HFM-GS: half-face mapping 3DGS avatar based real-time HMD removal

Kangyu Wang, Jian Wu, Runze Fan, Hongwen Zhang, Sio Kei Im, and Lili Wang

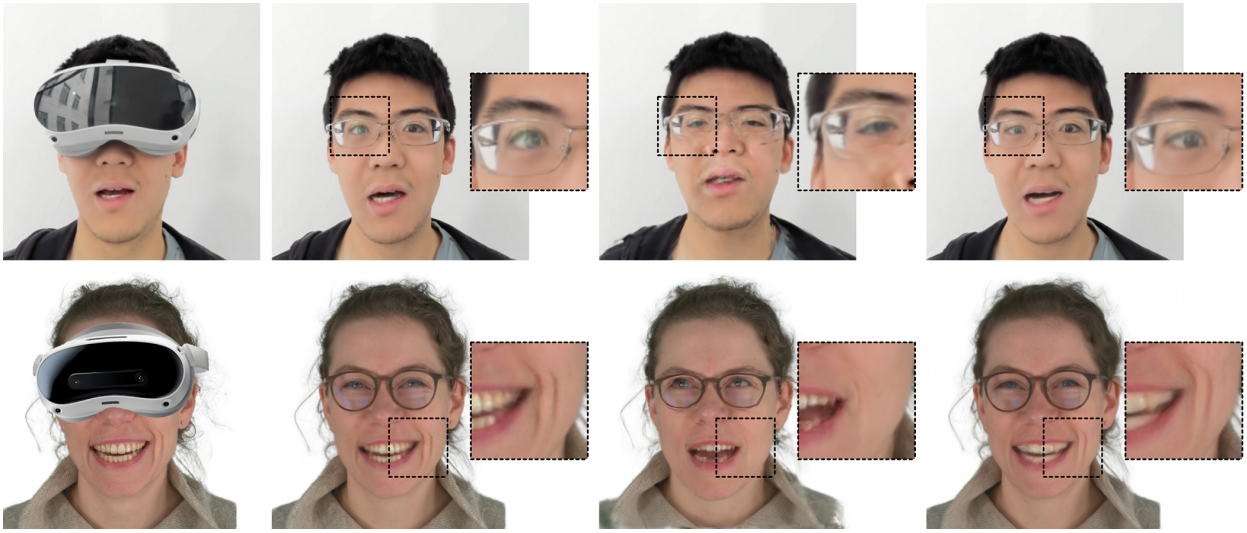

Fig. 1: Given facial video frames occluded by an XR Head Mounted Device (first column), our method, HFM-GS (second column), achieves higher-quality occlusion removal compared to DECA+GaussianAvatars [7, 27] (third column) and LatentAvatar [34] (fourth column) on both collected volunteer HMD occlusion data and the Nersemble dataset [17]. Moreover, our method operates in real-time at 134 FPS. In the first row, our method better preserves the specular highlights on the glasses and the shape of the frames and eyebrows. In the second row, our method more accurately reconstructs tooth contours and fine-grained facial wrinkles.

Abstract— In extended reality (XR) applications, enhancing user perception often necessitates head-mounted display (HMD) removal. However, existing methods suffer from low time performance and suboptimal reconstruction quality. In this paper, we propose a half face mapping 3D Gaussian splatting avatar based HMD removal method (HFM-GS), which can perform real-time and high-fidelity online restoration of the complete face in HMD-occluded videos for XR applications after a short un-occluded face registration. We establish a mapping field between the upper and lower face Gaussians to enhance the adaptability to deformation. Then, we introduce correlation weight-based sampling to improve time performance and handle variations in the number of Gaussians. At last, we ensure model robustness through Gaussian Segregation Strategy. Compared to two state-of-the-art methods, our method achieves better quality and time performance. The results of the user study show that fidelity is significantly improved with our method.

Index Terms—Virtual reality, HMD removal, 3D Gaussian avatar, Deformation field

1 Introduction

Extended Reality (XR) technologies, including Virtual Reality (VR) and Augmented Reality (AR), have demonstrated significant potential across various domains such as education, medical simulation, and social interaction. In many XR applications, it is crucial to perceive the full facial expressions of interaction partners to enhance communication and understanding, particularly in scenarios like remote teaching and conferencing. For example, in a virtual classroom, teachers and students wearing head-mounted displays (HMDs) still wish to see each other's complete faces and expressions to better perceive one another's state. However, HMDs, which serve as essential devices connecting users to virtual content, inevitably occlude a large portion of the upper face. This obstruction severely undermines the natural face-to-face interaction experience in XR applications.

Various HMD removal methods have been proposed to generate complete unoccluded facial images or videos from HMD-occluded inputs. Existing methods can be broadly categorized into image-based methods [3, 11, 12] and geometry-based reconstruction methods [8, 28, 37]. Image-based methods, such as those utilizing Generative Adversarial Networks (GANs), synthesize the missing facial regions. However, these methods often lack real-time performance and facial consistency and fail to provide the multi-view coherence required for VR applications. Geometry-based methods reconstruct and animate

- *Lili Wang is the corresponding author.*
- *Lili Wang, Kangyu Wang, Jian Wu, and Runze Fan are with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing, China. E-mail: wanglily@buaa.edu.cn, 1362483479@qq.com, lanayawj@buaa.edu.cn, by2106131@buaa.edu.cn.*
- *Hongwen Zhang is with the School of Artificial Intelligence, Beijing Normal University, Beijing, China. E-mail: zhanghongwen@bnu.edu.cn*
- *Sio Kei Im is with Macao Polytechnic University. E-mail: marcusim@mpu.edu.mo.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

the face using 3D models, e.g. Zhao et al. [37] employs large-scale point clouds. However, these methods typically rely on specialized hardware and face challenges in balancing reconstruction quality and computational efficiency. In recent years, Neural Radiance Fields (NeRF) [23] and 3D Gaussian Splatting (3DGS) [15] have offered new insights into facial reconstruction and animation [9, 27, 33, 34], but no existing methods have integrated these techniques with HMD occlusion removal. Most existing methods rely on 3D Morphable Model (3DMM) parameters [1, 26], which are difficult to obtain under HMD occlusion. While LatentAvatar [34] avoids the dependency on 3DMM, it falls short in both reconstruction quality and real-time performance. In general, HMD removal faces two key challenges: (1) achieving robust and consistent high-fidelity face restoration from partially visible inputs, and (2) enabling high-resolution real-time online video processing.

In this paper, we propose HFM-GS, a real-time HMD removal method that utilizes latent code and 3D Gaussians as intermediate representations within an end-to-end network framework. First, we establish a mapping field between the upper and lower face Gaussians to enhance the adaptability to deformation. Second, we introduce correlation weight-based sampling to improve time performance and handle variations in the number of Gaussians. Third, we ensure model robustness through Gaussian segregation strategy. Finally, compared to two state-of-the-art (SOTA) methods, DECA + GaussianAvatars [7, 27] and LatentAvatar [34], our method achieves better performance in peak signal-to-noise ratio (PSNR, 30.98), structural similarity index (SSIM, 0.908), learned perceptual image patch similarity (LPIPS, 0.079) and frames per second (FPS, 134). Figure 1 shows the comparison. We also designed a user study to confirm that our method delivers the highest fidelity of HMD removal.

In summary, the contributions of this work can be listed as follows. 1) We propose an HMD removal pipeline based on half-face mapping 3DGS, which enables the reconstruction of facial information occluded by HMDs in real-time video, following a brief user registration process. 2) We propose half-face mapping 3DGS, which captures the relationship between the upper and lower face models. This allows for the estimation of the Gaussian parameters for the upper face under HMD occlusion based on the Gaussian representation of the lower face. 3) We propose an acceleration algorithm, correlation weight based sampling, to reduce the number of Gaussians in half-face mapping. 4) We propose a Gaussian segregation strategy that ensures the proper distribution of Gaussians of the upper and lower face by refining the density control method and loss functions.

2 Related Works

Numerous studies have been conducted in related areas. We review the existing works from three perspectives: occluded facial image inpainting, HMD occlusion removal, and animatable head avatar.

2.1 Occluded Facial Image Inpainting

Early methods to facial image inpainting were originally developed for restoring old photographs, where the missing content was typically minimal and located in non-critical regions. These methods primarily relied on low-level image features, using patch-based techniques to fill occluded areas by matching or copying similar content from surrounding regions [13]. With the advancement of deep learning, GAN significantly advanced facial inpainting by learning complex mappings between incomplete and complete images. Dolhansky et al. [6] utilized GAN to restore occluded eye regions, while Ge et al. [10] and Yang et al. [35] enhanced GAN-based face completion using pre-trained CNNs and facial landmarks, respectively. Nitzan et al. [24] fine-tuned the StyleGAN model to facilitate identity-preserving modifications. More recently, diffusion models have also been employed for facial inpainting, demonstrating promising results [32]. Facial video inpainting extends image inpainting to the temporal domain, requiring constraints for motion consistency. Early methods used recurrent feedback [16] and attention mechanisms [36], while recent work [20] incorporates facial landmarks to enhance temporal coherence. Our method also addresses occluded facial image inpainting. Compared to the previous methods, we ensure real-time online video processing while maintaining inter-frame stability.

2.2 HMD Occlusion Removal

HMD occlusion removal is a specialized form of facial inpainting, addressing structured and large-scale occlusions that cover critical facial regions. Early methods were primarily image-processing-based, e.g. Takemura et al. [29] directly overlaid an unoccluded reference image for restoration.

Similar to face inpainting, neural network-based methods, particularly GANs, have been widely applied to HMD removal. Wang et al. [31] employed a GAN to inpaint occluded regions using a reference image, while Numan et al. [25] incorporated depth information to enhance GAN output quality. Ghorbani et al. [11] focused on ensuring temporal consistency of facial expressions in occluded video sequences. Beyond GAN-based methods, Gupta et al. [12] introduced landmark-based supervision within an encoder-decoder framework to refine eye-region reconstruction without directly generating facial images. Chen et al. [3] utilized a lightweight neural network to restore facial details in stitched facial images. Although these methods have achieved notable progress, they often rely on extensive additional inputs or face challenges in real-time online processing of video streams.

Another category of methods reconstructs and drives 3D head avatars, leveraging offline-generated personalized 3D face models that are dynamically adjusted based on input. Burgos-Artizzu et al. [2] reconstructed a user-specific 3D facial model from multiple images and reprojected it into video frames based on the final HMD position and facial markers. Frueh et al. [8] employed an RGB-D camera to acquire 3D facial data and integrated eye-tracking technology for more precise facial motion driving. Rekimoto et al. [28] utilized a combination of a 3D depth scanner, infrared cameras, and reflectance mirrors to enhance the reconstruction quality. Zhao et al. [37] constructed a user's facial model with large-scale point clouds, and focused on improving eye region reconstruction. More recently, Lu et al. [21] proposed a method that predicts occluded facial landmarks and constructs facial UV maps to drive the 3D face model. Model-based methods provide high-quality and consistent facial reconstruction, with some enabling multi-view rendering essential for VR applications—an advantage that image-based methods often fail to achieve. However, these methods often come with high computational complexity or require specialized hardware, limiting their practicality. In contrast, our method follows a model-based paradigm for HMD occlusion removal while achieving efficient reconstruction with minimal input requirements, offering a more efficient and stable solution.

2.3 Animatable Head Avatar

Face2face [30] significantly contributed to the advancement of digital avatar reconstruction and real-time animation. With the development of novel scene representations, NeRF [23] and Gaussian Splatting [15], head avatars have achieved high reconstruction quality, fast rendering speeds, and strong facial expression reproduction capabilities. Gafni et al. [9] proposed a NeRF based method to learn an expressive head avatar from monocular video input. LatentAvatar [34] directly learned latent code from images, enhancing the ability to capture fine facial details. Instant [38] improved NeRF rendering speed by leveraging the deformation of triangular patches in the FLAME model [18]. The introduction of 3D Gaussian Splatting [15] has demonstrated the potential for constructing high-quality scenes with shorter training time and higher rendering time performance, leading to various Gaussian-based digital avatar methods. Qian et al. [27] bound Gaussians to the FLAME model, enabling deformation driven by FLAME's transformations. Dhamo et al. [4] utilized redundant Gaussians, manipulating their color and transparency to control the virtual avatar's appearance. Ma et al. [22] integrated the BlendShapes method, achieving real-time Gaussian avatar rendering through interpolation. Xu et al. [33] initialized Gaussians using point clouds and directly fitted Gaussian deformations with Multilayer Perceptrons (MLPs).

Despite these advancements, existing head avatar methods face several limitations. Many methods require predefined spatial positions and coordinates of the face, necessitating complex camera calibration,

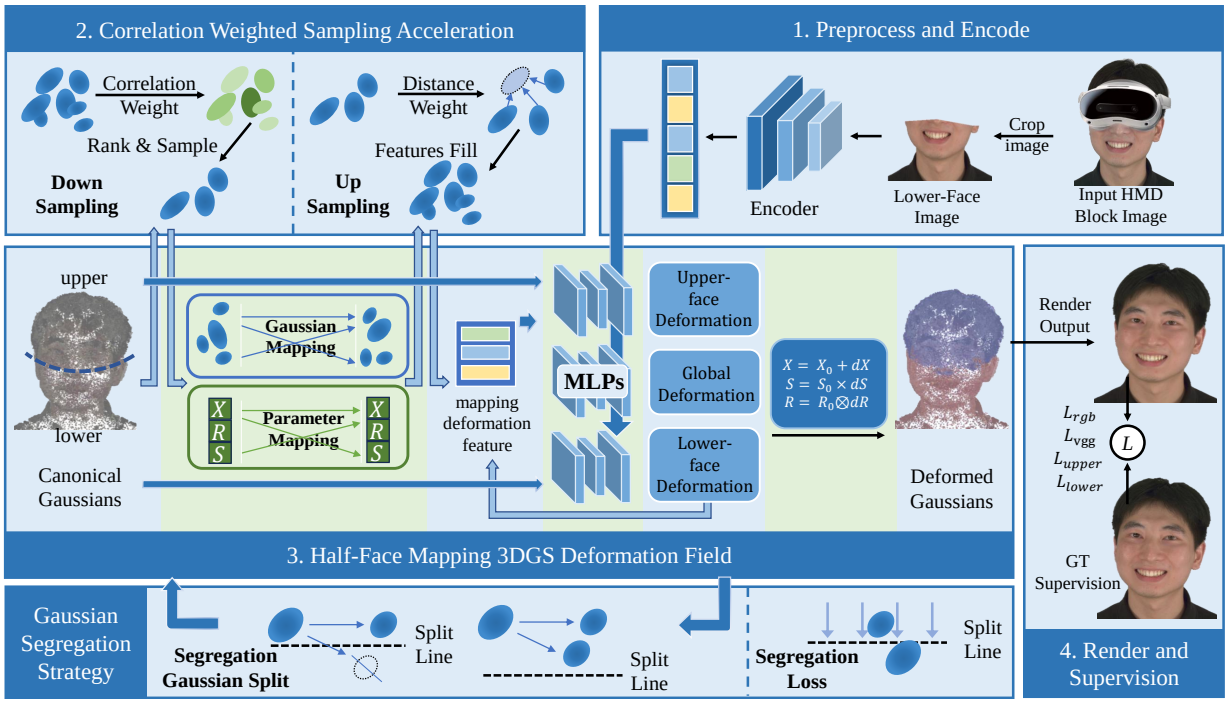

Fig. 2: The pipeline of our HFM-GS HMD removal.

which restricts VR applications. Additionally, most avatar methods heavily rely on facial expression parameters and 3DMM models [1, 26], such as FLAME parameters and mesh, which are challenging and expensive to obtain when obstructed by an HMD. Our method addresses the HMD removal problem based on 3D Gaussian Splatting, ensuring high rendering quality and efficiency while minimizing input requirements. We compare our method with two SOTA methods based on NeRF and 3DGS: LatentAvatar [34] and GaussianAvatars [27] (in conjunction with DECA [7]).

3 Method

3.1 Pipeline

Our method, HFM-GS, is designed to achieve real-time facial de-occlusion in the presence of HMD occlusions. The input consists of two types of facial RGB videos: 1) a pre-recorded, single-view, unoccluded facial video provided by the user during the registration phase, and 2) a real-time, single-view, HMD-occluded facial video captured during the usage phase. The output is an HMD-occlusion-free facial RGB video that restores the missing facial regions.

As shown in Figure 2, our pipeline consists of four steps. The first step includes preprocessing the input user frames and encoding frames to generate a latent code that describes the facial image features. We preprocess the user-provided facial images I_{full} by removing the background [14, 19] and estimating their approximate 2D facial landmarks. Based on these landmarks, we extract the lower half of the face, denoted as I_{lower}, which includes the nasal tip, cheeks, and the regions below—areas that remain visible even under HMD occlusion. The cropped images I_{lower} serve as the network's input. Inspired by LatentAvatar [34], we encode I_{lower} using an encoder E_{img} to obtain a latent code θ. The encoder consists of a sequence of convolutional layers followed by attention modules. Due to the limited accuracy of explicit facial alignment under severe HMD occlusion, which results in reduced reconstruction fidelity as demonstrated by the DECA+GaussianAvatars baseline in Section 4.2, we enable the encoder to simultaneously capture facial expression features and spatial positioning within the frame. In the second step, the upper and lower half-face Gaussian model G_0 in the canonical space are mapped and deformed to the Gaussian model G in the deformed space using a mapping deformation field. Similar to other controllable Gaussian models [4, 27, 33], the Gaussian head model in the canonical space consists of N Gaussians. Each Gaussian is parameterized by its position X, scale S, rotation R, color C, and opacity O (Section 3.2). In the third step, we apply a correlation weight-based sampling algorithm to downsample the Gaussians in the canonical space, reducing the number of Gaussians involved in the mapping computation. After the mapping computation in the second step, we calculate the upsampling weight matrix to restore features for all Gaussians (Section 3.3). In the final step, given the deformed Gaussians G and the camera parameters c, we employ the standard Gaussian rendering pipeline f_{render} to rasterize the head model, generating the final RGB image I_{output}. Supervised learning is performed using full-face reference images I_{full}. We also propose a Gaussian segregation strategy to ensure the correct distribution of Gaussians, involving refinements to both Gaussian density control and loss functions (Section 3.4). During the training phase, we initialize the canonical-space Gaussians G_0 using the FLAME mesh and use I_{lower} and I_{full} of the reference video as input and supervision separately. Through the four-step pipeline, we optimize G_0 along with other network parameters. During inference, we take the canonical Gaussians G_0 and I_{lower} of occluded frames as inputs, and produce full-face outputs via the trained network.

3.2 Half-face Mapping 3DGS Deformation Field

We propose the half-face mapping 3DGS deformation field, which deforms the head Gaussians from the canonical space to the deformed space based on the input latent code θ. The core idea of the half-face mapping 3DGS deformation field is to divide the face into upper and lower regions, establish a mapping relationship between their Gaussian representations, and compute the Gaussian deformations for each region separately. Specifically, the lower face, which has complete reference input, can accurately describe Gaussian deformations and thus requires only a simple deformation field. In contrast, the upper face lacks direct reference information and relies on neural network predictions. Given the holistic nature of facial expressions (e.g., when the mouth opens in surprise, the eyes tend to widen), we establish the mapping relationship between the Gaussians of the upper and lower face and incorporate the mapping features as input to the deformation prediction of the upper face.

The complete Gaussian model G is divided into the upper-face Gaussians G_{upper} and lower-face Gaussians G_{lower}. The deformation field

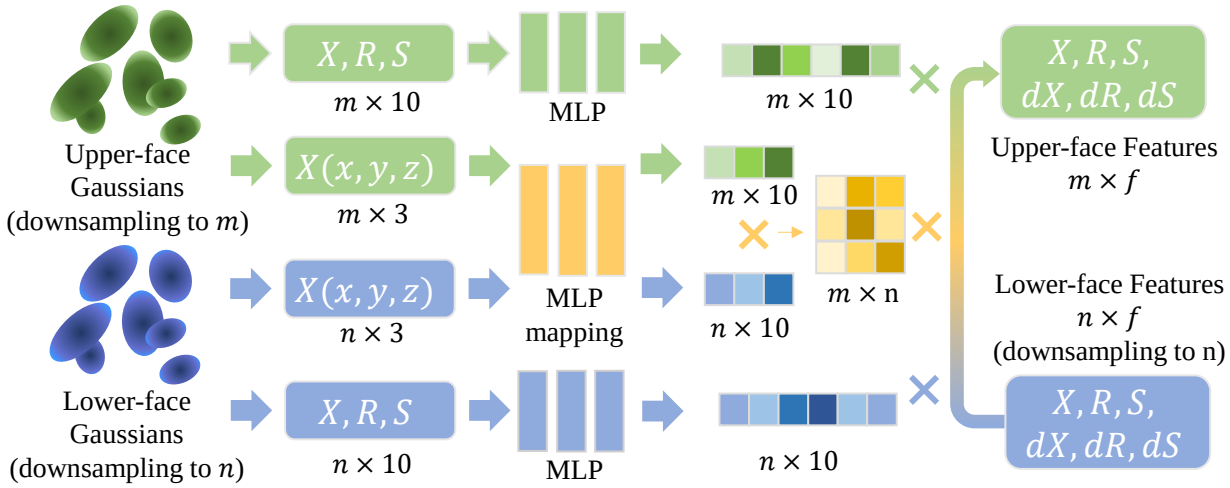

Fig. 3: Gaussian mapping and parameter mapping.

computes the Gaussian deformation dG for the upper and lower-face Gaussians, including variations in position (dX), scale (dS), and rotation (dR) to model the deformation process. As adopted in prior Gaussian avatar methods [22, 27, 33], we fix the color and opacity to prevent the model from adapting to facial motions by simply changing the two attributes, thereby avoiding the generation of redundant Gaussians and maintaining structural stability. This is formulated as:

$$X = X_0 + dX, S = S_0 \cdot dS, R = R_0 \otimes dR \tag{1}$$

Here, the position variation dX is applied through direct addition. The scale variation dS is a multiplicative factor that adjusts the original scale S_0. The rotation variation dR is composed with the initial rotation R_0 using quaternion multiplication $\otimes$. We also employ an embedding method that projects the original 3D coordinates X into a higher-dimensional feature space.

For the lower-face Gaussians, we employ simple MLPs (MLP_X, MLP_S, MLP_R) to model the Gaussian variations:

$$dG_{lower}(dX_{lower}, dS_{lower}, dR_{lower}) = MLPs(X_{0,lower}, \theta) \tag{2}$$

where $X_{0,lower}$ represents the lower-face Gaussians in canonical space, and dX_{lower}, dS_{lower} and dR_{lower} denotes the variation in the position, scale and rotation of the lower-face Gaussians separately.

For the upper-face Gaussians, we first establish a structured mapping relationship between the upper-face and lower-face Gaussians, which includes two parts: Gaussian mapping and parameter mapping. The mapping process is shown in Figure 3. We then compute the deformation of the upper-face Gaussians based on the upper-face Gaussian parameters, the parameters and their deformation from the lower-face Gaussians in the mapping relationship, and the latent code.

For the Gaussian mapping, we establish mapping between Gaussians by computing the dependency weights of each upper-face Gaussian on each lower-face Gaussian. To compute the dependency weights, we first define an MLP-based function MLP_{gm} that encodes the positional, scaling, and rotational attributes of upper and lower-face Gaussians into 10-dimensional Gaussian mapping feature embeddings ϕ:

$$\phi = MLP_{gm}(X, S, R) \tag{3}$$

where X, S, and R denote the position, scale, and rotation of the upper- and lower-face Gaussians, respectively. Given the upper face feature embeddings ϕ_{upper} and lower face feature embeddings ϕ_{lower}, we compute the Gaussian mapping dependency weight matrix W_{gm} through the inner product operation:

$$W_{gm} = \text{Softmax}_{\text{row}}\left(\phi_{\text{upper}} \cdot \phi_{\text{lower}}^T\right) \tag{4}$$

Each entry $W_{gm,ij}$ represents the influence of the j-th lower-face Gaussian on the i-th upper-face Gaussian. To ensure that these weights form a valid probability distribution, we also apply a softmax normalization across each row of W_{gm}.

The computed Gaussian mapping weight matrix W_{gm} is then used to propagate the deformation information from the lower face to the upper face. Specifically, we use it to compute the variation of Gaussian parameters for the upper face:

$$dG_{upper} = MLP_{mapping}(G_{0,upper}, \theta, W_{gm} \times (dG_{lower}, G_{lower})) \tag{5}$$

Here, dG_{upper} represents the estimated deformation of the upper-face Gaussians, computed by an MLP-based function $MLP_{mapping}$ that takes the upper-face Gaussian parameters $G_{0,\text{ upper}}$, weighted sum of the lower-face Gaussian parameters G_{lower} and corresponding deformation dG_{lower}, along with the latent code θ, as input. Among them, $G_{0,\text{ upper}}$ is optimized during the training phase and used during inference, while all other parameters serve as intermediate variables in both phases.

For the parameter mapping, we quantify the influence of each component of parameters (X, R, S, dX, dR, dS) on the mapping process for every Gaussian involved in the computation. Specifically, we define an MLP-based function MLP_{pm} that takes the upper- and lower-face Gaussian parameters as input and outputs the corresponding weights for each component of parameters in the mapping process.

Formally, given a Gaussian set G_0 in canonical space, we compute a parameter mapping weight vector W_{pm} as:

$$W_{pm} = MLP_{pm}(G_0). \tag{6}$$

Incorporating these lower- and upper-face parameter mapping weight $W_{pm,lower}$ and $W_{pm,upper}$, we refine the deformation propagation Formula (5) as follows:

$$\begin{aligned} dG_{upper} = & MLP_{mapping}\Big(G_{0,upper}, \theta, \\ & W_{pm,upper} \times W_{gm} \times W_{pm,lower} \times (dG_{lower}, G_{lower})\Big) \end{aligned} \tag{7}$$

The introduction of the parameter mapping allows the network to adaptively adjust the influence of different parameters and parameter components during deformation.

To account for global facial transformations, we introduce a global Gaussian deformation dG_{global} modeled by an MLP. This MLP predicts the global translation dX_{global} and rotation dR_{global}, which are applied to all Gaussians:

$$dG_{global}(dX_{global}, dR_{global}) = MLP_{global}(\theta). \tag{8}$$

3.3 Correlation Weight based Sampling Acceleration

A head avatar reconstructed using Gaussian Splatting typically consists of 10^4 to 10^5 Gaussians. If all Gaussians were involved in computing the half-face mapping 3DGS deformation field, the resulting mapping weight matrix would have a size of $10^5 \times 10^5$, leading to substantial computational and memory overhead. Moreover, certain regions, such as the cheeks and shoulders, do not require highly detailed Gaussian deformations, making a full-scale computation inefficient and unnecessarily expensive.

Therefore, we propose a correlation weight based sampling algorithm, which performs downsampling on Gaussians based on their contribution to the deformation mapping. These sampled Gaussians G_{key} participate in the mapping computation. After mapping computation, we apply upsampling to propagate the features of the key Gaussians to all Gaussians.

Specifically, we utilize the MLP_{gm} introduced in Section 3.2, which generates a weight vector for each Gaussian parameter component. We extract the x, y, and z components of position X and compute their square root to serve as the correlation weight for downsampling. We then rank all Gaussians based on the weights and select the top-ranked Gaussians from the upper and lower face regions separately. Even as Gaussians are dynamically added or removed during training, using the x,y, and z components as indicators of Gaussian mapping weights ensures the stability of the model. After the training is completed, the selection of the key Gaussians remains unchanged during inference, as it depends on Gaussian parameters in the canonical space, which remain fixed.

We compute the dependency weight W_{dist} of upper-face Gaussians on key Gaussians based on their pairwise spatial distances:

$$W_{dist,ij} = \frac{1/(Distance(G_{key,j}, G_{upper,i}) + eps)}{\sum_{G_{key}} 1/(Distance(G_{key}, G_{upper,i}) + eps)} \tag{9}$$

where i and j denote the i-th upper-face Gaussian and the j-th sampled Gaussian, respectively. *Distance* is the spatial distance and *eps* represents a small positive value. After completing the Gaussian deformation mapping, we use W_{dist} to restore missing feature for all upper-face Gaussians. Formula (7) is further refined as:

$$\begin{aligned} dG_{upper} = & MLP_{mapping}\Big(G_{0,upper}, \theta, \\ & W_{dist} \times W_{pm,upper} \times W_{gm} \times W_{pm,lower} \times (dG_{key}, G_{key})\Big) \end{aligned} \tag{10}$$

3.4 Gaussian Segregation Strategy

Since we divide the Gaussians into upper-face and lower-face regions, some Gaussians may shift into the opposite region during training, potentially affecting the model's stability. To address this issue, we introduce the Gaussian Segregation Strategy, which adjusts Gaussian density control and modifies the loss function.

For Gaussian density control, as shown in Algorithm 1, if a newly generated Gaussian from Gaussian split crosses the half-face threshold y_t and exceeds a predefined margin m, the Gaussian is discarded (lines 6-10). If at least one newly generated Gaussian from the split does not fall into the opposite region, the new Gaussians are retained, and the original Gaussian is removed; otherwise, the split is discarded (lines 11-13).

Algorithm 1: Gaussian Segregation Density Control

```
1  Input: Gaussian sets G_upper, G_lower, threshold y_t, margin m
2  Output: Processed Gaussian sets G_upper, G_lower
3  for G_seg ∈ {G_upper, G_lower} do
4      for g ∈ select_for_split(G_seg) do
5          g_1, g_2 ← gaussian_split(g)
6          for g_i ∈ {g_1, g_2} do
7              if G_seg = G_upper and g_i.y < y_t − m then
8                  g_i ← None
9              else if G_seg = G_lower and g_i.y > y_t + m then
10                 g_i ← None
11         if g_1 ≠ None or g_2 ≠ None then
12             G ← G ∪ {g_1, g_2}
13             delete(g)
```

To prevent Gaussians from crossing the boundary during parameter adjustment, we introduce novel loss functions to enforce this constraint:

$$L_{\text{upper}} = \text{ReLU}(-(G_{upper}.y - (y_t - d))), \tag{11}$$

$$L_{\text{lower}} = \text{ReLU}(G_{lower}.y - (y_t + d)). \tag{12}$$

where the threshold y_t and margin m are the same as those used in Gaussian segregation density control. Here, $G_{upper}.y$ and $G_{lower}.y$ represent the y-coordinates of the position parameters for the upper-and lower-face Gaussians, respectively. Then, the final loss function is formulated as a weighted sum:

$$L = \lambda_{\text{rgb}} L_{\text{rgb}} + \lambda_{\text{vgg}} L_{\text{vgg}} + \lambda_{\text{split}} (L_{\text{upper}} + L_{\text{lower}}), \tag{13}$$

where L_{rgb} represents the original 3DGS loss, which includes L_1 loss and L_{d-ssim} loss. L_{vgg} computes a VGG-based perceptual loss between the rendered image and the ground truth. λ_{rgb}, λ_{vgg} and λ_{split} are hyperparameters controlling the relative contributions of reconstruction accuracy, perceptual quality and segregation consistency, respectively.

4 Evaluation

4.1 Implementation

Our method is trained on a per-subject basis using monocular facial video sequences. During the inference stage, the viewpoint remains fixed. To assess the performance of our half-face mapping 3DGS method, we conducted a series of experiments on the *Nersemble* dataset [17], selecting 9 representative subjects. For each subject, a consistent viewpoint is used for both training and inference. For each subject, we utilize 8 video sequences, and the image resolution is downsampled to 512×512. Across all sequences, each subject contains approximately 1,500 frames in total. When splitting the data into training and testing sets, we reserve either the first or last 20% of the frames from each video sequence for testing, while the remaining 80% is allocated for training. Given that background removal and re-insertion are relatively straightforward for applications, we remove the background for the video sequences, similar to prior works [34], and focus on reconstructing the face.

Our encoder consists of a five-layer convolutional network. Prior to encoding, the input image I_{full} is cropped to extract the lower-face region I_{lower} and subsequently downsampled to 128×128 resolution. The encoder then generates a 128-dimensional latent code, which serves as a compact and efficient representation of the lower face. For Gaussian mapping of half-face mapping 3DGS deformation field, we use MLP_{gm} to generate a ten-dimensional feature vector ϕ, with the element-wise product of two vectors determining the corresponding mapping weight matrix W_{gm}. For 3D Gaussian sampling based on correlation weights, we select the 512 highest-weighted Gaussian components from both the upper and lower facial regions, which are then used in the Gaussian mapping calculation to produce a 512×512 Gaussian mapping weight matrix.

We employ the Adam optimizer [5] with a fixed learning rate of 1×10^{-4}. In our model, the trainable parameters include the three-dimensional Gaussian parameters $\{X_0, S_0, R_0, C_0, O_0\}$, the encoder parameters$\{E_{img}\}$, the parameters of half-face mapping 3D Gaussian deformation field $\{MLP_X, MLP_S, MLP_R, MLP_{gm}, MLP_{mapping}, MLP_{pm}, MLP_{global}\}$. The model is trained for a total of 10,000 iterations. The final model consists of approximately 100,000 3D Gaussians after training is completed. The loss function incorporates weighted components to strike a balance between reconstruction accuracy and stability, with the following weights applied: $\lambda_{\text{rgb}} = 0.4, \lambda_{\text{vgg}} = 0.1, \lambda_{\text{split}} = 0.01$ All the following experiments were conducted on a workstation equipped with a 13th Gen Intel® Core™ i9-13900F processor, 64GB of RAM, and an NVIDIA GeForce RTX 4080 GPU.

4.2 Comparison

We compare our proposed HFM-GS method with two methods for head reconstruction: LatentAvatar [34] and GaussianAvatars [27]. To adapt these methods to our task, we introduce necessary modifications. Similar to our method, all methods are trained and tested using monocular images without relying on the original input camera parameters or FLAME parameters [18]. During inference, only facial image sequences occluded by the HMD are provided. Specifically, for the LatentAvatar, we enhance the encoder by incorporating additional learning of facial position information. For the GaussianAvatars, considering the real-time requirements of VR environments, we employ the online FLAME parameter estimation method, DECA [7], to estimate FLAME parameters dynamically.

4.2.1 Quality

Figure 4 presents a comparison between our method (4th column), the other two methods (2nd and 3rd columns), and the ground truth (5th column) across four different identities, given occluded facial images as input (1st column). The red and green squares highlight specific detail regions, which are magnified below each image. Specifically, due to the potentially large errors in facial pose estimation by DECA + GaussianAvatars, the highlighted regions in the second column differ from those in the other columns, allowing for a more intuitive comparison of the corresponding facial details.

For the results generated by DECA + GaussianAvatars (the second column in Figure 4), the predicted facial pose may exhibit significant errors. For example, in the second row, the face appears to lean forward, while in the fourth row, the head tilts noticeably to the right, deviating from the ground truth. Additionally, the method faces challenges in reconstructing fine details, as seen in the blurriness of the teeth in the third-row image and the eyes in the fourth-row image. Furthermore, it fails to accurately restore fine facial wrinkles, such as the mouth corner creases in the fourth-row image. These issues can be attributed to two main factors. First, although DECA provides fast processing, it estimates FLAME parameters [18] and poses independently for each frame, leading to large estimation errors and inconsistencies across frames. Although GaussianAvatars can fine-tune input parameters during training, it faces difficulties in compensating for excessive estimation errors from DECA. Second, when the upper face is heavily occluded during the usage phase, DECA's facial pose estimation becomes unreliable, and it fails to estimate FLAME parameters for the upper face, preventing GaussianAvatars from generating a plausible reconstruction.

For the results generated by LatentAvatar (the third column in Figure 4), the overall detail reconstruction quality is inferior to our method. For example, in the second row, the green-boxed region of the glasses exhibits noticeable aliasing and unnatural shading. In the third row, while the reconstructed teeth contours are distinguishable, the number of teeth is incorrect, and the realism is suboptimal. Additionally, the reconstruction of details related to facial expression and pose variation is weaker than in our method. For instance, wrinkles near the mouth corners in the first and fourth rows are less accurately restored, the forehead creases in the first row are not well represented, and in the second row, the shading on the glasses is incorrect, with erroneous non-transparent blurring at the edges. LatentAvatar employs NeRF as the decoder, which has inherent limitations in image reconstruction quality.

Our method achieves the closest visual resemblance to the ground truth, compared to DECA + GaussianAvatars and LatentAvatar. Our method excels in facial expression restoration, accurately reconstructing the pose, expression, teeth, and wrinkles for both the upper and lower parts of the face. First, our method directly generates the latent code from the occluded image, making it independent of FLAME models and the inaccuracies in FLAME and pose parameter estimation. This allows for better preservation of fine details, such as wrinkles, ensuring more reliable reconstruction. Additionally, our approach is based on 3D Gaussian Splatting, which provides a powerful reconstruction capability, enabling high-quality Gaussian representation for the face. Finally, our method employs half-face mapping to establish a deformation relationship between the upper and lower halves of the face, enhancing the utilization of full-face deformation information compared to methods where each facial region independently relies on encoder-provided features. Our method reconstructs a personalized facial model for each individual. For the Nersemble dataset, our approach is able to faithfully reconstruct a wide range of expressions present in the training data, including smile, laughter, surprise, anger, and confusion, along with high-quality upper-face details such as forehead wrinkles, frowning, eyebrow raising, eye gaze changes, and wide-open eyes. For user-captured data, the diversity of real-time de-occluded facial expressions depends on the diversity of expressions in the pre-captured data.

For quantitative comparison, all methods were evaluated by computing mean squared error (MSE), L1 loss, peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), and learned perceptual image patch similarity (LPIPS) to quantify the difference between the generated images and the ground truth. As shown in Table 1, our method significantly outperforms the other two methods across all metrics.

Specifically, compared to DECA+GaussianAvatars, our method achieves a 96% reduction in MSE, an 81% reduction in L1 loss, a 13.98 dB increase in PSNR, a 0.163 improvement in SSIM, and a 0.208 decrease in LPIPS. Compared to LatentAvatar, our method reduces MSE by 50%, L1 loss by 32%, increases PSNR by 3.04 dB, improves SSIM by 0.037, and decreases LPIPS by 0.042.

4.2.2 Time performance

The last column of Table 1 compares the inference speed of our method against DECA + GaussianAvatars and LatentAvatar. The results demonstrate that our method significantly outperforms the two comparison methods in computational efficiency, achieving a 7.44× speedup over DECA + GaussianAvatars and a 6.09× speedup over LatentAvatar. Although GaussianAvatars has a relatively high rendering rate, its animation heavily depends on the estimation of FLAME shape and pose parameters [18], which incurs a high computational cost, limiting overall inference speed. DECA processes images at only 22 FPS and while multi-threading can improve throughput, it significantly increases processing latency. LatentAvatar, constrained by the inherent computational cost of NeRF and additional super-resolution modules, has a low processing speed. In contrast, our method achieves 134 FPS at a 512×512 resolution and maintains over 60 FPS even at a 1024×1024 resolution, fully meeting the real-time requirements of XR applications while ensuring a seamless user experience.

4.3 Ablation Studies

In this section, we conduct ablation studies to evaluate the impact of each key part in our proposed half-face Mapping 3DGS on image quality and rendering time performance.

4.3.1 Ablation on half-face mapping 3DGS deformation field

To evaluate the impact of each module on reconstruction quality, we conduct an ablation study by systematically removing different parts from our full method. The complete model is denoted as Full. First, we remove the global pose prediction, instead optimizing a fixed global pose that does not vary with the input image. This variant is denoted

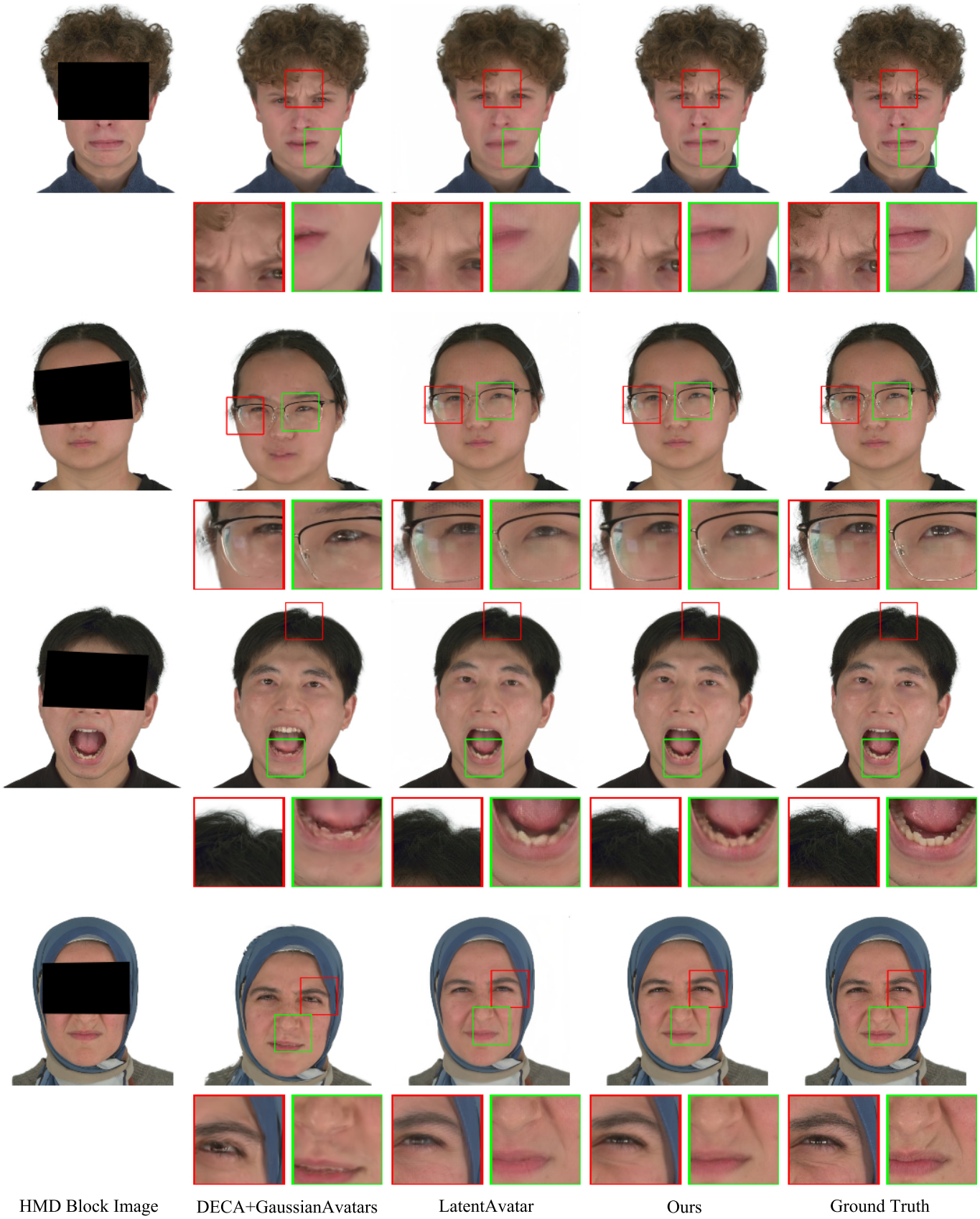

Fig. 4: Comparison of HMD removal images with our HFM-GS method and the previous methods on Nersemble dataset.

Table 1: Comparison of methods in terms of MSE, L1, PSNR, SSIM, LPIPS, and FPS

Method	MSE↓	L1↓	PSNR↑	SSIM↑	LPIPS↓	FPS↑
DECA [7] + GaussianAvatars [27]	0.0214	0.0616	17.00	0.745	0.287	18
LatentAvatar [34]	0.0016	0.0169	27.94	0.871	0.121	22
Ours	**0.0008**	**0.0115**	**30.98**	**0.908**	**0.079**	**134**

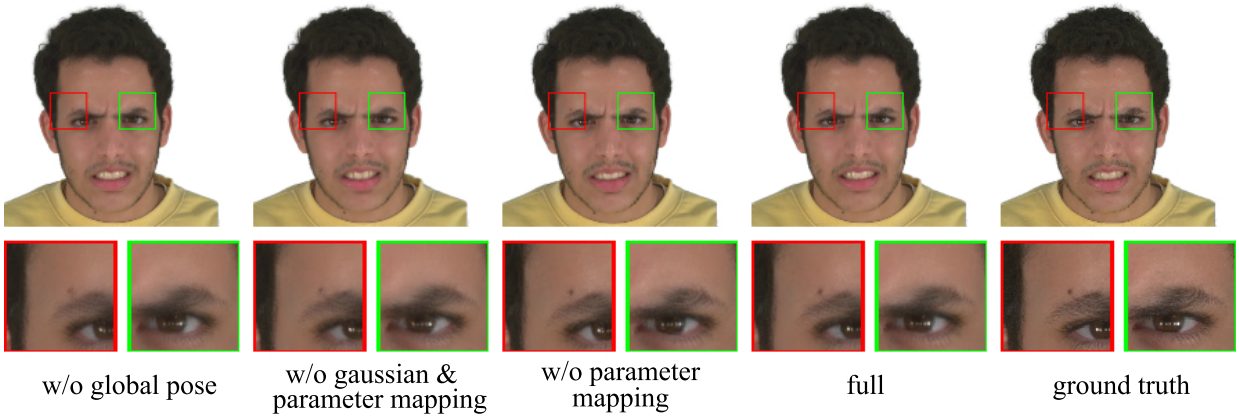

Fig. 5: Qualitative comparisons of ablation methods.

as w/o global pose. Next, we remove the Gaussian mapping module. Since the feature mapping module relies on Gaussian mapping, both modules are removed together. We use simple MLP_X, MLP_S, and MLP_R to predict deformation parameters for the entire face Gaussians together with global pose prediction. This variant is denoted as w/o Gaussian & feature mapping. Finally, we remove only the feature mapping module, ignoring the feature mapping weights during computation. This variant is denoted as w/o feature mapping.

Table 2 presents the experimental results for different module combinations. Specifically, compared to w/o global pose, our Full method improves PSNR by 0.47 dB, SSIM by 0.003, and reduces LPIPS by 0.003. Compared to w/o feature mapping, it improves PSNR by 0.19 dB, reduces LPIPS by 0.002, while SSIM remains unchanged. Compared to w/o Gaussian & feature mapping, it further improves PSNR by 1.00 dB, SSIM by 0.008, and reduces LPIPS by 0.009. Overall, our Full method achieves the highest quality across all evaluation metrics (PSNR, SSIM, and LPIPS).

Table 2: Comparison of ablation methods in terms of PSNR, SSIM, LPIPS, and FPS

Method	PSNR↑	SSIM↑	LPIPS↓
w/o global pose	30.51	0.905	0.082
w/o Gaussian & feature mapping	29.98	0.900	0.088
w/o feature mapping	30.79	0.908	0.081
Full	**30.98**	**0.908**	**0.079**

Figure 5 further demonstrates the effectiveness of each module in our method. Compared to w/o global pose (first column), w/o Gaussian & feature mapping (second column), and w/o feature mapping (third column), our method (fourth column) achieves superior performance in preserving the details of dynamic facial expressions, closely approximating the ground truth (fifth column). In the red-boxed region, the black spot above the eyebrow in the Full method exhibits the most similar shape and color to the ground truth while avoiding the noticeable blurriness seen in the first column. Similarly, in the green-boxed region, the eye highlights and the overall eyebrow shape generated by our method are the most natural and closest to the ground truth.

4.3.2 Ablation on correlation weight based sampling

We evaluate the effectiveness of correlation weight-based sampling in reducing computational overhead by measuring frame rates with and without sampling under different numbers of Gaussians, denoted as Ours and w/o Gaussian sampling, respectively. As shown in Table 3, for a smaller number of Gaussians (e.g., 40K), our full method achieves approximately 1.53× the frame rate of w/o Gaussian sampling. For Gaussian rendering, 100K Gaussians provide optimal visual quality, where Ours achieves 1.16× the rendering speed of 40K Gaussians (w/o Gaussian Sampling). Beyond 40K Gaussians, w/o Gaussian sampling exceeds the device's memory limit, constraining facial rendering quality and achieving only $\sim 28dB$ PSNR on the test identity.

Table 3: Ablation on correlation weight based sampling

Method	Gaussian numbers	fps
w/o Gaussian sampling	$\sim 40K$	112
w/o Gaussian sampling	$\sim 70K$	memory out
w/o Gaussian sampling	$\sim 100K$	memory out
Ours	$\sim 40K$	176
Ours	$\sim 70K$	149
Ours	$\sim 100K$	130

4.3.3 Ablation on Gaussian segregation strategy

Our half-face mapping 3DGS deformation field requires establishing a mapping between the upper and lower face. Properly splitting and maintaining this segregation during training is crucial for model stability. Without Gaussian segregation strategy, Gaussians may diffuse beyond the boundary, leading to incorrect training results. Figure 6 illustrates the Gaussian distributions of the upper and lower face in a single rendering with (left) and without (right) Gaussian segregation strategy. The upper-face Gaussians are represented in blue, while the lower-face Gaussians are shown in orange. It can be observed that with Gaussian segregation strategy achieves a well-separated distribution between the upper and lower Gaussians, whereas w/o Gaussian segregation strategy exhibits mixed Gaussians across the two regions, which negatively impacts Gaussian deformation computation.

5 USER STUDY

The experimental protocol was approved by the Biology and Medical Ethics Committee of Beihang University and conducted in accordance with ethical standards. We designed a within-subject study to evaluate the perceptual synthesis quality on the Nersemble dataset [17] of our

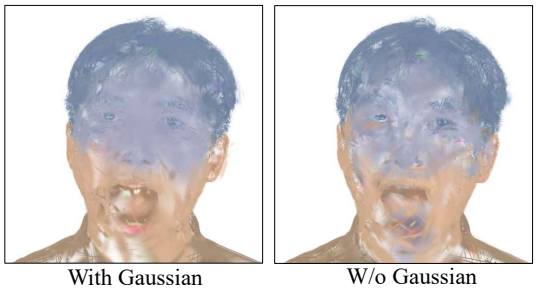

Fig. 6: Gaussian distribution of with or w/o Gaussian segregation strategy

HFM-GS method compared with the DECA+GaussianAvatars and LatentAvatar.

Participants and Setup We recruited 14 participants (7 males and 7 females, aged between 21-35 years), all of whom had experience using XR HMDs and had normal or corrected-to-normal vision. Each participant wore a Pico 4 headset for the experiment.

Conditions The conditions included: the ground truth results in datasets (GT), our method (EC), DECA + Gaussian (CC1), and LatentAvatar (CC2). Both methods are trained on the same sequences of the Nersemble dataset. We set the optimal parameters reported in their paper.

Task We randomly selected consecutive video frames from the test set and generated HMD removal videos based on these frame sequences in XR and normal views, as shown in Figure 7. The input frame rate for the videos was capped at 30 frames per second (fps). If a method's processing speed was insufficient to maintain 30 fps, unprocessed frames were skipped, with the preceding frame used as a substitute. We utilized 4 identities from the Nersemble dataset, generating 8-second-long videos at 30 fps for each identity. To ensure accuracy and fairness in comparison, the user's viewing perspective was fixed, and the visual content of the video remained consistent across all methods.

Participants were presented with the videos in VR and asked for realistic ratings. For each identity, the GT video was first shown to the participants, and they were informed that it was the ground truth. Subsequently, videos generated by EC, CC1 and CC2 were displayed in a randomized order. The number of times each method was viewed was balanced across the experiment. After watching the videos, participants were presented a questionnaire to give the fidelity score of the HMD removal result on a 5-point Likert scale, where 1 indicated "very unrealistic compared to the ground truth", and 5 indicated "very realistic compared to the ground truth". To mitigate visual fatigue, participants were given a 10-second break before proceeding to the next video.

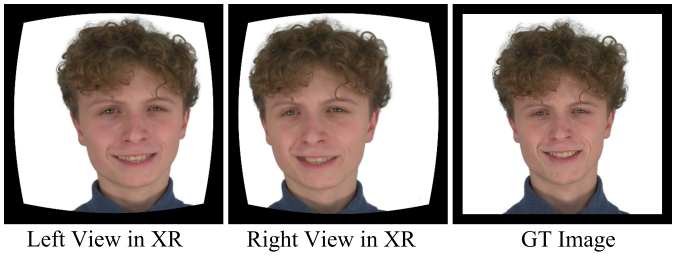

Fig. 7: Visualizaition of XR views.

Results Figure 8 presents the statistical results of the average scores across all identities for the three evaluated methods. Our proposed method, EC, achieves an average fidelity score of 4.57 with a standard deviation of 0.27, while the two comparison methods, CC1 and CC2, obtain average fidelity scores of 3.29 and 1.61 with standard deviations of 0.26 and 0.23, respectively. To assess the significance of the fidelity score differences, we computed the p-values and Cohen's d effect sizes. The score difference between EC and CC1 is statistically significant ($p-value < 0.001$) with a huge effect size ($Cohen'sd = 11.7925$). Similarly, the score difference between EC and CC2 is also statistically significant ($p-value < 0.001$) with a huge effect size ($Cohen'sd = 4.906$). These results demonstrate that our method significantly enhances the realism of HMD occlusion removal in videos compared to other methods. This improvement can be attributed to the use of 3D Gaussian splatting pipeline, combined with a half-face Gaussian mapping strategy that effectively models facial deformations, achieving high-fidelity and high-frame-rate occlusion removal.

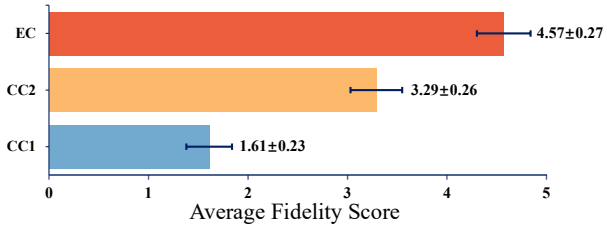

Fig. 8: The user's average fidelity scores and standard deviations for all conditions in our user study.

6 Conclusion, Limitations and Future Work

We proposed HFM-GS, the first Gaussian-based HMD removal pipeline, which achieves high-efficiency and high-fidelity HMD removal based on user-registered images and real-time input of HMD-occluded videos, while also providing VR binocular rendering. Our method introduces three novel components. The half-face mapping 3DGS deformation field establishes a deformation mapping between the upper and lower face Gaussians, improving the ability to fit and predict deformations. The correlation weight-based sampling acceleration is designed to enhance efficiency and address the issue of excessive Gaussians. The Gaussian segregation strategy ensures the correct segmentation of upper and lower face Gaussians, maintaining model accuracy. We conducted objective and subjective evaluations on the Nersemble dataset [17], demonstrating that HFM-GS significantly surpasses existing NeRF- and 3DGS-based SOTA methods in both rendering quality and time performance. Ablation studies further validate the effectiveness of parts of our proposed method. Finally, the user study confirms that our method provides the highest-fidelity result compared to other methods.

Our method has several limitations. First, as it focuses on achieving HMD removal with minimal input information, it does not currently incorporate additional eye-related data, which may result in limited blinking and eye movement. In future work, we plan to integrate eye information as additional input during training to better control blinking and eye movement. During inference, we aim to either introduce periodic custom signals for control or develop efficient, low-latency real-time signal processing as HMD technology advances. Second, our method relies on the lighting conditions present in the training images. If the lighting environment in real-world usage differs significantly from that of the training data, the rendering results may exhibit noticeable discrepancies in illumination. To address this, we plan to explore the relationship between illumination and Gaussian color and transparency deformation fields so as to introduce Gaussian-based illumination processing techniques and leverage advancements in Gaussian-based scene relighting techniques in future research. Lastly, our current method constructs and drives an individual avatar for each user, without considering shared facial characteristics across different individuals. This approach still requires a diverse set of registration frames with varied expressions and poses and limits the expression diversity when the pre-collected videos used for training are insufficient or lack variation. In the future, we aim to develop a pre-trained model generalized across different faces, which can be fine-tuned for individual users, thereby reducing the dependence on extensive registration sequences.

Acknowledgments

This work is supported by the National Natural Science Foundation of China through Project 61932003, 62372026, 62377004, by Beijing Science and Technology Plan Project Z221100007722004, by National Key R&D plan 2019YFC1521102, and by the Fundamental Research Funds for the Central Universities 2233100028.

REFERENCES

[1] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Transactions on pattern analysis and machine intelligence*, 25(9):1063–1074, 2003. 2, 3

[2] X. P. Burgos-Artizzu, J. Fleureau, O. Dumas, T. Tapie, F. LeClerc, and N. Mollet. Real-time expression-sensitive hmd face reconstruction. In *SIGGRAPH Asia 2015 Technical Briefs*, pp. 1–4. 2015. 2

[3] Z. Chen, Z. Zhang, J. Yuan, Y. Xu, and L. Liu. Show your face: Restoring complete facial images from partial observations for vr meeting. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 8688–8697, 2024. 1, 2

[4] H. Dhamo, Y. Nie, A. Moreau, J. Song, R. Shaw, Y. Zhou, and E. Pérez-Pellitero. Headgas: Real-time animatable head avatars via 3d gaussian splatting. In *European Conference on Computer Vision*, pp. 459–476. Springer, 2024. 2, 3

[5] K. Diederik. Adam: A method for stochastic optimization. *(No Title)*, 2014. 6

[6] B. Dolhansky and C. C. Ferrer. Eye in-painting with exemplar generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7902–7911, 2018. 2

[7] Y. Feng, H. Feng, M. J. Black, and T. Bolkart. Learning an animatable detailed 3d face model from in-the-wild images. *ACM Transactions on Graphics (ToG)*, 40(4):1–13, 2021. 1, 2, 3, 6, 8

[8] C. Frueh, A. Sud, and V. Kwatra. Headset removal for virtual and mixed reality. In *ACM SIGGRAPH 2017 Talks*, pp. 1–2. 2017. 1, 2

[9] G. Gafni, J. Thies, M. Zollhofer, and M. 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. 2

[10] S. Ge, C. Li, S. Zhao, and D. Zeng. Occluded face recognition in the wild by identity-diversity inpainting. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(10):3387–3397, 2020. 2

[11] F. Ghorbani Lohesara, K. Egiazarian, and S. Knorr. Expression-aware video inpainting for hmd removal in xr applications. In *Proceedings of the 20th ACM SIGGRAPH European Conference on Visual Media Production*, pp. 1–9, 2023. 1, 2

[12] S. Gupta, S. S. Jinka, A. Sharma, and A. Namboodiri. Supervision by landmarks: An enhanced facial de-occlusion network for vr-based applications. In *European Conference on Computer Vision*, pp. 323–337. Springer, 2022. 1, 2

[13] B.-W. Hwang and S.-W. Lee. Reconstruction of partially damaged face images based on a morphable face model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(3):365–372, 2003. 2

[14] Z. Ke, J. Sun, K. Li, Q. Yan, and R. W. Lau. Modnet: Real-time trimap-free portrait matting via objective decomposition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 1140–1147, 2022. 3

[15] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 2

[16] D. Kim, S. Woo, J.-Y. Lee, and I. S. Kweon. Deep video inpainting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5792–5801, 2019. 2

[17] T. Kirschstein, S. Qian, S. Giebenhain, T. Walter, and M. Nießner. Nersemble: Multi-view radiance field reconstruction of human heads. *ACM Transactions on Graphics (TOG)*, 42(4):1–14, 2023. 1, 5, 8, 9

[18] T. Li, T. Bolkart, M. J. Black, H. Li, and J. Romero. Learning a model of facial shape and expression from 4d scans. *ACM Trans. Graph.*, 36(6):194–1, 2017. 2, 6

[19] S. Lin, L. Yang, I. Saleemi, and S. Sengupta. Robust high-resolution video matting with temporal guidance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 238–247, 2022. 3

[20] F. G. Lohesara, K. Eguiazarian, and S. Knorr. Towards realistic landmark-guided facial video inpainting based on gans. *Electronic Imaging*, 36:1–6, 2024. 2

[21] T. Lu, Z. Peng, X. Xing, X. Xu, and J. Pang. A general method of realistic avatar modeling and driving for head-mounted display users. *IEEE Transactions on Cognitive and Developmental Systems*, 14(3):916–925, 2021. 2

[22] S. Ma, Y. Weng, T. Shao, and K. Zhou. 3d gaussian blendshapes for head avatar animation. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–10, 2024. 2, 4

[23] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2

[24] Y. Nitzan, K. Aberman, Q. He, O. Liba, M. Yarom, Y. Gandelsman, I. Mosseri, Y. Pritch, and D. Cohen-Or. Mystyle: A personalized generative prior. *ACM Transactions on Graphics (TOG)*, 41(6):1–10, 2022. 2

[25] N. Numan, F. Ter Haar, and P. Cesar. Generative rgb-d face completion for head-mounted display removal. In *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pp. 109–116. IEEE, 2021. 2

[26] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3d face model for pose and illumination invariant face recognition. In *2009 sixth IEEE international conference on advanced video and signal based surveillance*, pp. 296–301. Ieee, 2009. 2, 3

[27] S. Qian, T. Kirschstein, L. Schoneveld, D. Davoli, S. Giebenhain, and M. 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. 1, 2, 3, 4, 6, 8

[28] J. Rekimoto, K. Uragaki, and K. Yamada. Behind-the-mask: A face-through head-mounted display. In *Proceedings of the 2018 international conference on advanced visual interfaces*, pp. 1–5, 2018. 1, 2

[29] M. Takemura and Y. Ohta. Diminishing head-mounted display for shared mixed reality. In *Proceedings. International Symposium on Mixed and Augmented Reality*, pp. 149–156. IEEE, 2002. 2

[30] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2387–2395, 2016. 2

[31] M. Wang, X. Wen, and S.-M. Hu. Faithful face image completion for hmd occlusion removal. In *2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pp. 251–256. IEEE, 2019. 2

[32] J. Xu, S. Motamed, P. Vaddamanu, C. H. Wu, C. Haene, J.-C. Bazin, and F. De la Torre. Personalized face inpainting with diffusion models by parallel visual attention. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 5432–5442, 2024. 2

[33] Y. Xu, B. Chen, Z. Li, H. Zhang, L. Wang, Z. Zheng, and Y. Liu. Gaussian head avatar: Ultra high-fidelity head avatar via dynamic gaussians. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1931–1941, 2024. 2, 3, 4

[34] Y. Xu, H. Zhang, L. Wang, X. Zhao, H. Huang, G. Qi, and Y. Liu. Latentavatar: Learning latent expression code for expressive neural head avatar. In *ACM SIGGRAPH 2023 Conference Proceedings*, pp. 1–10, 2023. 1, 2, 3, 5, 6, 8

[35] Y. Yang and X. Guo. Generative landmark guided face inpainting. In *Pattern Recognition and Computer Vision: Third Chinese Conference, PRCV 2020, Nanjing, China, October 16–18, 2020, Proceedings, Part I 3*, pp. 14–26. Springer, 2020. 2

[36] Y. Zeng, J. Fu, and H. Chao. Learning joint spatial-temporal transformations for video inpainting. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pp. 528–543. Springer, 2020. 2

[37] Y. Zhao, Q. Xu, W. Chen, C. Du, J. Xing, X. Huang, and R. Yang. Mask-off: Synthesizing face images in the presence of head-mounted displays. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 267–276. IEEE, 2019. 1, 2

[38] W. Zielonka, T. Bolkart, and J. Thies. Instant volumetric head avatars. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4574–4584, 2023. 2