

# Reconstruction and Reenactment Separated Method for Realistic Gaussian Head

Zhiling Ye, Cong Zhou, Xiubao Zhang, Haifeng Shen, Weihong Deng, Quan Lu

Mashang Consumer Finance Co., Ltd.

{zhiling.ye, cong.zhou01, xiubao.zhang, haifeng.shen, weihong.deng, quan.lu}@msxf.com

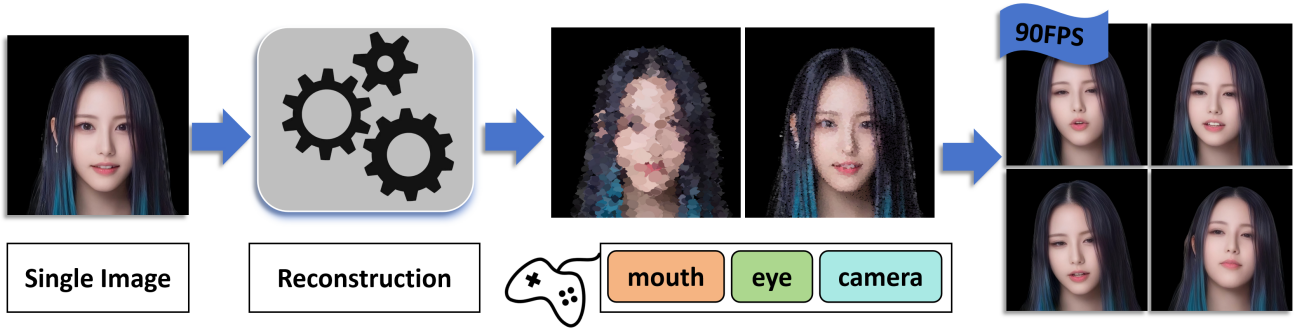


Figure 1: Our method reconstruct a single portrait image into 3D Gaussian model. The model can be reenacted with fine-grained options and rendered at 90 FPS. The person in the picture is a virtual character generated by AIGC cited from the Internet, named Yuri.

## Abstract

In this paper, we explore a reconstruction and reenactment separated framework for 3D Gaussians head, which requires only a single portrait image as input to generate controllable avatar. Specifically, we developed a large-scale one-shot gaussian head generator built upon WebSSL (Fan and others. 2025) and employed a two-stage training approach that significantly enhances the capabilities of generalization and high-frequency texture reconstruction. During inference, an ultra-lightweight gaussian avatar driven by control signals enables high frame-rate rendering, achieving 90 FPS at a resolution of  $512 \times 512$ . We further demonstrate that the proposed framework follows the scaling law, whereby increasing the parameter scale of the reconstruction module leads to improved performance. Moreover, thanks to the separation design, driving efficiency remains unaffected. Finally, extensive quantitative and qualitative experiments validate that our approach outperforms current state-of-the-art methods.

## Introduction

Avatar reconstruction from a single portrait image is challenging yet promising with applications in video conferencing, filmmaking, and game production. Researchers globally have explored solutions using end-to-end 2D methods and approaches that leverage 3D priors.

Specifically, 2D-based approaches (Goodfellow and others. 2014; Isola and others. 2017; Karras and others. 2019, 2020; Guo and others. 2024) mainly rely on deep convolutional networks and generative adversarial models, achieving good control of facial expressions and body postures in the source image through the construction and modulation of warp field. Notably, in recent years, with the rapid development of generative models such as image and video diffusion models, research based on 2D schemes has achieved a series of significant advancements (Cui and others. 2024; Tian and others. 2024; Chen and others. 2025). However, the distinct advantages were demonstrated in many practical applications, their inherent limitation lies in the lack of

explicit 3D structural priors. The lack of 3D priors as guidance leads to the need of more complex model structures and larger model sizes for 2D methods to achieve end-to-end solutions. Consequently, this not only requires substantial computational resources but also inevitably introduces higher latency issues.

On the other hand, cutting-edge 3D synthesis technologies such as NeRF (Neural Radiance Fields) (Mildenhall and others. 2020) and 3DGS (3D Gaussian Splatting) (Kerbl and others. 2023) can be employed to achieve efficient and accurate character avatar reconstruction and reenactment, while effectively maintaining consistency and coherence from multiple viewpoints. It is worth noting that although these methods (Gafni and others. 2021; Bai and others. 2023; Yu and others. 2023; Zheng and others. 2023; Chu and others. 2024b,a; He and others. 2025) have demonstrated high precision in both theoretical and experimental settings, they generally rely on the precise estimation of the 3D pose of the character from a single image. This step introduces estimation errors that can cause texture inaccuracies and expression distortions. Additionally, the methods rely heavily on the 3DMM mesh, which is a widely-used 3D morphable model, to drive expressions, but its limited capabilities make reproducing subtle, natural expressions challenging.

We proposed a **Reconstruction And Reenactment** separated method, named **RAR**. RAR predicts 3D Gaussians from a single head image, allowing control over eyes, mouth shape, and expression. Our decoupled architecture separates appearance reconstruction and expression control, ensuring precise reconstruction and high frame-rate reenactment. The appearance feature is extracted by using WebSSL-7B (Fan and others. 2025) to create a standardized feature map with a canonical expression, which is then processed by a 3D Gaussian Generator to obtain texture and structural details. A lightweight reenactment module converts driving information into positions that reenact the 3D gaussian head at 90 FPS. Our method outperforms most state-of-the-art ap-

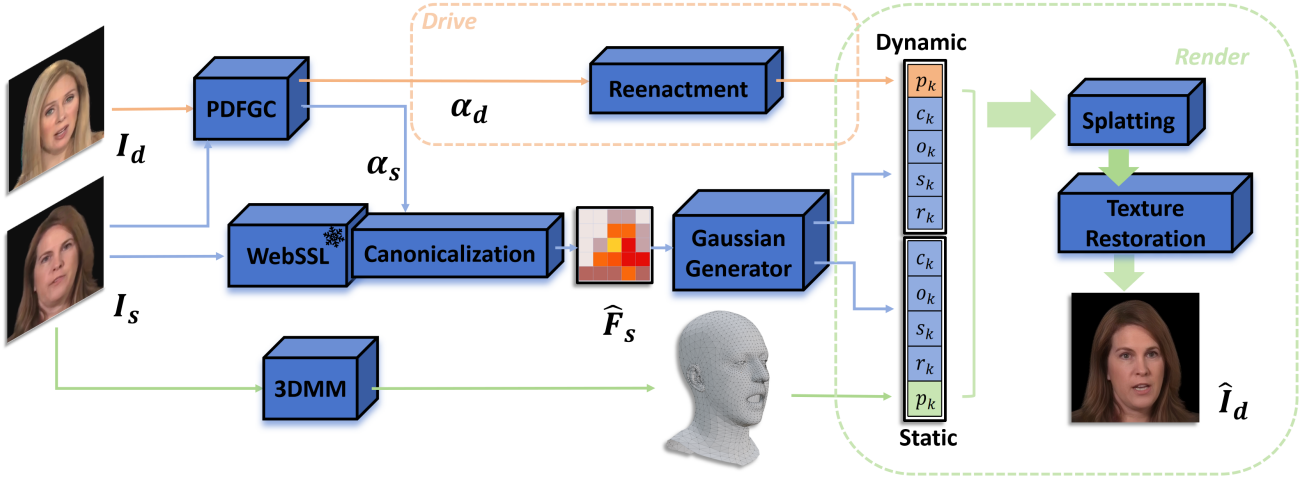


Figure 2: **The full pipeline of RAR.** For source portrait image  $I_s$ , we extract control condition  $\alpha_d$  from driving image  $I_d$  to independently adjust features (eyes, mouth, expressions) and obtain the rendered  $\hat{I}_d$ . Firstly, we extract the corresponding canonical feature  $\hat{F}_s$  from  $I_s$ . Then the gaussian generator predicts appearance details (colors, opacity, scales), splitting them into static  $\mathcal{G}_{static}$  and dynamic  $\mathcal{G}_{dynamic}$  parts. The positions of static part  $\mathcal{G}_{static}$  were proposed by a mesh of 3D Morphable Models, namely FLAME (Li and others. 2017). The reenactment module maps  $\alpha_d$  to the positions  $p_k$  of dynamic part  $\mathcal{G}_{dynamic}$ . The image splatted from static and dynamic 3D Gaussians was refined by the texture restoration module to produce the final output  $\hat{I}_d$ . More details about each module can be found in the supplementary material.

proaches on extensive experiments and comparisons.

And errors introduced by 3D estimation can result in inaccurate recovery of appearance textures during avatar reconstruction. Such artifacts are particularly prone to occur in regions with high-frequency textures, such as teeth, hair, and eyelashes. On the other hand, 2D end-to-end approaches (Wang and others. 2021a; Guo and others. 2024) use a warping field to distort the input image and generate pseudo-3D motions. These distortions approximate 3D motion using relatively simple transformations in 2D space, making it difficult for the human visual system to distinguish between these 2D transformations and true 3D motion. However, in terms of algorithmic learning complexity, the pseudo-3D motions patterns generated by the 2D end-to-end approach are much easier to learn than real 3D motion patterns. Therefore, to eliminate the artifacts induced by errors in 3D estimation, we add an additional texture restoration module and synthetic data in a second phase after the pre-training. Specifically, we freeze the previously trained model and train the texture restoration module solely using synthetic data. In this work, we use the 2D end-to-end model Live Portrait (Guo and others. 2024) to generate the synthetic data. To address the challenge of capturing fine-grained expression details. We abandon control schemes based on 3DMM (Li and others. 2017) or geometric structures and introduce a approach based on implicit parameters. This allows for independent control of facial features such as the eyes and mouth, and it can reproduce delicate and natural expressions. Leveraging this characteristic, we train an audio-to-motion (audio2motion) model that maps speech to mouth shape parameters, achieving a speech-driven avatar reconstruction and further validating the excellent control capability. This speech-driven experiment can be found in the supplementary material.

Our main contributions are as follows:

- We designed a decoupled architecture for appearance reconstruction and expression reenactment, employing a larger-scale WebSSL backbone to enhance the generalization and accuracy of the reconstruction module. meanwhile, the expression driving utilizes a lightweight model to achieve a driving speed of 90 FPS.
- We introduced a texture restoration module and a two-stage training process using synthetic data, effectively eliminating artifacts in high-frequency textures caused by 3D reconstruction errors.
- Extensive experiments and comparisons on a benchmark dataset verify the effectiveness and breakthrough performance of our method.
- Finally, our experiments show that the separation architecture follows the scaling law. Namely increasing the parameter scale of the reconstruction module improves the performance of algorithm. This enhancement does not compromise efficiency during driving.

## Related work

Recent advances in generating controllable head avatar with single portrait image can be broadly divided into two categories: 2D end-to-end image synthesis approaches and 3D explicit structural prior-based methods.

The 2D techniques primarily utilize convolutional neural networks (CNNs) (Goodfellow and others. 2014; Isola and others. 2017; Karras and others. 2020) to achieve end-to-end image synthesis, and they extensively employ generative adversarial networks (GANs). Early methods (Zakharov and others. 2019; Burkov and others. 2020; Wang and others. 2023) focused on integrating both expression and pose features into the generator network, typically using architectures such as U-Net or StyleGAN. Other approaches in this category (Burkov and others. 2020; Guo and others. 2024;





Figure 3: Visualization of cross-reenacted results on the VFHQ and HDTF dataset.

Hong and others. 2022; Zhang and others. 2023) represent expressions and poses as deformation fields applied to the source image. Thanks to advancements in image and video diffusion networks, recently diffusion model-based techniques (Cui and others. 2024; Tian and others. 2024; Chen and others. 2025) have been adopted to improve synthesis quality. However, these methods still face challenges such as high latency and computational resource demands. Moreover, the lack of an explicit 3D structure in 2D-based approaches often leads to unrealistic distortions or changes in identity features when handling significant pose or expression variations. Although some methods (Blanz and others. 1999; Gerig and others. 2018; Li and others. 2017; Paysan and others. 2009) introduce 3D morphable model, namely 3DMM, (Li and others. 2017) to mitigate these issues, they ultimately fall short in supporting free-viewpoint rendering due to insufficient concrete 3D structural constraints. In contrast, methods incorporating 3D structural priors offer stronger geometric consistency and free-viewpoint rendering capabilities. Early 3D approaches (Khakhulin and others. 2022; Xu and others. 2020) used 3DMM for head avatar reconstruction, and later, the emergence of neural radiance

fields (NeRFs) (Mildenhall and others. 2020) prompted numerous recent methods (Bai and others. 2023; Chu and others. 2024b; Deng and others. 2024a,b; Chu and others. 2024a; He and others. 2025; Ye and others. 2024; Yu and others. 2023; Zheng and others. 2023; Zielonka and others. 2023; Ye and others. 2025). However, NeRF-based methods typically require large amounts of training data, including multi-view or even monocular videos, which raises privacy concerns and limits their generalization to unseen identities. Some methods (Sun and others. 2023, 2024; Tang and others. 2024; Xu and others. 2023; Zhuang and others. 2022) attempt to bypass the need for extensive datasets by training the generator with random noise and then reconstructing specific identities through inversion, though the accuracy of inversion remains challenging. Additionally, test-time optimization has been explored as an alternative, but its computational cost hinders practical application.

Recent studies (Chu and others. 2024b; Deng and others. 2024a,b; Li and others. 2023; Han and others. 2024; Ye and others. 2024) have investigated single-sample 3D head reconstruction methods to address the limitations in data requirements and computational efficiency. These methods

leverage various techniques, including tri-plane features, deformation fields, point-based expression fields, and vertex feature transformers. Despite these advances, NeRF-based approaches still struggle to achieve real-time rendering. Recently, 3D Gaussian Splatting (Kerbl and others. 2023) has emerged as a promising alternative, capable of generating high-quality results at significantly faster rendering speeds. However, existing gaussian splatting methods (Qian and others. 2024; Xu and others. 2024) typically depend on training with video data of specific individuals, thus limiting their generalization to novel identities. The latest work, GAGA-vatar (Chu and others. 2024a) and LAM (He and others. 2025), proposes a single-sample 3D Gaussian head digital avatar generation method. However, it still relies on precise 3D pose estimation of the training data. The inherent errors in estimating 3D poses from monocular videos make it challenging to accurately reconstruct the appearance of subject.

Our method not only makes effective use of 3D structural priors to maintain geometric consistency, free-viewpoint rendering capability, and efficiency, but also leverages the high visual quality of 2D end-to-end synthesis to address the high-frequency texture degradation caused by pose estimation errors in 3D methods.

## Methodology

As illustrated in the Fig.2, RAR is a reconstruction and reenactment separated scheme that predicts the corresponding 3D Gaussian representation  $\mathcal{G}$ , with the single input portrait image  $I_s$ . By utilizing the driving image  $I_d$ , we extract an implicit control condition  $\alpha_d$ , which enables independent manipulation of the eyes, mouth, and expressions to achieve the desired rendered result  $\hat{I}_d$ . For the source image  $I_s$ , our reconstruction module predicts appearance-related information including colors, opacity, and scales of the portrait. This appearance-related information is split into static  $\mathcal{G}_{static}$  and dynamic  $\mathcal{G}_{dynamic}$  parts. The positions of static  $\mathcal{G}_{static}$  is proposed by FLAME which is a widely-used 3DMM model. On the other hand, the reenactment module maps the implicit control parameters  $\alpha_d$  to positions  $p_k$  of dynamic part  $\mathcal{G}_{dynamic}$ . These appearance-related attributes and position-related attributes are integrated into a complete 3D Gaussian representation, which is subsequently refined by the texture restoration module to produce the final output.

### Condition Extraction

We estimate a shape mesh (excluding expressions and poses) from the source image  $I_s$  using the FLAME (Li and others. 2017) model. The positions of 5,023 vertices in this mesh serve as the fixed points for the static component of the 3D Gaussian representation, denoted as  $\{p_k\}_{static}$ . The corresponding appearance attributes (colors, etc.) are predicted by the gaussian generator which we will introduce in the following part.

PDFGC (Wang and others. 2023) decomposes facial dynamics (e.g., lip motion, head pose, eye gaze/blink) into orthogonal latent codes through progressive representation learning. Our framework adopts PDFGC to extract control priors from images. For both the source image  $I_s$  and driv-

ing image  $I_d$ , we derive their respective control parameters  $\alpha_s$  and  $\alpha_d$  via PDFGC:

$$\alpha_s = PDFGC(I_s), \alpha_d = PDFGC(I_d), \quad (1)$$

### Reconstruction Module

The reconstruction module comprises two components: the feature extraction block, the canonicalization block. Differ from previous method (Deng and others. 2024b; Chu and others. 2024a; He and others. 2025), we adopt the pre-trained WebSSL as the feature extraction block. Compared with DinoV2 (Oquab and others. 2024), WebSSL benefits from a larger scale and a more extensive training dataset, thereby enabling the extraction of superior features for more effective downstream task performance. A input portrait image  $I_s$  is passed through the feature extraction block to obtain its corresponding feature representation  $F_s$ . The feature representation  $F_s$  of the source image along with its corresponding  $\alpha_s$  is fed into the canonicalization block for feature normalization. The canonicalization block employs cross-attention to facilitate the interaction between  $\alpha_s$  and  $F_s$ , ultimately outputting the normalized feature representation  $\hat{F}_s$ . The following equations show the simplified pipeline:

$$F_s = \mathcal{F}(I_s), \hat{F}_s = \mathcal{C}(F_s, \alpha_s), \quad (2)$$

where  $\mathcal{F}$  is denoted as WebSSL-based feature extraction block, and canonicalization block is denoted as  $\mathcal{C}$ .

### Gaussian Generator

The Gaussian Generator be made up of 2D convolutional layers and is divided into two components. The first is Static Block which predicts static appearance-related attributes from  $\hat{F}_s$  while contains the positions derived from FLAME mesh as  $\{p_k\}_{static}$  to integrate the static 3D Gaussian. The static 3D Gaussian ensure the foundational identity representation, primarily encoding low-frequency textures (e.g., facial contours and skin tone uniformity) in rendered outputs. And Dynamic Block processes the canonical features  $\hat{F}_s$  to predict dynamic appearance attributes. These attributes, combined with PDFGC-driven positions  $\{p_k\}_{dynamic}$ , form the dynamic 3D Gaussian. This component specializes in rendering high-frequency details (e.g., eyes, wrinkles, teeth) and motion dynamics (e.g., blinking, opening mouth, expression changes). The static Gaussian  $\mathcal{G}_{static}$  and dynamic Gaussian  $\mathcal{G}_{dynamic}$  components are concatenated into a unified 3D representation  $\mathcal{G}$ . Through rasterization, this hybrid representation is rendered into the final output image  $\hat{I}_d$  achieving photorealistic animation with preserved identity consistency and dynamic details.

$$\{c_k, o_k, s_k, r_k, p_k\}_{static} = \mathcal{B}_{static}(\hat{F}_s) + \{p_k\}_{static}, \quad (3)$$

where  $\{c_k, o_k, s_k, r_k, p_k\}_{static} = \mathcal{G}_{static}$  is the static 3D Gaussian, and  $\mathcal{B}_{static}$  is Static Block of Gaussian Generator.  $\{c_k\}$ ,  $\{o_k\}$ ,  $\{s_k\}$ ,  $\{r_k\}$ ,  $\{p_k\}$  are colors, opacity, scales, rotation and positions of 3D Gaussian respectively. For the dynamic 3D Gaussian component, we formulated as:

$$\{c_k, o_k, s_k, r_k, p_k\}_{dynamic} = \mathcal{B}_{dynamic}(\hat{F}_s) + \mathcal{D}(\alpha_d), \quad (4)$$

Method	Self Reenactment								Cross Reenactment		
	CSIM↑	LPIPS↓	SSIM↑	PSNR↑	FLMD↓	MLMD↓	APC↑	AEC↑	CSIM↑	APC↑	AEC↑
P4D (Deng and others. 2024a)	0.704	0.286	0.676	16.688	2.661	3.806	0.843	0.836	0.361	0.586	0.326
P4D-v2 (Deng and others. 2024b)	0.744	0.262	0.691	17.272	2.408	3.529	0.910	0.897	0.379	0.644	0.383
GPAvatar (Chu and others. 2024b)	0.737	0.246	0.718	18.257	2.223	2.869	0.852	0.835	0.386	0.586	0.418
LivePortrait (Guo and others. 2024)	0.808	0.212	0.733	19.386	1.806	2.611	0.966	<u>0.944</u>	0.381	0.664	0.410
StyleHEAT (Yin and others. 2022)	0.626	0.337	0.655	15.439	2.827	5.071	0.819	0.789	0.313	0.668	0.372
Real3DPortrait (Ye and others. 2024)	0.750	0.251	0.726	18.598	1.967	3.031	0.938	0.875	0.272	0.634	0.378
GAGAvatar (Chu and others. 2024a)	0.826	0.207	0.746	19.473	1.315	1.921	0.963	0.940	0.405	0.713	0.450
RAR	<b>0.836</b>	<u>0.181</u>	<u>0.763</u>	<u>19.738</u>	<u>1.226</u>	<u>1.788</u>	<u>0.973</u>	<u>0.944</u>	<b>0.425</b>	<u>0.738</u>	<u>0.471</u>
RAR (fine-tune)	<u>0.832</u>	<b>0.180</b>	<b>0.773</b>	<b>19.832</b>	<b>1.214</b>	<b>1.709</b>	<b>0.985</b>	<b>0.950</b>	<u>0.415</u>	<b>0.748</b>	<b>0.487</b>

Table 1: Quantitative comparison on the VFHQ dataset. We highlight the **best** and second best results.

Method	Self Reenactment								Cross Reenactment		
	CSIM↑	LPIPS↓	SSIM↑	PSNR↑	FLMD↓	MLMD↓	APC↑	AEC↑	CSIM↑	APC↑	AEC↑
P4D (Deng and others. 2024a)	0.797	0.206	0.729	18.687	1.546	2.183	0.917	0.841	0.399	0.757	0.348
P4D-v2 (Deng and others. 2024b)	0.841	0.187	0.740	18.932	1.283	1.903	0.959	0.906	0.432	0.791	0.379
GPAvatar (Chu and others. 2024b)	0.860	0.164	0.781	20.797	0.993	1.514	0.963	0.892	0.438	0.782	0.397
LivePortrait (Guo and others. 2024)	0.882	0.154	0.780	21.042	1.005	1.406	0.980	0.951	0.438	0.786	0.374
StyleHEAT (Yin and others. 2022)	0.787	0.264	0.683	15.923	1.449	2.416	0.952	0.851	0.424	0.786	0.392
Real3DPortrait (Ye and others. 2024)	0.839	0.183	0.766	20.263	1.139	1.679	0.962	0.902	0.379	0.771	0.355
GAGAvatar (Chu and others. 2024a)	0.878	0.163	0.783	20.822	0.848	1.165	0.979	0.933	0.451	0.807	0.400
RAR	<u>0.884</u>	<u>0.146</u>	<u>0.785</u>	<u>21.084</u>	<u>0.811</u>	<u>1.172</u>	<u>0.987</u>	<u>0.964</u>	<b>0.466</b>	<u>0.817</u>	<u>0.407</u>
RAR (fine-tune)	<b>0.885</b>	<b>0.142</b>	<b>0.789</b>	<b>21.107</b>	<b>0.808</b>	<b>1.153</b>	<b>0.989</b>	<b>0.977</b>	<u>0.460</u>	<b>0.826</b>	<b>0.427</b>

Table 2: Quantitative comparison on the HDTF dataset. We highlight the **best** and second best results.

where  $\{c_k, o_k, s_k, r_k, p_k\}_{dynamic} = \mathcal{G}_{dynamic}$  is the dynamic 3D Gaussian.  $\mathcal{B}_{dynamic}$  is Dynamic Block of Gaussian Generator, and  $\mathcal{D}$  is reenactment module, which will be introduced in the following section. Finally, by concatenate  $\mathcal{G}_{static}$  and  $\mathcal{G}_{dynamic}$ , we got the completed 3D Gaussian  $\mathcal{G} = [\mathcal{G}_{static}, \mathcal{G}_{dynamic}]$  of avatar.

### Reenactment Module

In this subsection we give more details for reenactment module. The reenactment module first employs several layers of MLP to upscale the input control parameter vector  $\alpha_d$ , and then reshapes it into a 2D feature map. Next, it performs further feature extraction via 2D convolution and predicts the positions of the dynamic 3D Gaussian. During inference, following the one-shot reconstruction from source image  $I_s$ , all subsequent expression reenactments are processed solely through the reenactment module. Due to its extremely lightweight design, the reenactment module enables highly efficient synthesis and rendering.

### Texture Restoration Module

Most 3D methods estimate pose and scale from monocular videos/images, causing camera pose errors and high-frequency texture artifacts. In contrast, 2D methods use a warping field for a pseudo-3D transformation that visually approximates true 3D motion, their simpler motion patterns are easier to learn. To address 3D artifacts, we employed a lightweight texture restoration module after the 3D Gaussian Splatting stage, using a StyleGAN2 Generator with SFT modulation. Following the Live Portrait (Guo and others.

2024) and GFPGAN (Wang and others. 2021b) approaches, we synthesized about 70K training samples using FFHQ as facial appearance and VFHQ as the driving video.

### Training Strategy and Loss Functions

Our training process is divided into two stages: global pretraining and fine-tuning for texture restoration module. Global pretraining employs VFHQ as the training dataset. For each training sample, two video frames are randomly selected from the same video sequence to serve respectively as the source image and the drive image. The corresponding driving frame control parameters  $\alpha_d$  and the source image  $I_s$  are fed into the reconstruction module to predict the sync image  $\hat{I}_d$ . Since the source image and drive image belong to the same person, the drive image can be used as the target image to provide pixel-level supervision for the sync image  $\hat{I}_d$ . The loss function for global training mainly consists of a perceptual loss  $L_{lips}$  and an loss  $L_1 = \|\hat{I}_d - I_d\|_1$ .

$$L_{pretraining} = \lambda_1 L_{lips} + \lambda_2 L_1, \quad (5)$$

Where  $\lambda_1 = 0.01$  and  $\lambda_2 = 1$ .

During texture restoration module fine-tuning, we freeze all parameters except those of the texture restoration module and train on 70K synthetic data samples. The fine-tuning loss is also composed of a perceptual loss and an  $L_1$  loss, with additional supervision loss functions  $L_{eye\_teeth}$  applied to specific regions such as the eyes and teeth. Specifically, the  $L_{eye\_teeth}$  are implemented as follows. First, the eye and mouth bounding box regions in  $I_d$  are estimated using the landmarks calculated by the facial landmarks detector. Then,

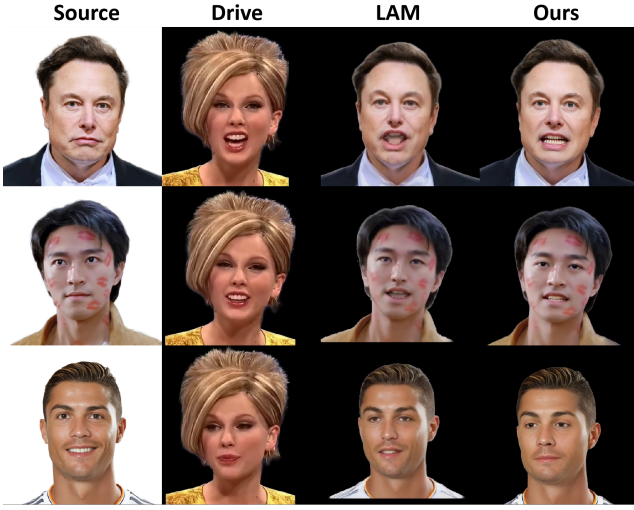


Figure 4: Visual comparison between LAM(He and others. 2025) and RAR.

the perceptual loss for the eye and mouth regions is calculated separately and summed.

$$L_{finetune} = \lambda_1 L_{lips} + \lambda_2 L_1 + \lambda_3 L_{eye\_teeth} \quad (6)$$

Where  $\lambda_1$  and  $\lambda_2$  keep consistent values as before, and  $\lambda_3 = 0.05$ .

## Experiment

### Data Preprocessing.

For both training and inference data, we adopt a consistent preprocessing procedure. First, the background is removed and the foreground subject is preserved by applying a matting algorithm. Subsequently, the 3D landmarks are estimated and utilized alongside a predefined standard model to compute the head pose. Thereafter, the camera pose relative to the standard model is derived from the computed head pose.

### Experimental Settings.

In global training, VFHQ is used as the training set. For each video frame sequence, one frame is sampled at an interval of five frames, forming a sampled subset as the training data. In total, over 600,000 frames were extracted from more than 15,000 videos. During fine-tuning of the restoration module, a subset of 7K person identity data is first filtered from VFHQ. Then, for each identity, 10 videos are randomly selected from VFHQ as driving videos. Finally, based on Live Portrait (Guo and others. 2024) and GFPGAN (Wang and others. 2021b), 70K video clips were generated, and one frame was sampled every five frames for a clip, yielding a final total of 1.5 million frames. All training data were resized to 512×512 resolution. The test data consist of 50 videos from the original test split in VFHQ, as well as all 365 videos from the HDTF dataset.

### Implementation Details.

We implemented the entire experiment using PyTorch with 8 H200 GPUs. In global training we set a global batch size



Figure 5: With texture restoration module and synthetic data, more detailed textures in the tooth region can be reconstructed

of 128 and ran training for 500K iterations. The learning rate was  $1e-4$ , and we used the ADAM optimizer. Next, we fine-tuned the restoration module. We kept the batch size and optimizer unchanged. We set the learning rate to  $5e-5$ . Regarding the weighting coefficients of the loss functions, the perceptual loss is weighted at 0.01 and the L1 loss at 1. Additionally, the loss function for specific regions is weighted at 0.05.

### Evaluation Metrics.

In the evaluation process, we focus on performance metrics for self-identity reenactment and cross-identity reenactment. For self-reenactment, the drive images serve as the ground truth, establishing a supervised evaluation. We assess the quality of synthesis along three dimensions: image quality, identity similarity, and expression/pose similarity. Regarding image quality, we adopt three quantitative metrics: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS). These metrics enable an effective comparison between synthesized and drive images. To evaluate identity similarity (CSIM), we calculate the cosine distance of face recognition features based on the methodology of (Deng and others. 2024a). For assessing the realism of expressions and poses, we use the average expression cosine distance (AEC) and average pose cosine distance (APC), as determined by a 3D Morphable Model (3DMM) estimator. Additionally, we employ the facial keypoint detector introduced to measure the facial landmark mean distance (FLMD) and the mouth landmark mean distance (MLMD), which offer deeper insights into the accuracy of the driving controls in our animation. In the case of cross-identity reenactment, ground truth are not available. Therefore, we rely solely on CSIM, AEC, and APC for evaluation. Except for minor changes in the AEC and APC calculation process, these metrics are consistent with those used in previous studies (Deng and others.



Backbone	Params	Self Reenactment								Cross Reenactment		
		CSIM↑	LPIPS↓	SSIM↑	PSNR↑	FLMD↓	MLMD↓	APC↑	AEC↑	CSIM↑	APC↑	AEC↑
Dinov2-vitb14	0.1B	0.879	0.151	0.771	20.826	0.833	1.209	0.981	0.953	0.456	0.810	0.388
Dinov2-vitg14	1.1B	0.881	0.151	0.774	20.932	0.829	1.173	0.982	0.959	0.456	0.810	0.397
Webssl-dino7b-full8b-518	7.0B	<b>0.884</b>	<b>0.146</b>	<b>0.785</b>	<b>21.084</b>	<b>0.811</b>	<b>1.172</b>	<b>0.987</b>	<b>0.964</b>	<b>0.466</b>	<b>0.817</b>	<b>0.407</b>

Table 3: Quantitative comparison of different backbone on HDTF. We highlight the **best** and second best results.

Method	StyleHeat	Real3D	P4D	P4D-v2	LivePortrait	GAGAvatar	LAM	RAR
FPS	19.82	4.55	9.51	9.62	77	67.12	<b>280.96</b>	90

Table 4: Running time of driving and rendering measured in FPS and tested on A100 GPU with  $512 \times 512$  resolution outputs. All results exclude the time for reconstruction and driving parameters estimation which can be calculated in advance. We highlight the **best** and second best results.

2024a,b; Chu and others. 2024a,b; He and others. 2025).

## Results

### Quantitative Results.

We evaluated RAR on the VFHQ and HDTF datasets. The results on these datasets are presented in Tab.1 and Tab.2, respectively. As shown in the tables, our method achieved the best image reconstruction quality, as evidenced by the PSNR, SSIM, and LPIPS metrics. Additionally, we maintained strong identity consistency, which is reflected in the CSIM metric. Moreover, we obtained accurate expressions and poses consistent with the driving image, as indicated by the FLMD, MLMD, AEC and APC metrics.

### Qualitative Results.

Fig.3 compares the cross reenactment results of RAR with other approaches on both the VFHQ and HDTF datasets. Compared to previous methods, our approach achieves superior reconstruction details in texture, better preserves identity consistency, and expressions and poses more aligned with the driving image. It is also evident that after fine-tuning with texture repair, our model shows significant improvements in high-frequency texture regions (such as teeth and eyes).

### Inference Speed.

During the inference phase, latency is primarily composed of two parts: drive and render. As shown in Fig.2, the modules involved in the drive and render stages are indicated by dashed boxes. The drive portion consists solely of the reenactment module, which involves only a few MLP and Conv2D operations, resulting in a negligibly short running time of less than 1 ms. The main inference time of RAR is concentrated in the render stage. Benefiting from the learnability of synthetic data, and with reference to StyleGAN2 (Karras and others. 2020), we stacked three SFT modulations to implement a lightweight texture completion module. This configuration allows the entire render stage to run in approximately 9 ms, while ensuring high-quality reconstruction of high-frequency textures. Ultimately, the combined inference speed of the drive and render stages reaches 90 FPS. In Tab.4, we compare RAR with several previous algorithms. The inference speed of RAR is second only to LAM,

however our method can achieve significantly superior visual results as shown in Fig.4. Additionally since LAM did not provide a complete custom testing sample creation process, we could only align with the test cases provided by LAM for comparison.

Control Priors	CSIM↑	APC↑	AEC↑
FLAME (Li and others. 2017)	0.451	0.801	0.399
PDFGC (Wang and others. 2023)	<b>0.466</b>	<b>0.817</b>	<b>0.407</b>

Table 5: Quantitative comparison of different control priors with cross-identity reenactment setting on HDTF. We highlight the **best** and second best results.

## Ablation Studies

### Pre-trained Backbone.

Based on the HDTF test sets, we compared the impact of different-scale models used as backbones in our method. As shown in Tab.3, as the scale of the pre-trained backbone increases, the performance of our algorithm correspondingly improves. Although our current backbone is based on Webssl-dino7b-full8b-518 (Fan and others. 2025), the architecture which decouples reconstruction and reenactment can theoretically support larger-scale pre-trained models as backbones, conforming to the performance enhancement indicated by the scaling law (Kaplan and others. 2020) while still maintaining the current high driving efficiency.

### Texture Restoration.

We compared the visual results to evaluate the effect differences caused by incorporating a texture restoration module and training with synthetic data. As shown in Fig.5, after fine-tuning, more detailed textures in the tooth region can be reconstructed. This confirms the effectiveness of texture restoration.

### Comparison of Control Priors

We conducted a comparative experiment to show that using PDFGC (Wang and others. 2023) as the control prior is better than using FLAME. We kept the model architecture and training method the same. We trained two models: one with PDFGC as the control prior and one with FLAME. We then evaluated the cross reenactment performance on the HDTF

datasets. Table 5 shows that the PDFGC-based approach performs better than the FLAME-based method according to the evaluation metrics.

## References

- Baevski, A.; and others. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *NeurIPS*.
- Bai, Y.; and others. 2023. High-fidelity Facial Avatar Reconstruction from Monocular Video with Generative Priors. in *CVPR*.
- Blanz, V.; and others. 1999. A Morphable Model for the Synthesis of 3D Faces. in *SIGGRAPH*.
- Burkov, E.; and others. 2020. Neural Head Reenactment with Latent Pose Descriptors. in *CVPR*.
- Chen, Y.; and others. 2025. HunyuanVideo-Avatar: High-Fidelity Audio-Driven Human Animation for Multiple Characters. in *ArXiv*.
- Chu, X.; and others. 2024a. Generalizable and Animatable Gaussian Head Avatar. in *NeurIPS*.
- Chu, X.; and others. 2024b. GPAvatar: Generalizable and Precise Head Avatar from Image(s). in *ICLR*.
- Cui, J.; and others. 2024. Hallo2: Long-Duration and High-Resolution Audio-Driven Portrait Image Animation. in *CoRR*.
- Deng, Y.; and others. 2024a. Portrait4D: Learning One-Shot 4D Head Avatar Synthesis using Synthetic Data. in *CVPR*.
- Deng, Y.; and others. 2024b. Portrait4D-v2: Pseudo Multi-View Data Creates Better 4D Head Synthesizer. in *ArXiv*.
- Fan, D.; and others. 2025. Scaling language-free visual representation learning. in *ArXiv*.
- Gafni, G.; and others. 2021. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. in *CVPR*.
- Gerig, T.; and others. 2018. Morphable Face Models - An Open Framework. in *FG*.
- Goodfellow, I. J.; and others. 2014. Generative Adversarial Nets. in *NeurIPS*.
- Guo, J.; and others. 2024. LivePortrait: Efficient Portrait Animation with Stitching and Retargeting Control. in *CoRR*.
- Han, H.; and others. 2024. CVTHead: One-shot Controllable Head Avatar with Vertex-feature Transformer. in *WACV*.
- He, Y.; and others. 2025. LAM: Large Avatar Model for One-shot Animatable Gaussian Head. in *ArXiv*.
- Hong, F.-T.; and others. 2022. Depth-Aware Generative Adversarial Network for Talking Head Video Generation. in *CVPR*.
- Isola, P.; and others. 2017. Image-to-Image Translation with Conditional Adversarial Networks. in *CVPR*.
- Kaplan, J.; and others. 2020. Scaling laws for neural language models. in *ArXiv*.
- Karras, T.; and others. 2019. A Style-Based Generator Architecture for Generative Adversarial Networks. in *CVPR*.
- Karras, T.; and others. 2020. Analyzing and Improving the Image Quality of StyleGAN. in *CVPR*.
- Kerbl, B.; and others. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. in *ACM TOG*.
- Khakhulin, T.; and others. 2022. Realistic One-Shot Mesh-Based Head Avatars. in *ECCV*.
- Li, T.; and others. 2017. Learning a model of facial shape and expression from 4D scans. in *ACM TOG*.
- Li, X.; and others. 2023. Generalizable One-shot 3D Neural Head Avatar. in *NeurIPS*.
- Mildenhall, B.; and others. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. in *ECCV*.
- Oquab, M.; and others. 2024. DINOv2: Learning Robust Visual Features without Supervision. in *TMLR*.
- Paysan, P.; and others. 2009. A 3D Face Model for Pose and Illumination Invariant Face Recognition. in *AVSS*.
- Prajwal, K.; and others. 2020. A lip sync expert is all you need for speech to lip generation in the wild. In *ACM MM*.
- Qian, S.; and others. 2024. GaussianAvatars: Photorealistic Head Avatars with Rigged 3D Gaussians. in *CVPR*.
- Sun, J.; and others. 2023. Next3D: Generative Neural Texture Rasterization for 3D-Aware Head Avatars. in *CVPR*.
- Sun, K.; and others. 2024. CGOF++: Controllable 3D Face Synthesis With Conditional Generative Occupancy Fields. in *IEEE Trans. Pattern Anal. Mach. Intell.*
- Tang, J.; and others. 2024. 3DFaceShop: Explicitly Controllable 3D-Aware Portrait Generation. in *TVCG*.
- Tian, L.; and others. 2024. EMO: Emote Portrait Alive Generating Expressive Portrait Videos with Audio2Video Diffusion Model Under Weak Conditions. in *ECCV*.
- Wang, D.; and others. 2023. Progressive Disentangled Representation Learning for Fine-Grained Controllable Talking Head Synthesis. in *CVPR*.
- Wang, T.-C.; and others. 2021a. One-Shot Free-View Neural Talking-Head Synthesis for Video Conferencing. in *CVPR*.
- Wang, X.; and others. 2021b. Towards real-world blind face restoration with generative facial prior. in *CVPR*, 9168–9178.
- Xu, E. Z.; and others. 2023. PV3D: A 3D Generative Model for Portrait Video Generation. in *ICLR*.
- Xu, S.; and others. 2020. Deep 3D Portrait From a Single Image. in *CVPR*.
- Xu, Y.; and others. 2024. Gaussian Head Avatar: Ultra High-Fidelity Head Avatar via Dynamic Gaussians. in *CVPR*.
- Ye, Z.; and others. 2024. Real3d-portrait: One-shot realistic 3d talking portrait synthesis. in *ArXiv*.
- Ye, Z.; and others. 2025. Realistic Real-Time Talking Head Synthesis with Grid Encoding and Progressive Conditioning. in *ICASSP*.
- Yin, F.; and others. 2022. StyleHEAT: One-Shot High-Resolution Editable Talking Face Generation via Pre-trained StyleGAN. in *ECCV*.

Yu, W.; and others. 2023. NOFA: NeRF-based One-shot Facial Avatar Reconstruction. *in SIGGRAPH*.

Zakharov, E.; and others. 2019. Few-Shot Adversarial Learning of Realistic Neural Talking Head Models. *in ICCV*.

Zhang, B.; and others. 2023. MetaPortrait: Identity-Preserving Talking Head Generation with Fast Personalized Adaptation. *in CVPR*.

Zhang, Y.; and others. 2024. Musetalk: Real-time high quality lip synchronization with latent space inpainting. *in ArXiv*.

Zheng, Y.; and others. 2023. PointAvatar: Deformable Point-Based Head Avatars from Videos. *in CVPR*.

Zhuang, P.; and others. 2022. Controllable Radiance Fields for Dynamic Face Synthesis. *in 3DV*.

Zielonka, W.; and others. 2023. Instant Volumetric Head Avatars. *in CVPR*.

## Supplementary

### A. Audio Driven

Our proposed method, RAR, enables independent control of facial features such as the eyes and mouth in a single-frame portrait. To further validate the excellent control capability and expand the application scope of our approach, we designed an audio2motion (audio to motion) model that maps the input audio to the PDFGC mouth parameters. The mouth parameters generated by audio2motion control the mouth shape of a 3D Gaussian head, ultimately realizing speech-driven lip synchronization of the 3D Gaussian head.

Specifically, the audio2motion model is implemented as a modular neural network that maps an audio sequence to a motion sequence. The architecture mainly consists of three components. First, a pre-trained Wav2Vec2.0 (Baevski and others. 2020) model is employed as the audio encoder, with its feature extraction layers frozen. The contextual output is then projected to a configurable hidden dimension via a linear layer. Second, the intermediate audio features are further processed by an audio encoder module, which optionally incorporates identity or category information to facilitate personalized modeling. Third, in the decoding stage, the architecture utilizes multiple sequentially arranged ConvNorm-Relu blocks. The block comprises a 1D convolution, layer normalization, and ReLU activation, with a final 1D convolution mapping the hidden state to the output dimension.

During training, we collected a dataset comprising 1,000 professional single-speaker lecture videos. From each frame, mouth features were extracted through PDFGC, and the corresponding audio segments served as inputs to constitute paired training samples. This dataset was subsequently used to train the audio2motion model which employed a mean square error (MSE) loss to constrain the predicts to the ground-truth.

Furthermore, quantitative experiments verified that our approach, when extended to audio-driven tasks, achieves highly competitive results. We randomly selected 50 task IDs and 50 audio clips from HDTF for cross-driving experiments. The accuracy of the lip synchronization was measured using the sync score (Prajwal and others. 2020), while

Method	Sync Score $\uparrow$	CSIM $\uparrow$
MuseTalk	6.33	0.302
Wav2Lip	<b>7.58</b>	0.306
HunyuanVideoAvatar	<u>7.34</u>	<u>0.336</u>
RAR	6.52	<b>0.351</b>

Table 6: **Quantitative Results of Audio-driven.** We highlight the **best** and second best results.

identity consistency was evaluated using the CSIM metric. For comparison, we selected Wav2Lip, MuseTalk, and HunyuanVideoAvatar as baseline methods. As shown in Table 6, detailed comparative results are provided: Wav2Lip (Prajwal and others. 2020), which directly uses the sync score as a supervisory loss, achieved the highest lip accuracy score. HunyuanVideoAvatar (Chen and others. 2025), benefiting from large-scale parameters and data, attained a sync score second only to that of Wav2Lip. And our method achieved a sync score slightly better than musetalk (Zhang and others. 2024), though lower than both Wav2Lip and HunyuanVideoAvatar. On the other hand, unlike other methods, our approach, benefiting from enhanced 3D priors, performed best in terms of the CSIM metric. Therefore, RAR can be directly extended to audio-driven tasks without additional training while achieving highly competitive results, further validating its strong independent control over facial features in a single-frame portrait.

### B. Comparison Methods

We carefully selected state-of-the-art approaches for comparison, including 2D end-to-end methods such as StyleHeat (Yin and others. 2022) and LivePortrait (Guo and others. 2024), as well as methods based on 3D prior structures such as CVTHead (Han and others. 2024), GPAvatar (Chu and others. 2024b), Real3DPortrait (Ye and others. 2024), Portrait4D (Deng and others. 2024a), Portrait4D-v2 (Deng and others. 2024b), and GAGAvatar (Chu and others. 2024a). For each method, we reproduced the results using the official open-source projects and models.

### C. Structure Details

In this part, we introduce more details of RAR.

**Feature Extraction Module.** The feature extraction module first leverages a pre-trained WebSSL model to extract embeddings at varying granularities. These embeddings are subsequently reshaped into 2D feature maps. After processing through convolutional layers (Conv2D), the multi-granularity feature maps are concatenated along the channel dimension. A final Conv2d layer then integrates these concatenated features, yielding a fused feature map. Additionally, the embedding from the final layer of the WebSSL model is selected as the global feature. The more intuitive process is illustrated in the Fig.6.

**Canonicalization Block.** The Canonicalization Block achieves expression neutralized feature on the source image by leveraging the source motion. The fused feature is initially partitioned into patches. Each patch is reshaped into

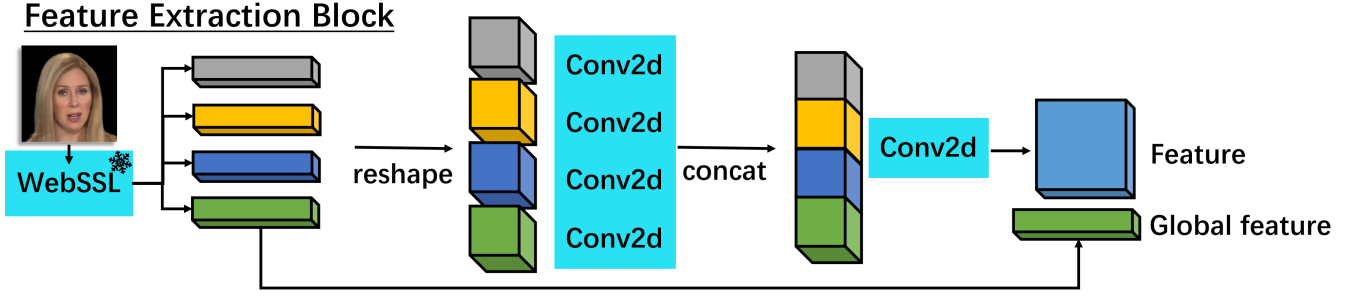


Figure 6: The module uses a pre-trained WebSSL model to extract multi-granularity embeddings, reshaping them into 2D maps. After Conv2D processing, these maps are concatenated and fused by a final Conv2D layer, while the last WebSSL layer provides the global feature.

### Canonicalization Block

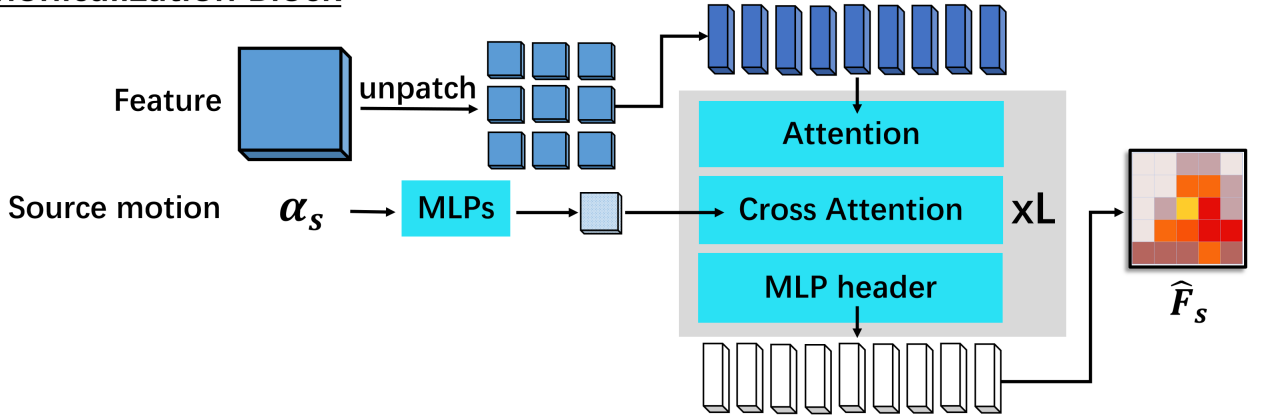


Figure 7: Canonicalization Block neutralizes the expression of features on the source image via source motion. The fused features are split into patches, reshaped into 1D token sequences, and processed through  $L$  transformer blocks. Finally, tokens are rearranged to reconstruct the canonicalized feature map, with the source motion vector transformed by a multi-layer MLP and injected as conditioning for Cross-Attention.

a 1D token sequence. These tokens are processed through  $L$  stacked transformer blocks, each comprising Multi-Head Attention, Cross-Attention, and a Multilayer Perceptron (MLP) head layer. Finally, the processed tokens are rearranged (unpatched) to reconstruct the canonicalized feature map. Crucially, the source motion vector undergoes transformation via a multi-layer MLP projection and is injected as the conditioning input for the Cross-Attention within this process. For more intuitive process can be found in the Fig.7.

**Gaussian Generator and Reenactment Module.** The Gaussian Generator output consists of two parts: the static Gaussian and the dynamic Gaussian. Regarding the static Gaussian, its positions are directly composed of the 3DMM vertices. Other appearance information is obtained by concatenating the global feature and the mesh embedding of the 3DMM vertices and predicting them through multiple layers of MLP. Regarding the dynamic Gaussian, its appearance information is predicted by the canonicalized feature  $\hat{F}_s$  after Conv2D processing. The positions of the dynamic Gaussian are obtained by mapping the drive motion  $\alpha_d$  through MLPs,

reshape, Conv2D and other operations. After the static Gaussian and the dynamic Gaussian are merged together, the final result is a model that represents the controllable 3D Gaussian corresponding to the source portrait image. More details can be found in Fig.8.

### D. More Visualization Results

In this section, we provide a comprehensive array of additional visualization results that serve to further illustrate our findings. These results include samples from the VFHQ and HDTF test sets, where we apply both self-reenactment and cross-reenactment settings to ensure a thorough examination of our approach under varied conditions. Specifically, by employing a strategic random sampling method, we capture a diverse range of instances that reveal the subtle intricacies in the data, from variations in texture and lighting to the delicate interplay of facial expressions. This process not only highlights the robustness of our technique but also demonstrates its capability to maintain fidelity and realism across different reenactment modalities. You can find all these



## Gaussian Generator

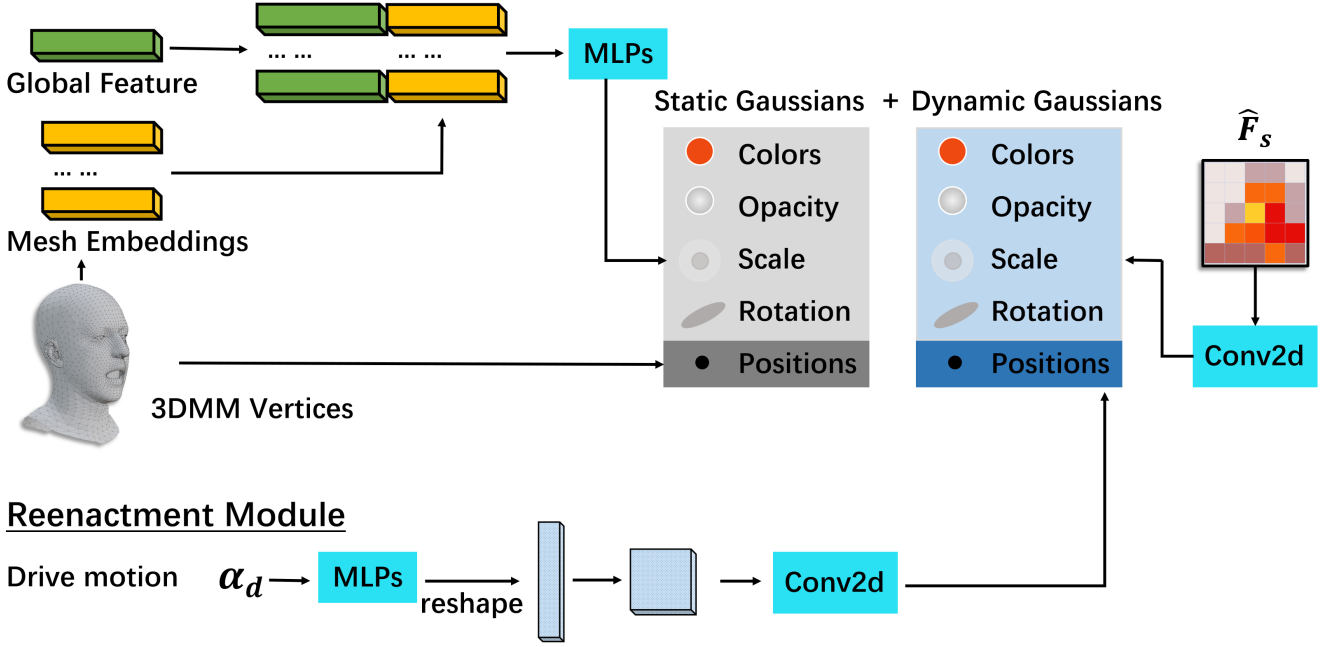


Figure 8: The Gaussian Generator produces two parts: the static and dynamic Gaussians. The static Gaussian uses 3DMM vertices for positions, while its appearance is predicted by concatenating a global feature with a mesh embedding of those vertices and processing them through MLP layers. The dynamic Gaussian’s appearance is predicted from the canonicalized feature  $\hat{F}_s$  after Conv2D processing, and its positions are derived from the drive motion  $\alpha_d$  using MLPs, reshaping, and Conv2D operations. Merging both produces a controllable 3D Gaussian for the source portrait.

detailed visual examples, complete with annotations and supplementary insights, in the subsequent Figs.9,10,11,12, where each image has been carefully presented to help you gain a deeper understanding of the underlying performance.

and human experiences are more seamlessly integrated.

## E. Ethical Consideration

Our innovative technique can synthesize high-fidelity, real-time talking head video using a single portrait image, leveraging advanced algorithms that capture intricate facial expressions and subtle nuances in movement. By integrating state-of-the-art machine learning models with robust data training over diverse datasets, we ensure that every generated video exudes lifelike clarity and realism. We aspire to extend the application of this pioneering approach to a broad array of endeavors that are not only technically groundbreaking but also significantly advantageous for society—improving remote communications, enhancing digital education, and fostering more inclusive virtual interactions.

As part of our unwavering commitment to ethical advancement in technology, we are enthusiastic about offering dedicated support and practical assistance to the deepfake detection community, which is essential in upholding digital integrity and security. We believe that with careful and judicious application, our method can serve as a catalyst for the wholesome progression of digital human technology by setting new benchmarks in authenticity and efficiency. Ultimately, our work paves the way for transformative innovations across sectors such as entertainment, telehealth, and interactive virtual reality, promising a future where digital

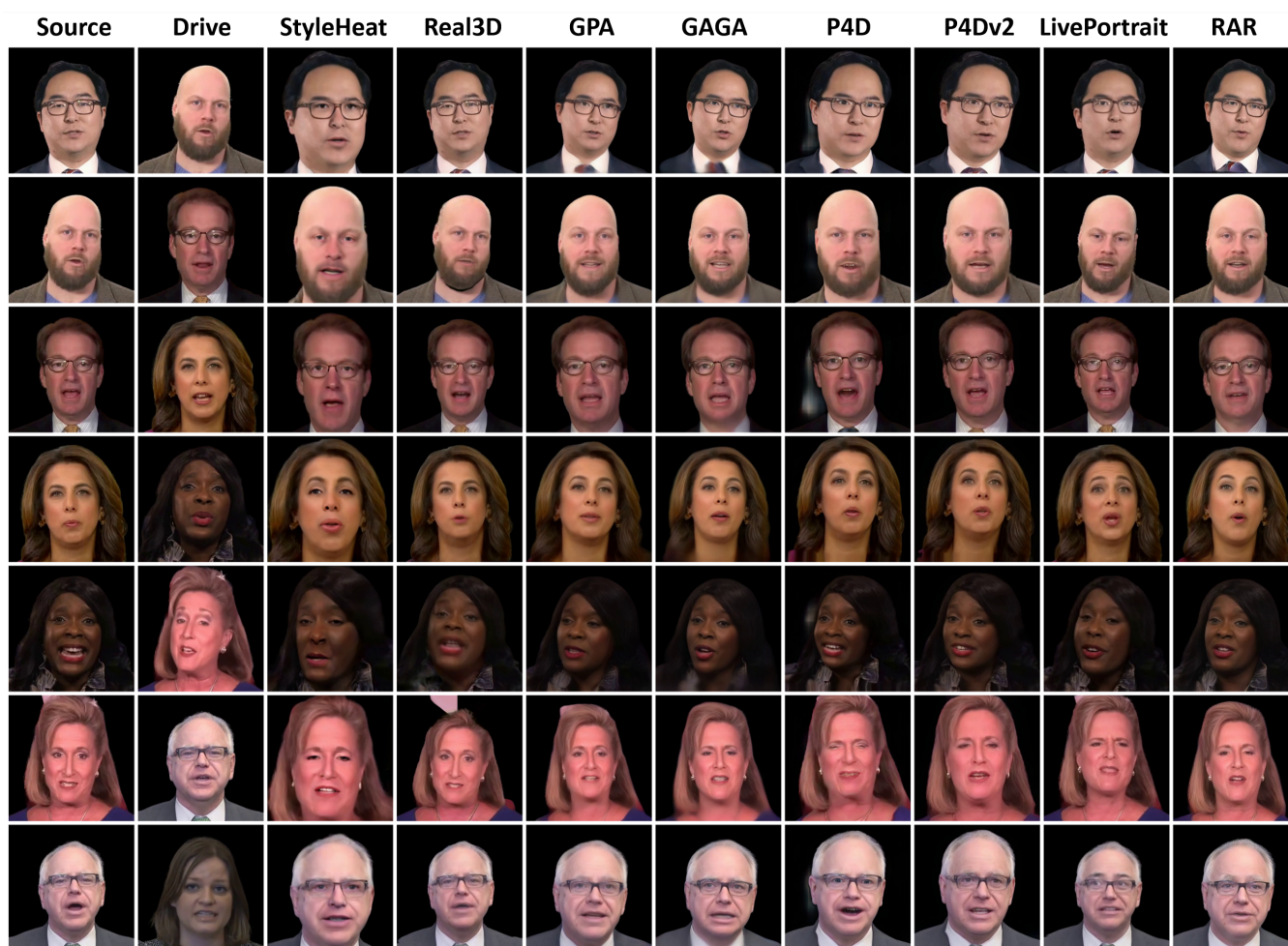


Figure 9: visualization of cross-reenacted results on the HDTF dataset.

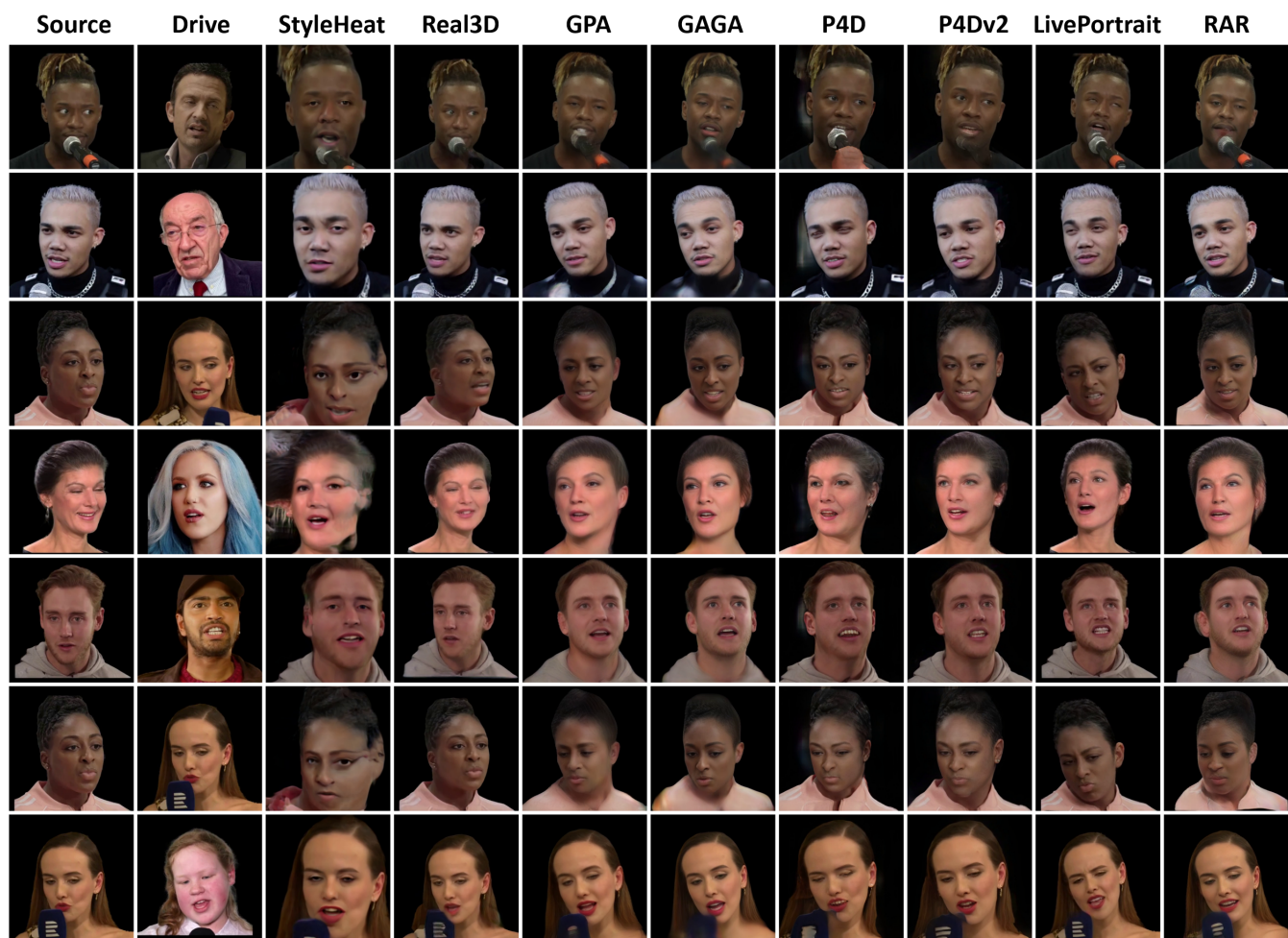


Figure 10: visualization of cross-reenacted results on the VFHQ dataset.



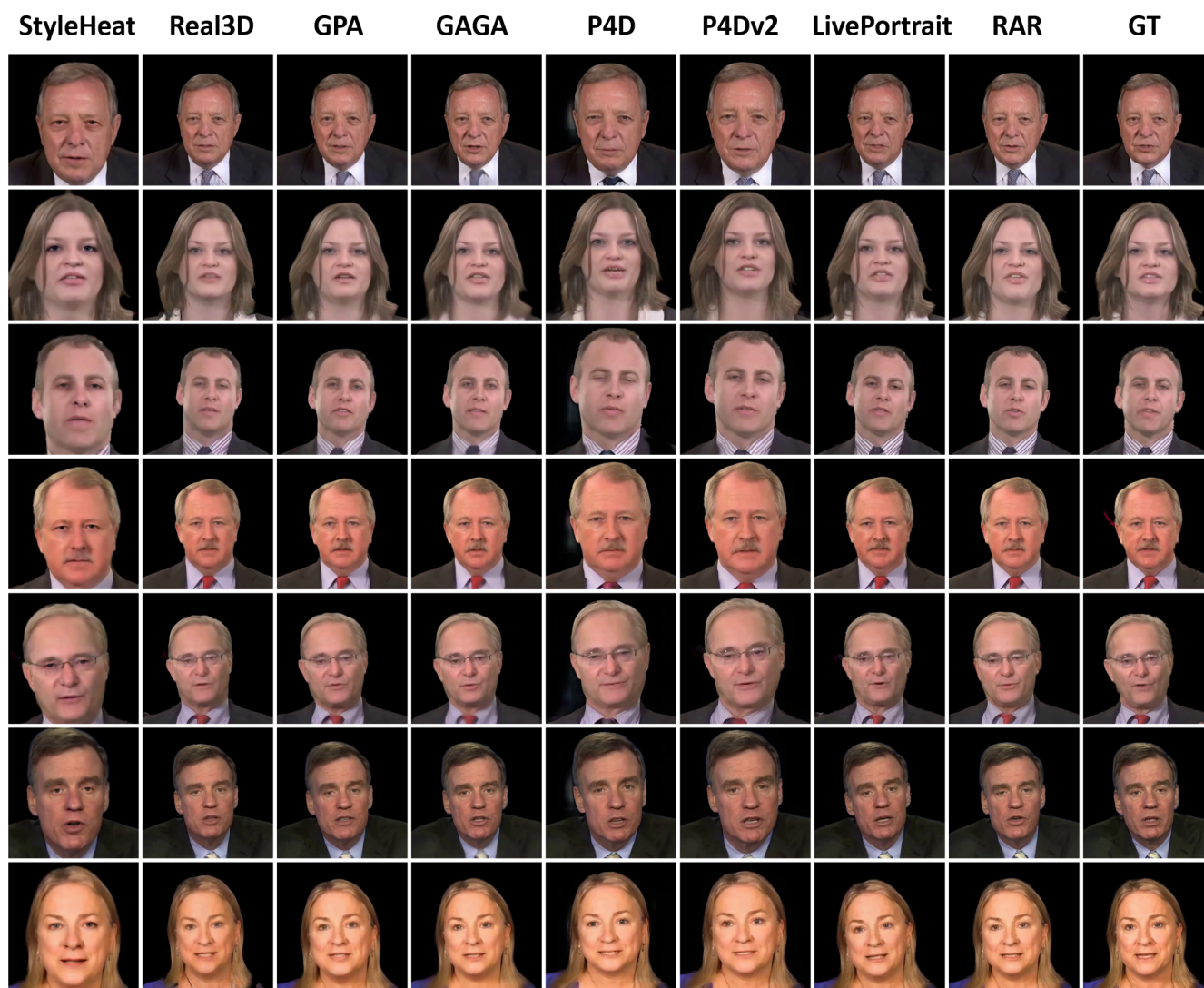


Figure 11: visualization of self-reenacted results on the HDTF dataset.



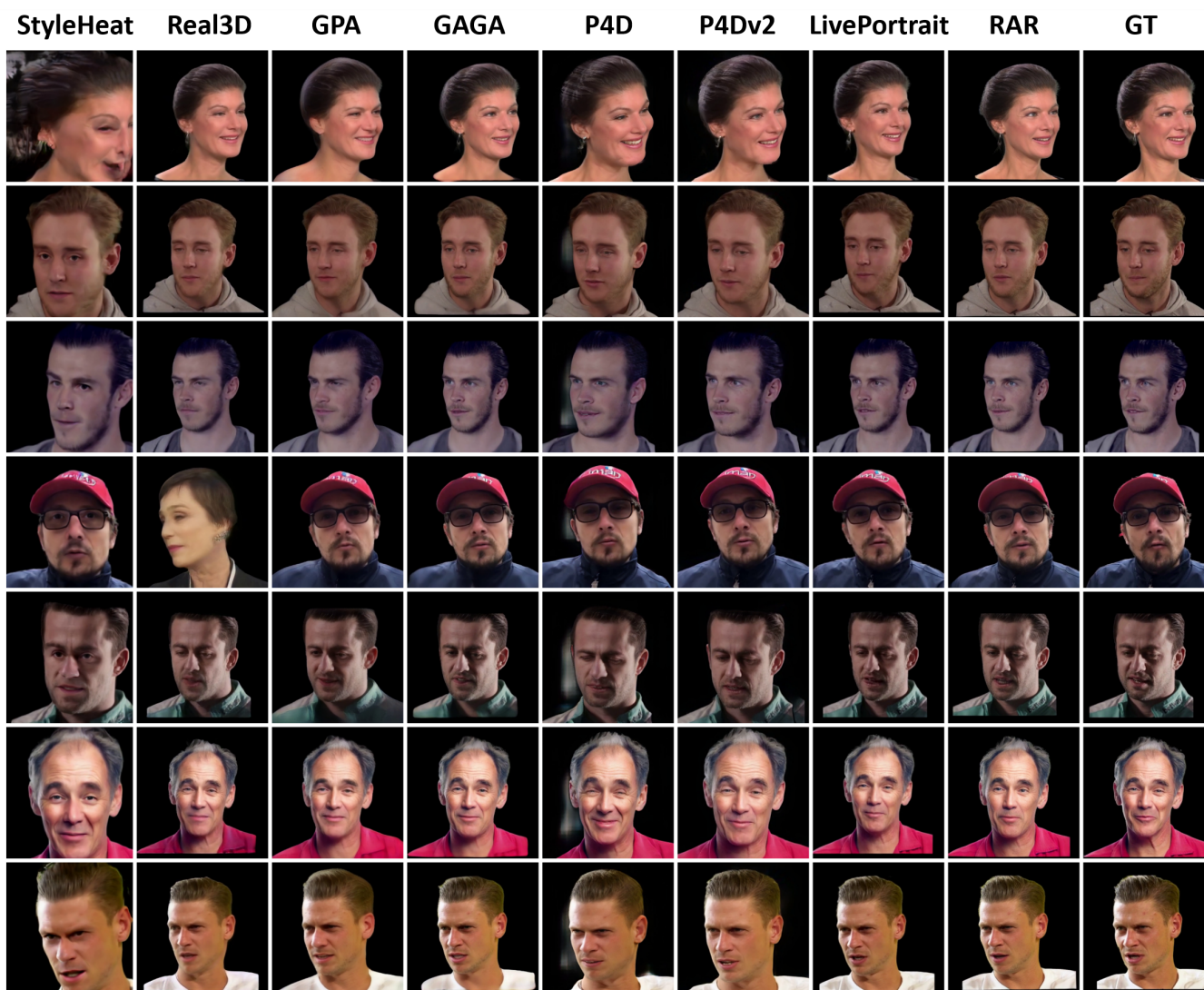


Figure 12: visualization of self-reenacted results on the VFHQ dataset.