RGBAvatar: Reduced Gaussian Blendshapes for Online Modeling of Head Avatars

Linzhou Li Yumeng Li Yanlin Weng Youyi Zheng Kun Zhou*
State Key Lab of CAD&CG, Zhejiang University

Figure 1. Our RGBAvatar reconstructs a high-fidelity head avatar from a 2-minute monocular video in about 80 seconds, using a reduced set of Gaussian blendshapes. These blendshapes are linearly combined to generate avatar animations in real time at about 400 FPS.

Abstract

We present Reduced Gaussian Blendshapes Avatar (RGBAvatar), a method for reconstructing photorealistic, animatable head avatars at speeds sufficient for on-the-fly reconstruction. Unlike prior approaches that utilize linear bases from 3D morphable models (3DMM) to model Gaussian blendshapes, our method maps tracked 3DMM parameters into reduced blendshape weights with an MLP, leading to a compact set of blendshape bases. The learned compact base composition effectively captures essential facial details for specific individuals, and does not rely on the fixed base composition weights of 3DMM, leading to enhanced reconstruction quality and higher efficiency. To further expedite the reconstruction process, we develop a novel color initialization estimation method and a batch-parallel Gaussian rasterization process, achieving state-of-the-art quality with training throughput of about 630 images per second. Moreover, we propose a local-global sampling strategy that enables direct on-the-fly reconstruction, immediately reconstructing the model as video streams in real time while achieving quality comparable to offline settings. Our source code is available at https://github.com/gapszju/RGBAvatar.

1. Introduction

Recent advances in 3D Gaussian Splatting (3DGS) [27] have significantly improved the reconstruction of animatable head avatars. Prior methods typically bind 3D Gaussians to parametric mesh models [29, 37, 39, 46, 55] or learn neutral deformation fields [17, 44, 49, 50]. While Gaussian Blendshapes [33] have recently demonstrated superior reconstruction quality and real-time performance through efficient linear blending, they rely on predefined blendshape bases from 3D Morphable Models (3DMM) like FLAME [2] or FaceWarehouse [12]. This reliance on predefined bases results in a large number of parameters, scaling linearly with the blendshape count, leading to increased computational cost and slower training.

In this paper, we introduce RGBAvatar, a reduced Gaussian Blendshapes representation for animated head avatar reconstruction. Our key innovation is to implicitly learn a compact set of Gaussian blendshapes that effectively represent animatable head avatars. We achieve this through a lightweight MLP that maps tracked FLAME parameters to reduced blendshape weights, which are then used to linearly blend a series of learnable bases to generate an animated head avatar. By optimizing the MLP alongside the blendshapes during training, the model dynamically discovers a compact, adaptive base composition, leading to enhanced reconstruction quality and higher efficiency (both in training and runtime).

To further expedite training, we introduce a novel color-initialization method and a batch-parallel Gaussian rasterization process. Specifically, we observe that initialized Gaussians can be treated as Gaussian kernels in 2D image space, allowing direct calculation of Gaussian properties us-

*Corresponding author (kunzhou@acm.org)

ing convolution instead of gradient descent, thus speeding up convergence with better initial estimates. Additionally, since head avatars usually require fewer than 100k Gaussians [33, 46], traditional Gaussian rasterization underutilize GPU resources. We develop a batch-parallel Gaussian rasterization to optimize GPU utilization by making the 3DGS renderer fully batch-parallel with only one GPU synchronization per training step. The acceleration enables us to achieve a training throughput of 630 images per second, reconstructing high-quality head avatars in just about 80 seconds.

With our compact representation and acceleration strategy, we significantly reduce the time required to reconstruct high-quality animatable avatars, making our method feasible for on-the-fly head avatar reconstruction. Online reconstruction allows for immediate visual feedback during data capture, which is particularly desirable in interactive applications and scenarios requiring iterative design adjustments. While our method achieves fast processing speeds, online reconstruction introduces additional challenge of sequential data arrival, thus limiting access to the entire dataset during optimization. To address this, we design a local-global sampling strategy that balances rapid adaptation to new data and rehearsal on past data, mitigating forgetting.

Experimental results demonstrate that our model captures subtle expression details using as few as 20 blendshapes and achieves nearly 400 FPS rendering speed on consumer-grade GPUs, while attaining reconstruction quality close to offline methods. Evaluations on publicly available monocular video datasets show that our approach reconstructs high-fidelity head avatar details and outperforms state-of-the-art Gaussian-based methods in both runtime and training speed.

We summarize our main contributions as follows:

- We introduce Reduced Gaussian Blendshapes which efficiently captures subtle facial expressions while reducing the number of bases required to represent a head avatar.
- We develop a novel color-initialization method and a batch-parallel Gaussian rasterization process, significantly enhancing training throughput and optimizing GPU utilization.
- We establish the first on-the-fly head avatar reconstruction framework with a local-global sampling strategy, enabling real-time, online reconstruction with quality comparable to offline methods, providing immediate feedback crucial for interactive applications.

2. Related Work

2.1. 3D Representations for Head Avatars

Researchers have proposed various 3D representations for head avatars. The early and seminal work [9] proposes 3DMM to model head shape and texture in low-dimensional principal component analysis (PCA) spaces. There are many works [12, 31, 36, 41, 43] along this line to improve the representation ability of 3DMM. The explicit mesh is also used for reconstructing riggable head avatars from image or video inputs [3, 8, 16, 23, 24, 26, 28, 38]. Recently, NHA [24] uses two networks to predict vertex offsets and textures of underlying mesh. FLARE [8] further learns a relightable and animatable avatar through physically-based differentiable rendering.

Thanks to the success of neural radiance fields (NeRF) [5–7, 34, 35], head avatars with implicit representation [1, 2, 4, 20–22, 22, 25, 47, 48, 52, 54, 56, 57, 59] have achieved impressive results. Among them, i3DMM [52] presents the first deep implicit 3DMM of full heads. IMAvatar [56] used signed distance functions (SDF) to depict an implicit head model. HeadNeRF [25] integrates NeRF to the parametric representation of the head avatar. PointAvatar [57] proposes a deformable point-based representation to overcome the limitation of explicit mesh and implicit NeRF modeling. INSTA [59] models a dynamic neural radiance field based on neural graphics primitives embedded around a parametric face model. It is able to reconstruct a head avatar in about ten minutes. NeRFBlendShape [22] express the head model with a set of disentangled and interpretable bases, which can be driven by expression coefficients. Besides, AvatarMav [47] decouples expression motion by neural voxels in a similar way.

Recently, 3DGS [27] has made great progress in scene reconstruction, which inspires a series of works. MonoGaussianAvatar [17] replaces the point cloud in PointAvatar [57] with 3D Gaussians to improve rendering quality. GaussianHead [44] leverages tri-plane to store appearance related attributes. MLP based deformation field are used in [44, 50] for animation control. Gaussian Head Avatar [49] adopts a super-resolution network to achieve high-fidelity head avatar rendering. With the interpretable nature of 3DGS, some works [37, 39, 46, 55] binds the Gaussians onto the template mesh for easier head expression controllability. Among them, GaussianAvatars [37] optimize mesh jointly for better image alignment. FlashAvatar [46] use an MLP for further adding dynamic spatial offset to Gaussians. Another line of works [19, 33, 60] use a group of blending bases to model the head avatar. HeadGaS [19] blends learnable latent features and decode color and opacity of Gaussians from them with an MLP. GEM [60] compresses a reconstructed Gaussian avatar into eigen bases. GaussianBlendshapes [33] directly blends all the Gaussian attributes and ensures the sematic consistency with mesh blendshapes, which achieves superior reconstruction quality and rendering performance. Our method follows this line of works but with a reduced blendshape model that adapts to the individual rather than maintaining the same sematic with generic 3DMM blendshapes. We demon-

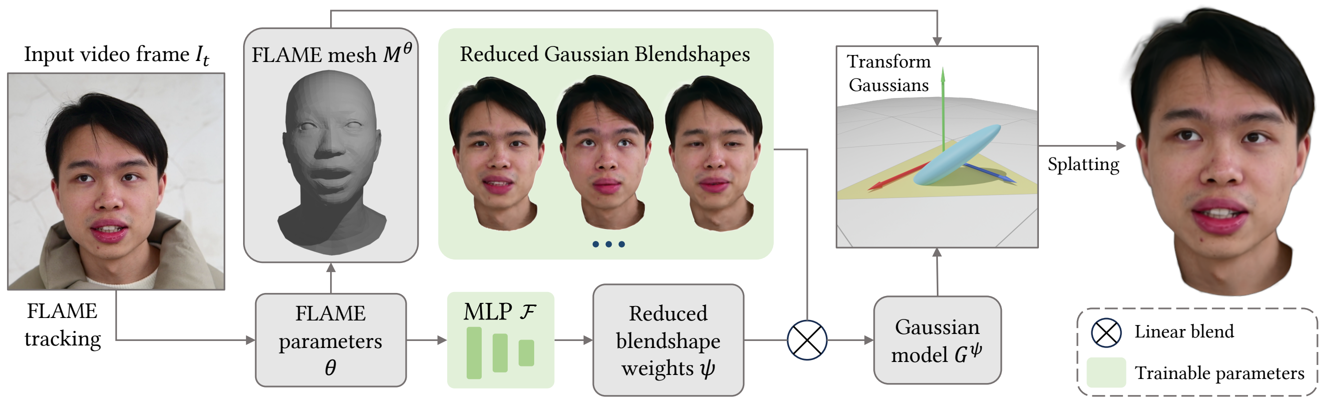

Figure 2. **Pipeline.** RGBAvatar represents the head avatar with a base model G_0 and a reduced set of Gaussian blendshapes $\{\Delta G_i\}_{i=1}^{K}$, each parametized as Gaussian attributes. For an input video frame I_t, we first track the FLAME parameters θ and generate FLAME mesh M^θ. Then, an MLP $\mathcal{F}$ is used to map the FLAME parameters θ to the reduced blendshape weights ψ. The Gaussian model G^ψ of the animated avatar is generated through linear blending with ψ. Finally, Gaussians are transformed into the deformed space for rendering according to the deformation of mesh triangles.

strate our approach delivers better reconstruction quality and higher efficiency with more compact representation.

2.2. Real-time Facial Animation

Weise et al. [45] firstly captures facial performances in real-time by fitting a parametric blendshape model to RGB-D data based on a commodity depth sensor. Follow up works [10, 18, 30, 61] focused on corrective shapes [30], dynamic facial expression space [10] and non-rigid mesh deformation [18, 61]. Although these works demonstrated impressive results, they rely on depth data which is typically unavailable in most video footage. Cao et al. [11] presents a real-time facial animation system that fits a blendshape model from 2D video frame. DDE [13] uses a generic regressor to get rid of calibration for each individual user. Cao et al. [14] further regresses fine-scale face wrinkles. Face2Face [40] captures pixel-level face performance. Our method builds upon these real-time facial animation trackers, and achieves on-the-fly photorealistic reconstruction of head avatar using 3D Gaussians.

3. Method

3.1. Reduced Gaussian Blendshape Representation

Previous blendshape modeling approaches, whether NeRF-based [22] or Gaussian-based [33], are all bound to template 3DMM blendshapes. To simplify the animation control, these methods directly use the tracked 3DMM parameters to drive their avatar model, requiring a one-to-one mapping with the 3DMM blendshapes. However, traditional 3DMM models, such as FLAME [2] based on PCA or FaceWareHouse [12] based on Facial Action Coding System (FACS), use a large number of bases to capture facial expression. To balance performance and memory usage, previous methods [19, 22, 33] typically truncate the number of blendshape bases to around 50, which constrains expressive capability.

In contrast, our reduced blendshape model leverages the full FLAME parameters, representing the head avatar with a compact set of blendshape bases that maintain both high fidelity and easy controllability. Specifically, our avatar model comprises a base Gaussian model G_0 and a set of Gaussian blendshapes $\Delta G_1, \Delta G_2, \ldots, \Delta G_K$, as shown in Fig. 2. Each model consists of Gaussians parameterized by position $\mathbf{x}$, rotation $\mathbf{q}$, scale $\mathbf{s}$, opacity α, and color $\mathbf{c}$. We generate arbitrary expressions through linear blending:

$$G^\psi = G_0 + \sum_{k=1}^{K} \psi_k \Delta G_k, \tag{1}$$

where $\psi \in \mathbb{R}^K$ represents the reduced blendshape weights. A shallow MLP $\mathcal{F}$ maps the tracked FLAME parameters $\theta \in \mathbb{R}^H$ (including expression coefficients and joint poses) to ψ:

$$\psi = \mathcal{F}(\theta). \tag{2}$$

To leverage geometric priors from the FLAME mesh M^θ, we further transform the blended Gaussian model G^ψ according to the mesh deformation, like [46]. Specifically, we define the Gaussian attributes in tangent space. For any frame t, we compute the TBN (tangent, bi-tangent, normal) matrices for each mesh triangle. Using these TBN matrices and the correspondences between Gaussians and mesh triangles, we transform the Gaussians from tangent space to the deformed space for rendering.

Our method achieves high-quality results with as few as $K = 20$ blendshapes (as shown in Fig. 3), delivering superior runtime efficiency compared to previous works. As

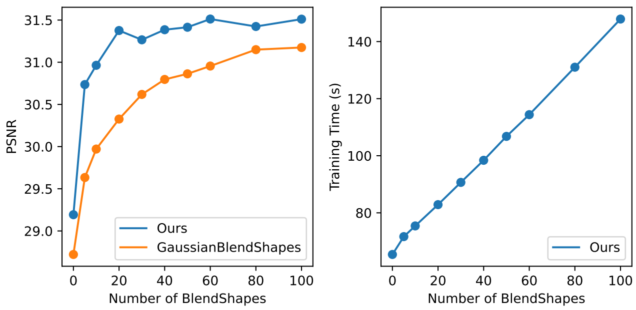

Figure 3. **Impact of number of blendshapes on reconstruction quality (left) and training time (right)**. We select 20 blendshapes to balance reconstruction quality with efficiency. Our method outperforms GaussianBlendShapes [33] using fewer number of blendshapes. Experiments are conducted on INSTA [59] dataset.

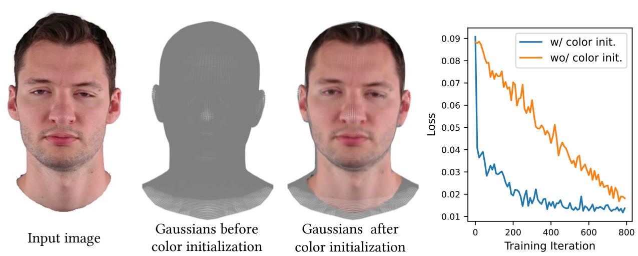

Figure 4. **Effect of color initialization.** This strategy is only applied once for each Gaussian during optimization. Left: results of our color initialization. Right: our color initialization strategy accelerates convergence speed in the early stage of training.

we adaptively reorganize arbitrary input into our reduced blendshape space, our model is also compatible with various 3DMM trackers, providing a robust, subject-adaptive avatar representation.

3.2. Training

Initialization. To facilitate online reconstruction, we aim to sufficiently utilize the information from the 3DMM prior. Specifically, we parametrize Gaussians in tangent space of the 3DMM, with positions initialized by sampling across the UV map of the 3DMM for an even distribution over the whole mesh, similar to FlashAvatar [46]. The rest of the Gaussian attributes are initialized with a default value for the base Gaussian model, while the Gaussian blendshapes $\{\Delta G_i\}$ model the residual and are initialized as zero. Gaussians will remain on the same triangle they are initialized with throughout the training process.

To further accelerate convergence, we propose a *color initialization* scheme that directly estimates Gaussian color attributes based on the observed pixel colors, as shown in Fig. 4. We notice that each Gaussian can be seen as a Gaussian kernel when projected into 2D pixel space, thus a natural first estimation with their color can be done with a convolution. The color of each Gaussian is computed as:

$$\mathbf{c}^{\text{init}} = \frac{\sum_{i=0}^{H}\sum_{j=0}^{W} w_{ij}\mathbf{I}_{ij}}{\sum_{i=0}^{H}\sum_{j=0}^{W} w_{ij}}, \tag{3}$$

where W and H denote the image width and height, and $\mathbf{I}_{ij}$ and w_{ij} are the RGB values and Gaussian splatting weights of pixel (i, j), respectively. This is performed only the first time each Gaussian's splatting weight w exceeds a threshold $\delta = 0.1$, skipping initialization for Gaussians that are not sufficiently visible to avoid poor estimations. This one-time initialization provides a more accurate starting value which speeds up convergence.

Loss Function. We apply a L_1 color loss to optimize the avatar's reconstruction quality, with random background colors to help keep the Gaussians constrained within the head region.

Batch-parallel Gaussian Rasterization. Previous methods [33, 37, 39, 46] often achieve suboptimal GPU utilization during training, typically below 60%, as they follow the original 3DGS training scheme of optimizing one sample per step (see Fig. 5a). Given that head avatars typically involve fewer than 100k Gaussians, we identified that the bottleneck lies not in GPU computation but in frequent GPU-CPU synchronizations.

To address this, we optimize a batch of samples in one step. For Gaussian rendering, a naive approach is to rasterize images in a batch sequentially like [17], but this method still requires GPU-CPU synchronization between `preprocess` and `rasterize` calculations (see Fig. 5b) because memory allocation for `rasterize` calculation

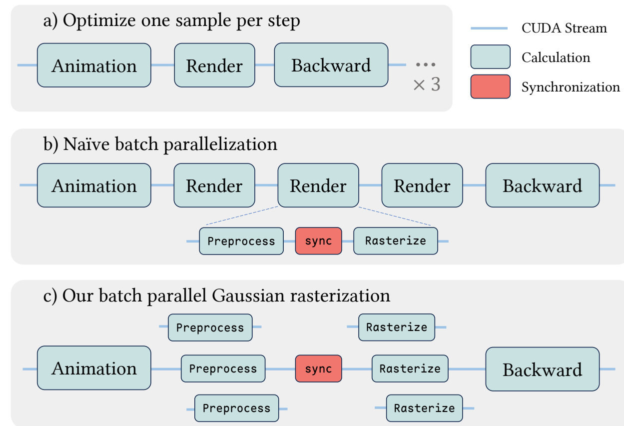

Figure 5. **Illustration of our batch-parallel Gaussian rasterization.** We take batch size = 3 as an example. a) Optimizing a single data sample per step, as used in [33, 39, 46], results in suboptimal GPU utilization. b) The naive batch-parallel approach, as used in [17], suffers from frequent synchronization overhead. c) Our batch-parallel Gaussian rasterization mitigates synchronization issues and maximizes parallelism by leveraging CUDA streams.

depends on the results of `preprocess` calculation. However, theoretically only one synchronization is needed per batch, as the GPU rasterization depends on the Gaussian preprocessing outcome regardless of batch size. Thus, we split Gaussian Splatting into two stages and perform synchronization only once after `preprocess`.

Additionally, since samples in a batch are fully parallelizable in both Gaussian preprocessing and rasterization stages, we assign them to different CUDA Streams to maximize computation, as shown in Fig. 5c. With this batch-based training scheme, we achieve 100% Stream Processor utilization during reconstruction, effectively mitigating the GPU-CPU synchronization bottleneck. Note that gsplat [51] also supports batched Gaussian rendering. However, it is limited to rendering the same set of Gaussians from different camera views, whereas our method allows rendering different sets of Gaussians from the same camera view.

3.3. Online Reconstruction Process

With our compact representation and efficient training scheme, our model is able to process input frames in a speed faster than real-time recording (25 FPS), which makes on-the-fly reconstruction possible. However, applying this to an online setting introduces challenges beyond convergence speed, particularly in managing streaming data. Unlike offline reconstruction, which has complete access to the dataset, online reconstruction operates on a continuous data stream. This setup creates two somewhat conflicting demands: (1) we need fast convergence on the newly arriving, previously unseen data to achieve *quick adaptation*, and (2) we must mitigate *catastrophic forgetting* of past data. To balance these goals, we introduce a *global-local sampling strategy*. This strategy maintains a local sampling pool $\mathcal{M}_l$ for newly introduced, under-optimized frames and a sampling pool $\mathcal{M}_g$ for historical frames. By sampling from both pools in each batch, the gradient calculations incorporate both recent and past information, effectively balancing rapid adaptation to new data with the retention of previously learned knowledge, thereby mitigating forgetting.

Online Optimization. During each optimization step, we split the batch size B into a local batch B_l and a global batch B_g according to a predefined ratio η. We then sample B_l from the local memory $\mathcal{M}_l$ and B_g from the global memory $\mathcal{M}_g$, optimizing the combined batch B_l, B_g via gradient descent. In our setup, we set $\eta = 0.7$, with $|\mathcal{M}_l| = 150$ and $|\mathcal{M}_g| = 1000$, prioritizing recent frames in the sampling process.

Processing Incoming Frames. For each incoming frame, we first extract the 3DMM parameters [13] and mask the background [32] using off-the-shelf methods. The new data sample is then added to $\mathcal{M}_l$ in a First-In-First-Out manner. When $\mathcal{M}_l$ reaches capacity, the oldest sample is removed and added to $\mathcal{M}_g$ using Reservoir Sampling [42], ensuring each frame in the data stream has an equal probability of being retained in the global pool. Please refer to supplementary materials for detailed algorithm.

4. Experiments

4.1. Setup

Baselines. We compare our method against five state-of-the-art Gaussian-based head avatar reconstruction approaches: SplattingAvatar [39] and GaussianAvatars [37], which embed Gaussians on mesh surfaces; FlashAvatar [46], which further models spatial offsets of 3D Gaussians using a MLP; MonoGaussianAvatar [17], which learns a Gaussian deformation field for avatar animation; and GaussianBlendshapes [33], which optimizes a set of Gaussian blendshape bases driven by FLAME blendshape weights.

Datasets. We evaluate our method on eight videos from the INSTA Dataset [59] and four videos from the GaussianBlendshapes Dataset [33]. All videos are cropped and resized to 512^2 resolution, with sequence lengths between 2000 and 5000 frames. Consistent with previous work [59], we reserve the last 350 frames of each video for self-reenactment testing. Backgrounds are removed using Robust Video Matting [32], and each frame's FLAME parameters are extracted via an offline tracker [58]. Please refer to supplementary materials for implementation details.

4.2. Comparisons

We evaluate our results using standard metrics in animatable avatar reconstruction, including PSNR, SSIM, and LPIPS [53]. As shown in Tab. 1, our method outperforms baselines in PSNR and SSIM across most cases, particularly in challenging scenarios such as the marcel video. Note that MonoGaussianAvatar [17] utilizes LPIPS loss during training which leads to a better LPIPS result. We show that our method achieves superior results if we add the LPIPS loss with a weight of 0.05 in training.

To better demonstrate our effectiveness, we recorded two additional videos with expressive expressions for a qualitative evaluation. Each contains 3k frames and the last 1.5k are held out as test set. As shown in Fig. 6, our method produces photorealistic rendering of wrinkles, hair, mouth and eye regions. SplattingAvatar [39] and GaussianAvatars [37] struggle to capture expressive facial details, as their Gaussian transformations are entirely driven by mesh deformation, limiting their expressiveness. MonoGaussianAvatar [17] and FlashAvatar [46] use MLPs to model Gaussian deformation but fail to capture high-frequency details, such as wrinkles, due to the smoothness bias of MLPs, even with extended training time (*e.g.* 9h in MonoGaussianAvatar). GaussianBlendshapes [33] can capture wrinkle-level details with 50 blendshape bases, but our method outperforms them in capturing finer details. Specifically, by leveraging all 100

Datasets		INSTA dataset								GaussianBlendShapes dataset			
		bala	biden	justin	malte_1	marcel	nf_01	nf_03	wojtek_1	subject1	subject2	subject3	subject4
PSNR ↑	SplattingAvatar	30.99	31.24	31.90	28.08	25.65	27.52	27.80	31.48	31.30	32.52	31.96	34.69
	GaussianAvatars	31.58	31.20	31.80	27.60	25.93	27.55	28.04	32.10	31.03	33.35	32.30	34.51
	FlashAvatar	31.98	23.76	31.50	28.41	23.19	26.89	26.97	31.60	29.67	32.61	30.19	32.29
	MonoGaussianAvatar	28.71	32.56	32.13	26.37	23.64	28.48	31.00	31.29	33.72	33.67	33.13	33.67
	GaussianBlendShapes	33.21	32.69	32.73	28.38	26.49	27.99	28.58	32.76	33.14	33.38	32.30	34.93
	Ours	33.89	32.93	32.99	29.54	27.72	28.87	30.79	32.77	34.85	34.53	34.47	35.77
	Ours (w/ LPIPS)	32.75	32.80	33.16	29.22	27.57	28.81	30.74	33.15	35.08	34.64	34.17	35.96
	Ours (online)	32.94	32.60	32.79	29.27	27.28	28.89	30.46	32.65	34.92	33.97	34.36	35.58
SSIM ↑	SplattingAvatar	0.9222	0.9561	0.9606	0.9344	0.9085	0.9285	0.9179	0.9505	0.9250	0.9506	0.9211	0.9615
	GaussianAvatars	0.9457	0.9593	0.9657	0.9424	0.9281	0.9422	0.9333	0.9622	0.9338	0.9635	0.9397	0.9673
	FlashAvatar	0.9154	0.8670	0.9481	0.9290	0.8962	0.9259	0.9069	0.9460	0.9115	0.9474	0.9134	0.9425
	MonoGaussianAvatar	0.9315	0.9681	0.9667	0.9417	0.9232	0.9416	0.9458	0.9612	0.9513	0.9677	0.9449	0.9631
	GaussianBlendShapes	0.9455	0.9658	0.9672	0.9455	0.9270	0.9474	0.9352	0.9640	0.9428	0.9607	0.9418	0.9693
	Ours	0.9549	0.9720	0.9721	0.9522	0.9388	0.9481	0.9474	0.9665	0.9549	0.9708	0.9507	0.9721
	Ours (w/ LPIPS)	0.9498	0.9713	0.9715	0.9501	0.9361	0.9472	0.9458	0.9660	0.9533	0.9690	0.9474	0.9706
	Ours (online)	0.9509	0.9699	0.9705	0.9502	0.9333	0.9464	0.9430	0.9655	0.9553	0.9699	0.9518	0.9718
LPIPS ↓	SplattingAvatar	0.1160	0.0599	0.0688	0.0841	0.1654	0.1333	0.1216	0.0773	0.1265	0.1022	0.1546	0.0944
	GaussianAvatars	0.0699	0.0538	0.0605	0.0701	0.1389	0.1153	0.0941	0.0533	0.1028	0.0779	0.1264	0.0745
	FlashAvatar	0.0853	0.1518	0.0745	0.0687	0.1620	0.1124	0.1189	0.0628	0.0893	0.0611	0.0969	0.0881
	MonoGaussianAvatar	0.0668	0.0370	0.0494	0.0591	0.1241	0.1025	0.0811	0.0431	0.0797	0.0478	0.0892	0.0579
	GaussianBlendShapes	0.0833	0.0537	0.0647	0.0722	0.1477	0.1210	0.0992	0.0589	0.1066	0.0838	0.1260	0.0705
	Ours	0.0663	0.0408	0.0518	0.0586	0.1249	0.1098	0.0865	0.0507	0.0847	0.0613	0.1026	0.0567
	Ours (w/ LPIPS)	0.0430	0.0330	0.0428	0.0434	0.0991	0.0919	0.0695	0.0323	0.0491	0.0361	0.0536	0.0347
	Ours (online)	0.0666	0.0430	0.0534	0.0609	0.1315	0.1135	0.0906	0.0529	0.0820	0.0561	0.0996	0.0545

Table 1. **Quantitative comparisons.** We compare our approach with five Gaussian-based state-of-the-art methods. The best and second results are highlighted.

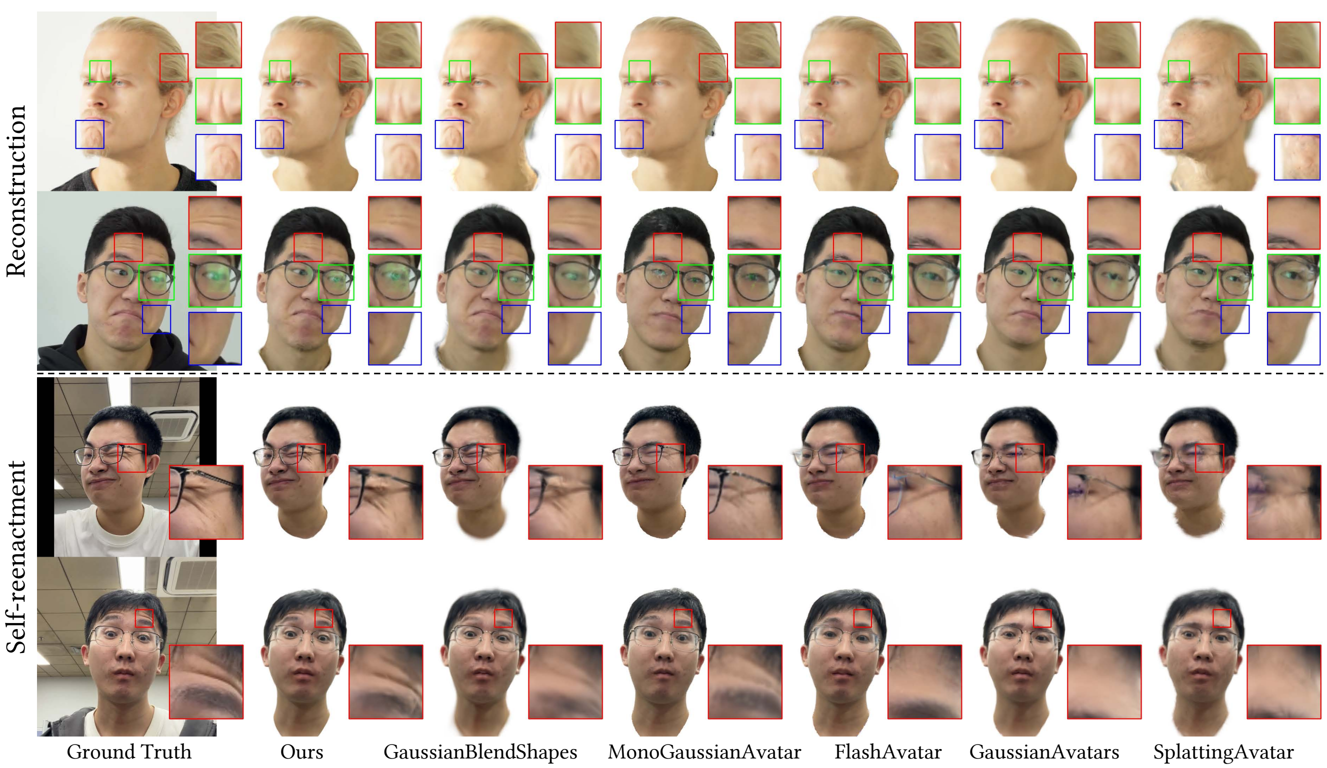

Figure 6. **Qualitative comparisons.** Our method produces photorealistic rendering of wrinkles, mouth, hair and eye regions. Compared to GaussianBlendShapes [33], which uses 50 blendshape bases, our approach requires only 20 bases while capturing even finer details such as reflections on eyeglasses and deep wrinkles.

Method	Training Time	Rendering FPS
SplattingAvatar	21min	300
GaussianAvatars	14min	177
FlashAvatar	17min	304
MonoGaussianAvatar	9h	17
GaussianBlendShapes	20min	267
Ours	**81s**	**398**

Table 2. **Performance comparisons.** We report the training time and run time FPS on RTX 3090. Note that we train GaussianAvatars [37] for 40k iterations for monocular setting. And the run time calculation includes both rendering and animation.

expression coefficients, we capture large facial expressions, such as deep wrinkles (4th and 5th rows in Fig. 6). Our method can also utilize pose rotations to improve performance on side views, capturing details like hair strands (1st row), eyeglass frames (4th row), and reflections on glass (2nd row), all of which are strongly correlated with pose.

Tab. 2 shows the training and runtime performance comparisons. Our method reconstructs a head avatar within about 80 seconds — 400 times faster than MonoGaussianAvatar and 10 times faster than GaussianAvatars. For rendering, our method is also the most efficient, achieving 398 FPS. Note that our runtime includes both rendering and animation calculation (compute the FLAME mesh, linearly blend the Gaussian attributes, and transform Gaussians with mesh deformation).

4.3. Blendshape Visualization

Fig. 7 visualizes the 10 out of 20 learned blendshapes for two subjects. We observe that each blendshape corresponds to a specific expression, with distinct semantic variations across individuals. In this experiment, we position the Gaussians in world space rather than tangent space to provide clearer visualization.

4.4. Applications

Fig. 8 demonstrates cross-identity reenactment, where tracked coefficients from a target avatar (top row) are used to drive a source avatar (bottom row). Our adaptive blendshapes are fully compatible with 3DMM, allowing for straightforward exchange of expression parameters to achieve seamless facial reenactment. In Fig. 9, we render the avatars from four novel viewpoints, distinct from the training views. Our model consistently maintains expressions across these different perspectives.

4.5. Ablation Studies

Reduced Blendshapes. As shown in Fig. 10, we evaluate the effectiveness of reducing blendshapes. In the first setup, we directly use the truncated FLAME expression coefficients as the weights to drive the blendshape model (*wo/ reducing*). In the second setup, we use an MLP to map the full FLAME parameters to reduced blendshape weights (*w/ reducing*). For the self-reenactment task, we observe that the *w/ reducing* setting achieves significantly better results, even with a lesser number of blendshapes.

Figure 7. **Blendshape visualization.** We show 10 out of 20 blendshapes learned by our method for two subjects. The blendshapes are diverse and adapted to each subject.

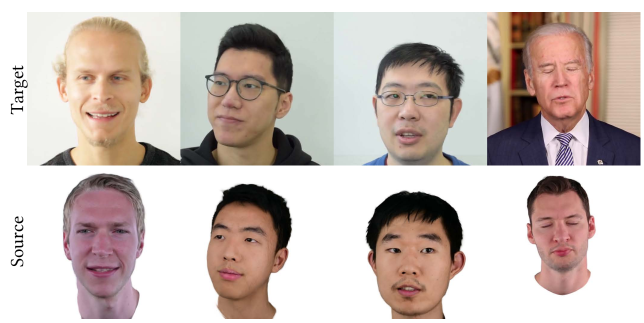

Figure 8. **Cross-identity reenactment.** Our reduced blendshapes can be driven by FLAME parameters easily. We achieve natural results on reenactment task.

Number of Blendshapes. In Fig. 3, we examine the effect of the number of reduced blendshapes on reconstruction quality and training efficiency. Our model achieves nearly optimal quality with 20 blendshapes, offering a balance between quality and efficiency as training time increases linearly with the number of blendshapes. Compared

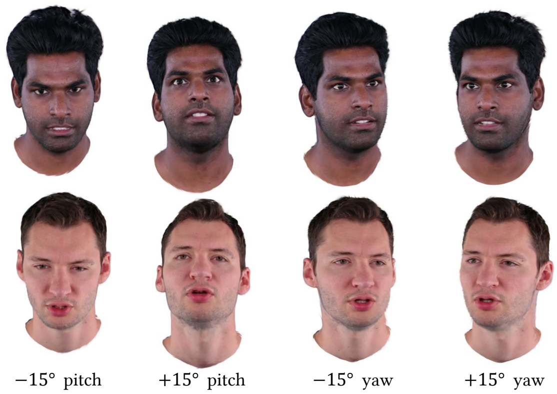

Figure 9. **Results of novel view extrapolation.**

Method	PSNR	SSIM	LPIPS
w/o global	30.26	0.9492	0.0801
w/o local	30.46	0.9508	0.0788
Ours	**30.86**	**0.9537**	**0.0765**

Table 3. **Ablation study on online reconstruction sampling strategy.** Experiments are conducted on INSTA dataset.

Method	Throughput	GPU Utilization
w/o batch parallel	151	52%
naive batch parallel	463	94%
Ours	**630**	**100%**

Table 4. **Ablation study on training scheme.**

to GaussianBlendshapes [33], our model performs better at any given number of blendshapes, whereas GaussianBlendshapes requires around 80 blendshapes to reach maximum quality.

Batch-Parallel Rasterization. In Tab. 4, we assess training throughput and GPU utilization under different schemes. Throughput is defined as frames processed per second. With a naive batch-parallel approach, throughput increases by $3\times$, and with our optimized batch-parallel rasterization process, throughput further improves by $4.2\times$, reaching 100% GPU utilization.

Online Reconstruction Strategy. In Tab. 3, we compare our online training strategy (*Ours*) to two variants: one without a global sampling pool (*w/o global*) and one without a local sampling pool (*w/o local*). Our method outperforms both alternative settings.

5. Limitation and Discussion

As shown in Fig. 11, our constructed avatar models can exhibit apparent artifacts in large novel-pose rendering that are far from training poses, a common problem for head avatar

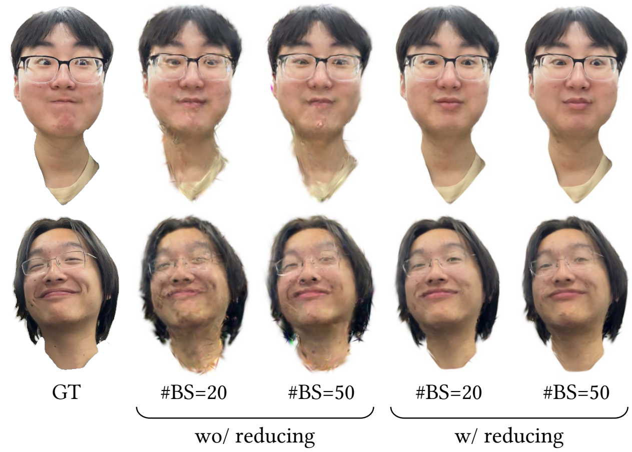

Figure 10. **Ablation study on reduced blendshapes.** The reduced blendshapes significantly improve the quality under novel expression. `#BS` indicates the number of blendshape bases.

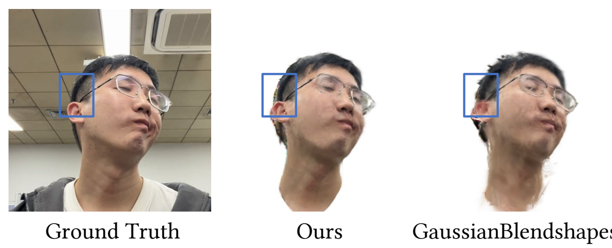

Figure 11. **Limitations.** Our approach produces suboptimal results under large novel-pose.

reconstruction methods such as GaussianBlendshapes [33]. It is worth noting that there is a risk of misuse of our method (e.g., the so-called DeepFakes). We strongly oppose applying our work to produce fake images or videos of individuals with the intention of spreading false information or damaging their reputations.

6. Conclusion

We introduce Reduced Gaussian Blendshapes for real-time photorealistic animated avatar reconstruction from monocular video streams. Our blendshape representation is adaptive to the subject, not aligned with 3DMM blendshapes, and has better reconstruction quality with fewer blendshapes. Besides, we introduce a color initialization method and a batch-parallel Gaussian rasterization to speed up the training process, enabling our method to support on-the-fly avatar reconstruction. Experiments show that our method outperforms prior arts both in quality and efficiency.

Acknowledgments

This work is supported by NSF China (No. 62172357 & 62421003) and the XPLORER PRIZE. We thank Bohong Chen and Qing Chang for being our capture subjects.

References

[1] ShahRukh Athar, Zexiang Xu, Kalyan Sunkavalli, Eli Shechtman, and Zhixin Shu. Rignerf: Fully controllable neural 3d portraits. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 20364–20373, 2022. 2

[2] ShahRukh Athar, Zhixin Shu, and Dimitris Samaras. Flame-in-nerf: Neural control of radiance fields for free view face animation. In *2023 IEEE 17th International Conference on Automatic Face and Gesture Recognition (FG)*, pages 1–8. IEEE, 2023. 1, 2, 3

[3] Ziqian Bai, Zhaopeng Cui, Xiaoming Liu, and Ping Tan. Riggable 3d face reconstruction via in-network optimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6216–6225, 2021. 2

[4] Ziqian Bai, Feitong Tan, Zeng Huang, Kripasindhu Sarkar, Danhang Tang, Di Qiu, Abhimitra Meka, Ruofei Du, Mingsong Dou, Sergio Orts-Escolano, et al. Learning personalized high quality volumetric head avatars from monocular rgb videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16890–16900, 2023. 2

[5] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5855–5864, 2021. 2

[6] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5470–5479, 2022.

[7] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19697–19705, 2023. 2

[8] Shrisha Bharadwaj, Yufeng Zheng, Otmar Hilliges, Michael J Black, and Victoria Fernandez-Abrevaya. Flare: Fast learning of animatable and relightable mesh avatars. *arXiv preprint arXiv:2310.17519*, 2023. 2

[9] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 157–164. 2023. 2

[10] Sofien Bouaziz, Yangang Wang, and Mark Pauly. Online modeling for realtime facial animation. *ACM Transactions on Graphics (ToG)*, 32(4):1–10, 2013. 3

[11] Chen Cao, Yanlin Weng, Stephen Lin, and Kun Zhou. 3d shape regression for real-time facial animation. *ACM Transactions on Graphics (TOG)*, 32(4):1–10, 2013. 3

[12] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425, 2013. 1, 2, 3

[13] Chen Cao, Qiming Hou, and Kun Zhou. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Transactions on graphics (TOG)*, 33(4):1–10, 2014. 3, 5, 1

[14] Chen Cao, Derek Bradley, Kun Zhou, and Thabo Beeler. Real-time high-fidelity facial performance capture. *ACM Transactions on Graphics (ToG)*, 34(4):1–9, 2015. 3

[15] Qing Chang, Yao-Xiang Ding, and Kun Zhou. Enhancing identity-deformation disentanglement in stylegan for one-shot face video re-enactment. In *The 39th Annual AAAI Conference on Artificial Intelligence*, 2024. 4

[16] Bindita Chaudhuri, Noranart Vesdapunt, Linda Shapiro, and Baoyuan Wang. Personalized face modeling for improved face reconstruction and motion retargeting. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 142–160. Springer, 2020. 2

[17] Yufan Chen, Lizhen Wang, Qijing Li, Hongjiang Xiao, Shengping Zhang, Hongxun Yao, and Yebin Liu. Monogaussianavatar: Monocular gaussian point-based head avatar. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–9, 2024. 1, 2, 4, 5

[18] Yen-Lin Chen, Hsiang-Tao Wu, Fuhao Shi, Xin Tong, and Jinxiang Chai. Accurate and robust 3d facial capture using a single rgbd camera. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3615–3622, 2013. 3

[19] Helisa Dhamo, Yinyu Nie, Arthur Moreau, Jifei Song, Richard Shaw, Yiren Zhou, and Eduardo Pérez-Pellitero. Headgas: Real-time animatable head avatars via 3d gaussian splatting. *arXiv preprint arXiv:2312.02902*, 2023. 2, 3

[20] Hao-Bin Duan, Miao Wang, Jin-Chuan Shi, Xu-Chuan Chen, and Yan-Pei Cao. Bakedavatar: Baking neural fields for real-time head avatar synthesis. *ACM Transactions on Graphics (TOG)*, 42(6):1–17, 2023. 2

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

[22] Xuan Gao, Chenglai Zhong, Jun Xiang, Yang Hong, Yudong Guo, and Juyong Zhang. Reconstructing personalized semantic facial nerf models from monocular video. *ACM Transactions on Graphics (TOG)*, 41(6):1–12, 2022. 2, 3

[23] Pablo Garrido, Michael Zollhöfer, Dan Casas, Levi Valgaerts, Kiran Varanasi, Patrick Pérez, and Christian Theobalt. Reconstruction of personalized 3d face rigs from monocular video. *ACM Transactions on Graphics (TOG)*, 35 (3):1–15, 2016. 2

[24] Philip-William Grassal, Malte Prinzler, Titus Leistner, Carsten Rother, Matthias Nießner, and Justus Thies. Neural head avatars from monocular rgb videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18653–18664, 2022. 2

[25] Yang Hong, Bo Peng, Haiyao Xiao, Ligang Liu, and Juyong Zhang. Headnerf: A real-time nerf-based parametric head model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20374–20384, 2022. 2

[26] Alexandru Eugen Ichim, Sofien Bouaziz, and Mark Pauly. Dynamic 3d avatar creation from hand-held video input. *ACM Transactions on Graphics (ToG)*, 34(4):1–14, 2015. 2

[27] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 1, 2

[28] Taras Khakhulin, Vanessa Sklyarova, Victor Lempitsky, and Egor Zakharov. Realistic one-shot mesh-based head avatars. In *European Conference on Computer Vision*, pages 345–362. Springer, 2022. 2

[29] Jaeseong Lee, Taewoong Kang, Marcel C Bühler, Min-Jung Kim, Sungwon Hwang, Junha Hyung, Hyojin Jang, and Jaegul Choo. Surfhead: Affine rig blending for geometrically accurate 2d gaussian surfel head avatars. *arXiv preprint arXiv:2410.11682*, 2024. 1

[30] Hao Li, Jihun Yu, Yuting Ye, and Chris Bregler. Realtime facial animation with on-the-fly correctives. *ACM Trans. Graph.*, 32(4):42–1, 2013. 3

[31] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ACM Trans. Graph.*, 36(6):194–1, 2017. 2

[32] Shanchuan Lin, Linjie Yang, Imran Saleemi, and Soumyadip Sengupta. Robust high-resolution video matting with temporal guidance, 2021. 5

[33] Shengjie Ma, Yanlin Weng, Tianjia Shao, and Kun Zhou. 3d gaussian blendshapes for head avatar animation. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–10, 2024. 1, 2, 3, 4, 5, 6, 8

[34] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2

[35] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022. 2

[36] Stylianos Ploumpis, Evangelos Ververas, Eimear O'Sullivan, Stylianos Moschoglou, Haoyang Wang, Nick Pears, William AP Smith, Baris Gecer, and Stefanos Zafeiriou. Towards a complete 3d morphable model of the human head. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):4142–4160, 2020. 2

[37] Shenhan Qian, Tobias Kirschstein, Liam Schoneveld, Davide Davoli, Simon Giebenhain, and Matthias Nießner. Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20299–20309, 2024. 1, 2, 4, 5, 7, 3

[38] George Retsinas, Panagiotis P Filntisis, Radek Danecek, Victoria F Abrevaya, Anastasios Roussos, Timo Bolkart, and Petros Maragos. 3d facial expressions through analysis-by-neural-synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2490–2501, 2024. 2

[39] Zhijing Shao, Zhaolong Wang, Zhuang Li, Duotun Wang, Xiangru Lin, Yu Zhang, Mingming Fan, and Zeyu Wang. Splattingavatar: Realistic real-time human avatars with mesh-embedded gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1606–1616, 2024. 1, 2, 4, 5, 3

[40] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2387–2395, 2016. 3

[41] Luan Tran and Xiaoming Liu. Nonlinear 3d face morphable model. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7346–7355, 2018. 2

[42] Jeffrey S Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1):37–57, 1985. 5, 1

[43] Daniel Vlasic, Matthew Brand, Hanspeter Pfister, and Jovan Popovic. Face transfer with multilinear models. In *ACM SIGGRAPH 2006 Courses*, pages 24–es. 2006. 2

[44] Jie Wang, Jiu-Cheng Xie, Xianyan Li, Feng Xu, Chi-Man Pun, and Hao Gao. Gaussianhead: High-fidelity head avatars with learnable gaussian derivation. *arXiv preprint arXiv:2312.01632*, 2023. 1, 2

[45] Thibaut Weise, Sofien Bouaziz, Hao Li, and Mark Pauly. Realtime performance-based facial animation. *ACM transactions on graphics (TOG)*, 30(4):1–10, 2011. 3

[46] Jun Xiang, Xuan Gao, Yudong Guo, and Juyong Zhang. Flashavatar: High-fidelity head avatar with efficient gaussian embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1802–1812, 2024. 1, 2, 3, 4, 5

[47] Yuelang Xu, Lizhen Wang, Xiaochen Zhao, Hongwen Zhang, and Yebin Liu. Avatarmav: Fast 3d head avatar reconstruction using motion-aware neural voxels. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–10, 2023. 2

[48] Yuelang Xu, Hongwen Zhang, Lizhen Wang, Xiaochen Zhao, Han Huang, Guojun Qi, and Yebin Liu. Latentavatar: Learning latent expression code for expressive neural head avatar. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–10, 2023. 2

[49] Yuelang Xu, Benwang Chen, Zhe Li, Hongwen Zhang, Lizhen Wang, Zerong Zheng, and Yebin 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*, pages 1931–1941, 2024. 1, 2

[50] Yixin Xuan, Xinyang Li, Gongxin Yao, Shiwei Zhou, Donghui Sun, Xiaoxin Chen, and Yu Pan. Faghead: Fully animate gaussian head from monocular videos. *arXiv preprint arXiv:2406.19070*, 2024. 1, 2

[51] Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, and Angjoo Kanazawa. gsplat: An open-source library for Gaussian splatting. *arXiv preprint arXiv:2409.06765*, 2024. 5

[52] Tarun Yenamandra, Ayush Tewari, Florian Bernard, Hans-Peter Seidel, Mohamed Elgharib, Daniel Cremers, and

Christian Theobalt. i3dmm: Deep implicit 3d morphable model of human heads. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12803–12813, 2021. 2

[53] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 5

[54] Xiaochen Zhao, Lizhen Wang, Jingxiang Sun, Hongwen Zhang, Jinli Suo, and Yebin Liu. Havatar: High-fidelity head avatar via facial model conditioned neural radiance field. *ACM Transactions on Graphics*, 43(1):1–16, 2023. 2

[55] Zhongyuan Zhao, Zhenyu Bao, Qing Li, Guoping Qiu, and Kanglin Liu. Psavatar: A point-based morphable shape model for real-time head avatar creation with 3d gaussian splatting. *arXiv preprint arXiv:2401.12900*, 2024. 1, 2

[56] Yufeng Zheng, Victoria Fernández Abrevaya, Marcel C Bühler, Xu Chen, Michael J Black, and Otmar Hilliges. Im avatar: Implicit morphable head avatars from videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13545–13555, 2022. 2

[57] Yufeng Zheng, Wang Yifan, Gordon Wetzstein, Michael J Black, and Otmar Hilliges. Pointavatar: Deformable point-based head avatars from videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21057–21067, 2023. 2

[58] Wojciech Zielonka, Timo Bolkart, and Justus Thies. Towards metrical reconstruction of human faces. In *European Conference on Computer Vision*, 2022. 5

[59] Wojciech Zielonka, Timo Bolkart, and Justus Thies. Instant volumetric head avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4574–4584, 2023. 2, 4, 5

[60] Wojciech Zielonka, Timo Bolkart, Thabo Beeler, and Justus Thies. Gaussian eigen models for human heads. *arXiv preprint arXiv:2407.04545*, 2024. 2

[61] Michael Zollhöfer, Matthias Nießner, Shahram Izadi, Christoph Rehmann, Christopher Zach, Matthew Fisher, Chenglei Wu, Andrew Fitzgibbon, Charles Loop, Christian Theobalt, et al. Real-time non-rigid reconstruction using an rgb-d camera. *ACM Transactions on Graphics (ToG)*, 33(4): 1–12, 2014. 3