tag{58}$$

Therefore, the smoothness constraint can be interpreted as a weighted total-variation regularization on β_t:

$$\mathcal{L}_{depth}^{weight} = \sum_t \sum_{x,y} w_t^{depth}(x,y) \left|\nabla\beta_t(x,y)\right|. \tag{59}$$

B.7 Global Depth Feature Fusion

The global motion feature d_t, static background feature b_t, and depth distribution weight β_t are jointly incorporated into the fusion module. Their channel-wise representation can be written as:

$$X_t^{depth} = Concat\left(\beta_t \cdot d_t, b_t\right). \tag{60}$$

The fusion operation is then:

$$\mathcal{G}_{depth} = \mathcal{F}_{fusion}\left(X_t^{depth}\right). \tag{61}$$

For a Transformer-based fusion process, the query, key, and value projections can be expressed as:

$$Q_t = X_t^{depth} W_Q, \qquad K_t = X_t^{depth} W_K, \qquad V_t = X_t^{depth} W_V. \tag{62}$$

The attention matrix is calculated as:

$$A_t = \mathrm{softmax}\left(\frac{Q_t K_t^{\mathsf{T}}}{\sqrt{d}}\right). \tag{63}$$

The fused feature is then obtained as:

$$\widetilde{\mathcal{G}}_{depth}^t = A_t V_t. \tag{64}$$

With residual transformation, the final depth representation can be expressed as:

$$\mathcal{G}_{depth}^t = \mathcal{P}\left(X_t^{depth} + \widetilde{\mathcal{G}}_{depth}^t\right). \tag{65}$$

where $\mathcal{P}(\cdot)$ denotes the output projection of the fusion module.

B.8 Perceptual Depth Fusion Consistency

The fused depth representation is projected into a perceptual feature space using the lightweight encoder $\phi(\cdot)$:

$$z_{depth}^t = \phi\left(\mathcal{G}_{depth}^t\right). \tag{66}$$

The temporal perceptual difference is:

$$\Delta z_{depth}^t = z_{depth}^t - z_{depth}^{t-1}. \tag{67}$$

The perceptual consistency loss is consequently:

$$\mathcal{L}_{fusion} = \sum_t \left\| \Delta z_{depth}^t \right\|_1. \tag{68}$$

Expanding the feature norm gives:

$$\mathcal{L}_{fusion} = \sum_t \sum_d \left| z_{depth,d}^t - z_{depth,d}^{t-1} \right|. \tag{69}$$

where d indexes the dimensions of the perceptual feature. This objective encourages neighboring frames to maintain consistent high-level depth representations.

B.9 Local Temporal Window Construction

For a target frame I_t, a local temporal window with radius K is constructed as:

$$\mathcal{W}_K(t) = \{H_{t-K}, \ldots, H_t, \ldots, H_{t+K}\}. \tag{70}$$

The corresponding temporal index set is:

$$\mathcal{T}_t = \{t-K, \ldots, t+K\}. \tag{71}$$

For a frame near the beginning or end of the sequence, the index is clipped to the valid range:

$$\widetilde{t+\tau} = \min\left(T, \max\left(1, t+\tau\right)\right), \qquad \tau \in [-K, K]. \tag{72}$$

The resulting window can therefore be written as:

$$\mathcal{W}_K(t) = \left\{H_{\widetilde{t-K}}, \ldots, H_t, \ldots, H_{\widetilde{t+K}}\right\}. \tag{73}$$

This preserves a fixed window size of $2K+1$ for every target frame.

B.10 Local Motion Encoding

The temporal encoder maps the local window features into a motion representation:

$$M_t^l(x,y) = \mathcal{F}_{window}\left(\mathcal{W}_K(t)\right). \tag{74}$$

To explicitly characterize temporal variations, the first-order temporal difference can be defined as:

$$\Delta H_{t+\tau} = H_{t+\tau} - H_{t+\tau-1}. \tag{75}$$

The corresponding local motion magnitude is:

$$A_{t+\tau}(x,y) = \left|\Delta H_{t+\tau}(x,y)\right|. \tag{76}$$

The temporal variation over the local window can consequently be represented by:

$$A_t^{local}(x,y) = \frac{1}{2K}\sum_{\tau=-K+1}^{K}\left|\Delta H_{t+\tau}(x,y)\right|. \tag{77}$$

The temporal encoder can jointly model these variations through convolutional or temporal filtering operations:

$$M_t^l = \mathcal{F}_{window}\left(H_{t-K:t+K}\right). \tag{78}$$

Thus, $M_t^l(x,y)$ contains local motion information associated with the spatial location (x,y).

B.11 Multi-Scale Local Motion Aggregation

For an $h \times h$ local receptive field centered at (i,j), the local aggregated motion representation is defined as:

$$R_n(i,j) = \frac{1}{|W_h(i,j)|}\sum_{(x,y)\in W_h(i,j)} M_t^l(x,y). \tag{79}$$

Since:

$$|W_h(i,j)| = h^2. \tag{80}$$

the local aggregation can also be written as:

$$R_n(i,j) = \frac{1}{h^2} \sum_{(x,y)\in W_h(i,j)} M_t^l(x,y). \tag{81}$$

Different values of h define different receptive fields and therefore different motion scales:

$$\mathcal{R} = \{R_1, R_2, \ldots, R_N\}. \tag{82}$$

The feature sequence associated with scale n is denoted as:

$$v_t^n = \mathcal{E}_n(R_n), \qquad n = 1, \ldots, N. \tag{83}$$

This multi-scale construction enables the representation to simultaneously capture fine local motion and larger regional motion patterns.

B.12 Multi-Scale Reconstruction Constraint

For each scale n, the feature v_t^n is decoded to reconstruct the corresponding frame:

$$\hat{f}_t^n = \mathcal{D}_n\left(v_t^n\right). \tag{84}$$

The target frame and reconstructed frame are mapped into the perceptual feature space:

$$z_t = \phi(f_t), \qquad \hat{z}_t^n = \phi(\hat{f}_t^n). \tag{85}$$

The scale-specific reconstruction loss is then:

$$\mathcal{L}_{scale} = \left\| z_t - \hat{z}_t^n \right\|_1. \tag{86}$$

B.13 Multi-Scale Saliency and Weight Prediction

Each scale-specific feature sequence is decoded into a multi-scale saliency map:

$$M^n(scale) = \mathcal{D}_{scale}^n\left(v_t^n\right). \tag{87}$$

Its feature representation is extracted as:

$$F\left(M^n(scale)\right) = \mathcal{E}_{scale}\left(M^n(scale)\right). \tag{88}$$

The corresponding scale weight is predicted using a fully connected layer:

$$\lambda_t^n = W_\lambda F\left(M^n(scale)\right) + b_\lambda. \tag{89}$$

For adaptive weighting, the raw scale scores can be normalized as:

$$\widetilde{\lambda}_t^n = \frac{\exp(\lambda_t^n)}{\sum_{m=1}^N \exp(\lambda_t^m)}. \tag{90}$$

The normalized weights satisfy:

$$\sum_{n=1}^{N} \widetilde{\lambda}_t^n = 1, \qquad \widetilde{\lambda}_t^n > 0. \tag{91}$$

The normalized weights therefore form a probability-like distribution over motion scales.

B.14 Scale Weight Smoothness

The difference between adjacent scale weights is:

$$\Delta\lambda_t^n = \lambda_t^{n+1} - \lambda_t^n. \tag{92}$$

The scale smoothness objective in the main method can be written as:

$$\mathcal{L}_{scale}^{weight} = \frac{1}{T} \sum_{t=1}^{T} \frac{1}{N-1} \sum_{n=1}^{N-1} \left\| \Delta\lambda_t^n \right\|_2 . \tag{93}$$

For scalar scale weights, the corresponding term reduces to:

$$\left\| \Delta\lambda_t^n \right\|_2 = \left| \lambda_t^{n+1} - \lambda_t^n \right| . \tag{94}$$

Minimizing this objective encourages adjacent motion scales to have smoothly varying contributions rather than abrupt changes in their importance.

B.15 Multi-Scale Alignment

The weighted multi-scale representation is constructed by combining the scale-specific feature sequences:

$$Z_t^{scale} = \sum_{n=1}^{N} \lambda_t^n v_t^n. \tag{95}$$

If normalized weights are used, this becomes:

$$Z_t^{scale} = \sum_{n=1}^{N} \widetilde{\lambda}_t^n v_t^n. \tag{96}$$

The final multi-scale pattern representation is then obtained through the alignment network:

$$\mathcal{G}_{scale} = \mathcal{F}_{alignment} \left(Z_t^{scale} \right) . \tag{97}$$

For a linear projection followed by normalization, the operation can be represented as:

$$\widetilde{\mathcal{G}}_{scale} = W_g Z_t^{scale} + b_g. \tag{98}$$

followed by:

$$\mathcal{G}_{scale} = \frac{\widetilde{\mathcal{G}}_{scale}}{\| \widetilde{\mathcal{G}}_{scale} \|_2 + \epsilon}. \tag{99}$$

This operation maps motion information from different receptive-field scales into a unified representation space.

B.16 Motion Region Stability Constraint

Let Mask_t denote the binary motion-region mask in the generated frame. The motion-region area can be measured by:

$$A_t = \|\mathrm{Mask}_t\|_1. \tag{100}$$

Since the mask is binary, A_t corresponds to the number of pixels classified as motion regions.

The temporal change in motion-region area is:

$$\Delta A_t = A_t - A_{t-1}. \tag{101}$$

The region stability loss is therefore:

$$\mathcal{L}_{alignment} = \sum_{t=1}^{T} |\Delta A_t|. \tag{102}$$

Equivalently:

$$\mathcal{L}_{alignment} = \sum_{t=1}^{T} \left| \|\mathrm{Mask}_t\|_1 - \|\mathrm{Mask}_{t-1}\|_1 \right|. \tag{103}$$

This objective constrains the temporal variation of the motion-region area and reduces abrupt expansion or contraction of the generated motion regions.

B.17 Joint Geometric Offset Prediction

The GMDR and LMMR modules provide complementary representations:

$$\mathcal{G}_{depth} = \mathcal{F}_{fusion}\left(\beta_t \cdot d_t + b_t\right). \tag{104}$$

and:

$$\mathcal{G}_{scale} = \mathcal{F}_{alignment}\left(\sum_{n=1}^{N} \lambda_t^n v_t^n\right). \tag{105}$$

These two representations are jointly provided to the geometric offset prediction module:

$$\Delta G = \mathcal{F}_{offset}\left(\mathcal{G}_{depth}, \mathcal{G}_{scale}\right). \tag{106}$$

With a branched architecture, the geometric offset can be decomposed into three prediction branches:

$$\Delta c = \mathcal{F}_c\left(\mathcal{G}_{depth}, \mathcal{G}_{scale}\right). \tag{107}$$

$$\Delta s = \mathcal{F}_s\left(\mathcal{G}_{depth}, \mathcal{G}_{scale}\right). \tag{108}$$

$$\Delta r = \mathcal{F}_r\left(\mathcal{G}_{depth}, \mathcal{G}_{scale}\right). \tag{109}$$

The resulting geometric offset is:

$$\Delta G = \{\Delta c, \Delta s, \Delta r\}. \tag{110}$$

The updated Gaussian parameters are then:

$$G_{new} = G + \Delta G. \tag{111}$$

Therefore, the overall geometric transformation can be summarized as:

$$G \xrightarrow{\mathcal{G}_{depth}, \mathcal{G}_{scale}} \Delta G \rightarrow G_{new}. \tag{112}$$

B.18 Overall Optimization Objective

The GMDR module contains the depth temporal consistency, depth-weight smoothness, and perceptual fusion consistency objectives. Their weighted combination can be expressed as:

$$\mathcal{L}_{GMDR} = \lambda_{light}\mathcal{L}_{light} + \lambda_{depth}\mathcal{L}_{depth} + \lambda_{depth}^{weight}\mathcal{L}_{depth}^{weight} + \lambda_{fusion}\mathcal{L}_{fusion}. \tag{113}$$

The LMMR module contains the multi-scale reconstruction, scale-weight smoothness, and motion-region alignment objectives:

$$\mathcal{L}_{LMMR} = \lambda_{window}\mathcal{L}_{window} + \lambda_{scale}\mathcal{L}_{scale} + \lambda_{scale}^{weight}\mathcal{L}_{scale}^{weight} + \lambda_{alignment}\mathcal{L}_{alignment}. \tag{114}$$

The complete representation learning objective is:

$$\mathcal{L}_{representation} = \mathcal{L}_{GMDR} + \mathcal{L}_{LMMR}. \tag{115}$$

Expanding the two components gives:

$$\mathcal{L}_{representation} = \lambda_{light}\mathcal{L}_{light} + \lambda_{depth}\mathcal{L}_{depth} + \lambda_{depth}^{weight}\mathcal{L}_{depth}^{weight} + \lambda_{fusion}\mathcal{L}_{fusion} + \tag{116}$$

$$\lambda_{window}\mathcal{L}_{window} + \lambda_{scale}\mathcal{L}_{scale} + \lambda_{scale}^{weight}\mathcal{L}_{scale}^{weight} + \lambda_{alignment}\mathcal{L}_{alignment}. \tag{117}$$

The learned depth and multi-scale representations are subsequently used to predict geometric offsets for the 3D Gaussian primitives:

$$\Delta G = \mathcal{F}_{offset}\left(\mathcal{G}_{depth}, \mathcal{G}_{scale}\right). \tag{118}$$

The final Gaussian representation is obtained as:

$$G_{new} = G + \Delta G. \tag{119}$$

Thus, the complete computational process can be summarized as:

$$\{I_t\}_{t=1}^{T} \rightarrow \{\mathcal{G}_{depth}, \mathcal{G}_{scale}\} \rightarrow \Delta G \rightarrow G_{new}. \tag{120}$$

During inference, the learned geometric representation and motion-aware representations provide the conditions required for generating temporally coherent talking portrait sequences from the input audio-driven motion process.

C LIGHTWEIGHT TEACHER–STUDENT MODEL

The two proposed strategies consist of multiple submodules. To further improve efficiency, we adopt a knowledge distillation strategy to distill the key submodules into a unified lightweight student model, as shown in Figure 7.

Specifically, our approach consists of eight core submodules: the brightness adjustment component $\mathcal{F}_{light}$, the depth estimation component $\mathcal{F}_{depth}$, the depth distribution weight component $\mathcal{F}_{depth-weight}$, the fusion component $\mathcal{F}_{fusion}$, the window component $\mathcal{F}_{window}$, the multi-scale feature component $\mathcal{F}_{scale}$, the multi-scale motion weight component $\mathcal{F}_{scale-weight}$, and the alignment component $\mathcal{F}_{alignment}$. These submodules are responsible for predicting and updating the parameters of Gaussian points for rendering, playing a critical role in generating the final output.

During training, we adopt a feature-level knowledge distillation scheme, where each student submodule $\mathcal{F}_k^{student}$ learns to approximate its teacher counterpart by minimizing feature discrepancies.

Here, $k \in 1, \ldots, 8$ denotes the index of the eight student submodules, including $\mathcal{F}_{light}^{student}$, $\mathcal{F}_{depth}^{student}$, $\mathcal{F}_{depth\text{-}weight}^{student}$, $\mathcal{F}_{fusion}^{student}$, $\mathcal{F}_{window}^{student}$, $\mathcal{F}_{scale}^{student}$, $\mathcal{F}_{scale\text{-}weight}^{student}$, and $\mathcal{F}_{alignment}^{student}$. This design enables a lightweight and efficient student model while preserving the performance of the teacher model.

For each submodule, we define the following distillation loss:

$$\mathcal{L}_k = \left\| \mathbf{F}_k^{student} - \mathbf{F}_k \right\|_2^2 . \tag{121}$$

where $\mathbf{F}_k$ denotes the output feature of the corresponding teacher submodule, including $\mathcal{F}_{light}$, $\mathcal{F}_{depth}$, $\mathcal{F}_{depth\text{-}weight}$, $\mathcal{F}_{fusion}$, $\mathcal{F}_{window}$, $\mathcal{F}_{scale}$, $\mathcal{F}_{scale\text{-}weight}$, and $\mathcal{F}_{alignment}$.

The total distillation loss is then formulated as a weighted sum:

$$\mathcal{L}_{total} = \sum_{k=1}^{8} \lambda_k \mathcal{L}_k = \sum_{k=1}^{8} \lambda_k \left\| \mathbf{F}_k^{student} - \mathbf{F}_k \right\|_2^2 . \tag{122}$$

where λ_k represents an adjustable hyperparameter used to balance the relative importance of different submodules.

As shown in Figure 8, we sample several frames from a continuous sequence to compare the ability of different models to reproduce subtle dynamic changes. The real video exhibits natural wrinkle variations, and the lightweight model achieves performance comparable to the full model, accurately capturing these subtle variations in motion regions.

Furthermore, we conduct quantitative comparisons. As shown in Table 3, we compare the proposed method with other methods and the full model. The results show that the lightweight model achieves the fastest generation speed, reaching 135 FPS. As shown in Table 4, we conduct ablation studies to validate the effectiveness of the lightweight model. The results demonstrate that the model effectively preserves visual quality. Overall, the proposed lightweight model improves generation efficiency while maintaining quality.

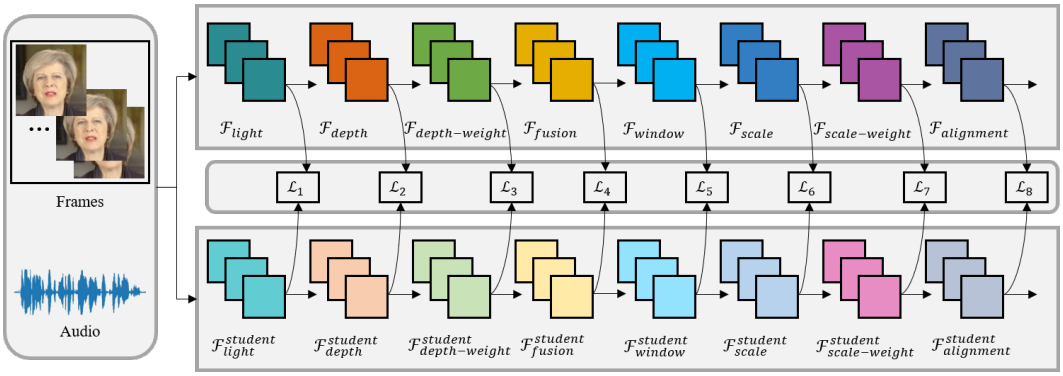


Figure 7: A lightweight model is proposed based on this paper.



Figure 8: Comparative results. We show keyframe visualizations. It can be seen that our method and the corresponding lightweight model outperform other state-of-the-art methods, demonstrating subtle dynamic changes.

Table 3: Quantitative comparison.

Methods	PSNR ↑	LPIPS ↓	SyncNet ↑	LSE-C ↑	LMD ↓	CSIM ↑	FPS ↑
Ground Truth	N/A	N/A	7.647	7.524	N/A	1	N/A
ER-NeRF	30.145	0.038	6.594	6.894	4.017	0.935	27
SyncTalk	30.446	0.039	6.348	6.784	3.984	0.928	23
S^{3}D-NeRF	30.349	0.038	6.764	6.861	4.152	0.934	9.5
PointTalk	30.484	0.037	6.854	7.021	3.764	0.929	69
InsTaG	30.494	0.038	6.497	7.016	3.814	0.932	72
GaussianTalker	30.816	0.037	6.585	6.946	3.906	0.933	81
Ours(Lightweight)	33.321	0.027	7.044	7.211	2.776	0.945	**135**
Ours	**33.324**	**0.025**	**7.045**	**7.212**	**2.775**	**0.946**	87

D Experimental Supplement

D.1 User Study

We designed a user study to evaluate the subjective performance of generated videos. We invited 50 participants to watch a large number of speaking videos, including our proposed method and the

Table 4: Quantitative ablation study.

Methods	PSNR ↑	LPIPS ↓	SyncNet ↑	LSE-C ↑	LMD ↓	CSIM ↑
Ground Truth	N/A	N/A	7.549	7.436	N/A	1
Ours w/o $\mathcal{F}_{light}$	30.412	0.042	6.573	6.814	3.824	0.932
Ours w/o $\mathcal{F}_{depth}$	30.602	0.041	6.416	6.758	4.115	0.928
Ours w/o $\mathcal{F}_{depth-weight}$	30.185	0.040	6.525	6.705	4.069	0.926
Ours w/o $\mathcal{F}_{fusion}$	30.376	0.039	6.621	6.822	3.958	0.929
Ours w/o $\mathcal{F}_{window}$	30.489	0.035	6.518	6.872	3.824	0.932
Ours w/o $\mathcal{F}_{scale}$	30.826	0.038	6.323	6.609	3.782	0.927
Ours w/o $\mathcal{F}_{scale-weight}$	30.638	0.037	6.417	6.725	3.707	0.930
Ours w/o $\mathcal{F}_{alignment}$	30.634	0.036	6.582	6.904	3.659	0.931
Ours(Lightweight)	33.325	0.027	7.056	7.157	2.785	0.945
Ours	**33.326**	**0.026**	**7.058**	**7.158**	**2.784**	**0.947**

previously mentioned comparative methods. Participants rated the videos on four aspects (1-5, with higher scores indicating better performance): image quality, video smoothness, identity consistency, and motion consistency.

Image quality: Evaluates clarity and detail reproduction, including skin texture, facial contours, and color fidelity. Rendering artifacts, noise, or distortions that reduce realism are also considered.

Video smoothness: Assesses the continuity and naturalness of facial and head movements, including smooth transitions, lip-sync with speech, and timely, consistent expression changes.

Identity consistency: Measures the stability and accuracy of facial features throughout the video, ensuring consistent identity across angles, and maintaining reasonable facial size, proportions, and structure.

Motion consistency: Evaluates coordination and realism of movements across facial regions, ensuring natural eye, mouth, and eyebrow motions, appropriate movement ranges, stable facial structure, and subtle micro-expressions.

The experimental results are shown in Figure 9. Our generated results are visually very close to real videos, demonstrating the practical value and generative capability of our method. Overall, combining quantitative metrics and subjective evaluations, our method achieves significant improvements in the realism and naturalness of talking videos.



Figure 9: User study.

D.2 PARAMETER SETTINGS

In the local motion multi-scale representation (LMMR) module, we characterize motion differences through the multi-scale variations of local motion regions, thereby ensuring structural consistency of motion regions.

Specifically, we introduce R_n to denote the n-th scale, where $1 \leq n \leq N$, and N is the total number of scales. Since the number of scales has a significant impact on model performance, we conduct a parameter study to evaluate different scale configurations, as illustrated in Figure 10.

In this study, we vary the number of receptive field windows used for aggregating local motion regions to evaluate the capability of the model to capture subtle motions under different scale settings. Then, we analyzed the contribution of multi-scale features to motion consistency and detail restoration, and determined the optimal scale configuration. Experimental results show that when $N = 4$, the structure of local motion regions is most complete, and the hair and head movements closely match those in real videos, effectively avoiding structural misalignment and discontinuities.



Figure 10: Comparison of parameters for the number of feature layers. We can see that when $N = 4$, the model performs best and can effectively avoid structural misalignment.

D.3 EFFICIENCY EVALUATION

We conduct experiments to evaluate the efficiency of different methods in terms of generation speed (the FPS column in Table 1) and training performance (Figure 11). To ensure a fair comparison, all models were trained on the same hardware platform with consistent key settings, including an initial learning rate of 1×10^{-3}, a batch size of 16 frames, and the same training schedule.

In terms of inference speed, NeRF-based methods are generally slow due to the need for extensive ray sampling, whereas 3DGS-based methods avoid complex volume rendering processes by utilizing sparse Gaussian points, thereby improving efficiency. During training, NeRF-based methods often exhibit significant loss fluctuations or slow convergence, whereas 3DGS-based methods adopt explicit representations and therefore converge faster and more stably. In contrast, our method represents the portrait geometry as global and local motion regions, significantly reducing model complexity and computational cost, thereby achieving faster inference speed and exhibiting the most stable convergence. In summary, the proposed method achieves higher efficiency while enhancing geometric representation consistency.

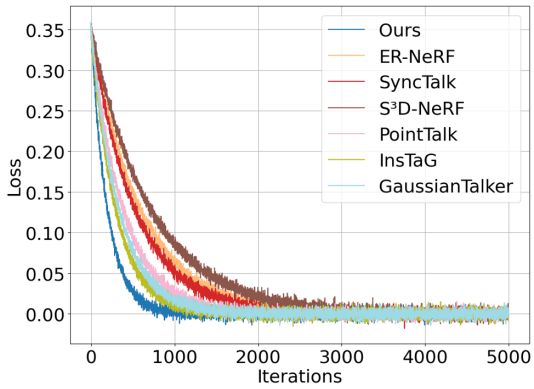


Figure 11: Model performance.

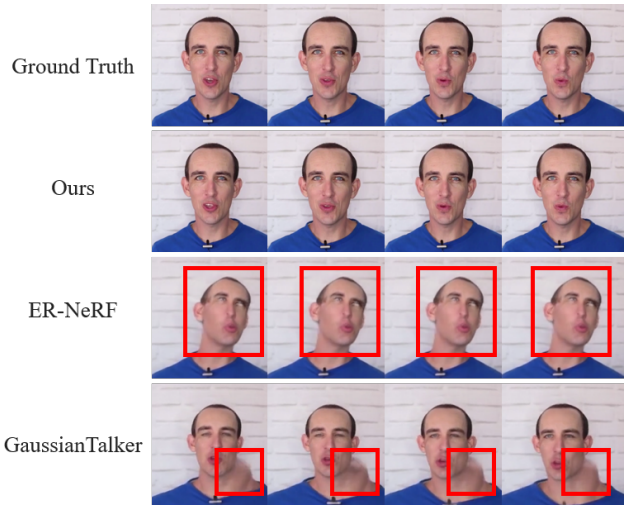


Figure 12: Comparison under short training video. Existing methods fail to reconstruct accurate portrait geometry under sparse training data, leading to noticeable distortions, while our method preserves more coherent and stable geometric structures.

D.4 Training with Short Videos

Our method is based on 3DGS, which typically relies on rich multi-view data. To evaluate its robustness under sparse training data, we use a 10-second training video. We then compare our approach with ER-NeRF based on NeRF and GaussianTalker based on 3DGS. As shown in Figure 12, with such limited input, both methods struggle to recover accurate portrait geometry, exhibiting noticeable structural distortions and inconsistencies. In contrast, our method better preserves geometric consistency and accuracy, producing visually more stable and plausible results. These results demonstrate that our method maintains strong geometric fidelity for portrait reconstruction even with extremely sparse training data.

D.5 Limitations

This paper proposes a 3DGS-based talking portrait generation method that effectively alleviates issues such as proportion instability and structural misalignment in motion regions. However, as shown in Figure 13, this method still has some limitations. The geometry of the hair is not finely modeled as individual hair strands, but rather as continuous, smooth, drooping blocks of color, even exhibiting unnatural perspective effects in some areas. This is mainly because 3DGS-based multi-

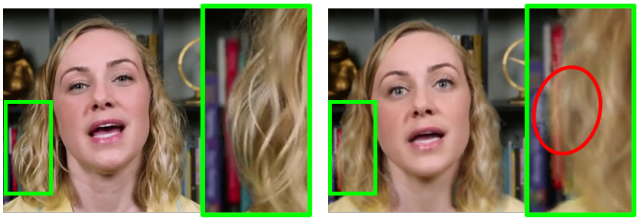


Figure 13: Some failure examples. Hair areas are often generated as continuous, smooth blocks of color, making it difficult to restore their true physical structure.

view synthesis methods still rely on high-quality, dense input data sequences, making it impossible to physically model actual 3D geometry.

In other words, the current method primarily relies on large amounts of image sequences, achieving smooth visual detail transitions by leveraging temporal continuity between adjacent frames, rather than constructing a complete 3D physical model. This limits the generalization performance of the model in complex scenes and extreme poses.

To this end, we plan to further study the appearance properties (C, α), building on the geometric property optimizations (c, s, r) presented in this paper, in order to explore effective strategies for detail recovery.