

GASP: Gaussian Avatars with Synthetic Priors

Jack Saunders^{1,2}, Charlie Hewitt¹, Yanan Jian¹, Marek Kowalski¹, Tadas Baltrusaitis¹, Yiye Chen^{1,3}, Darren Cosker^{1,2}, Virginia Estellers¹, Nicholas Gydé¹, Vinay P. Namboodiri², and Benjamin E Lundell¹

¹Microsoft, ² University of Bath, ³ Georgia Tech

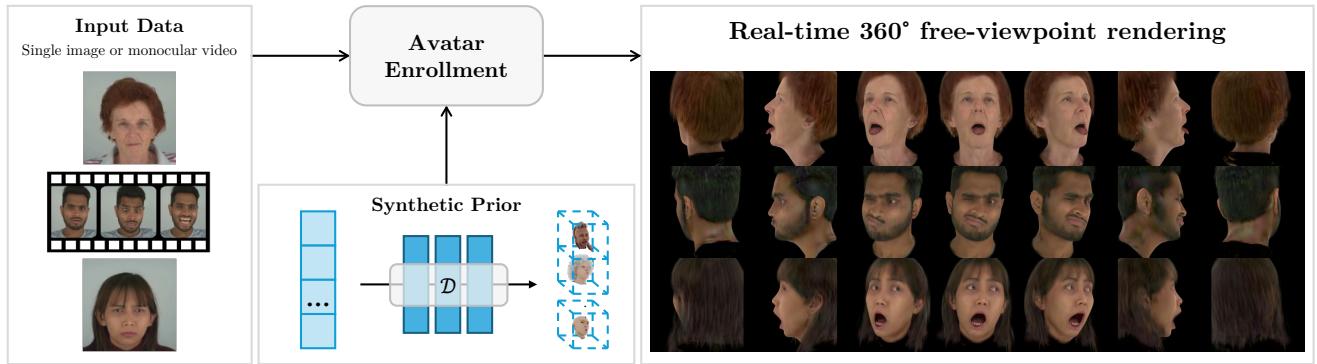


Figure 1. We propose **GASP**, a novel model for creating photorealistic, realtime, animatable, 360° avatars from easily-captured data. We train a generative prior model of Gaussian Avatars on Synthetic data. The prior allows our model to be fit using a single image or a short video with the prior accounting for the unseen views. This lets users create their avatar with only a webcam or smartphone.

Abstract

*Gaussian Splatting has changed the game for real-time photo-realistic rendering. One of the most popular applications of Gaussian Splatting is to create animatable avatars, known as Gaussian Avatars. Recent works have pushed the boundaries of quality and rendering efficiency but suffer from two main limitations. Either they require expensive multi-camera rigs to produce avatars with free-view rendering, or they can be trained with a single camera but only rendered at high quality from this fixed viewpoint. An ideal model would be trained using a short monocular video or image from available hardware, such as a webcam, and rendered from any view. To this end, we propose **GASP: Gaussian Avatars with Synthetic Priors**. To overcome the limitations of existing datasets, we exploit the pixel-perfect nature of synthetic data to train a Gaussian Avatar prior. By fitting this prior model to a single photo or video and fine-tuning it, we get a high-quality Gaussian Avatar, which supports 360° rendering. Our prior is only required for fitting, not inference, enabling real-time application. Through our method, we obtain high-quality, animatable Avatars from limited data which can be animated and rendered at 70fps on commercial hardware. See our project page ¹ for results.*

¹<https://microsoft.github.io/GASP/>

1. Introduction

Creating high-quality digital humans unlocks significant potential for many applications, including Virtual Reality, gaming, video conferencing, and entertainment. Digital humans must be photorealistic, easy to capture and capable of real-time rendering. The vision and graphics communities have long worked towards this goal, and we are rapidly approaching the point where such digital humans are possible.

A series of works based first on NeRFs [25] raised the bar in creating exceptional visual quality [2, 3, 6, 9, 19, 47]. However, NeRFs remain slow to render and are unsuitable for real-time applications. Gaussian Splatting-based works have led to significant improvements in both quality and rendering speed [4, 8, 15, 28, 33, 40, 41]. Despite these improvements, the list of suitable applications for these methods is small. Each of these models suffers from one of two drawbacks: either they require expensive capture setups with multiple synchronized cameras, which prevents easy user enrollment [8, 28, 41], or they train on a single camera but exhibit significant quality degradation when rendered from views with more than a minimal variation in camera pose [4, 33, 40]. Furthermore, to maximize visual qual-

ity, some of these methods use a large CNN *after* rendering, which prevents real-time rendering without a powerful GPU [8, 41].

For mass adoption, an avatar model should achieve high-quality 360° rendering in real-time and require only the amount of data a user can practically provide. In most cases a user can only capture a monocular, frontal image or video using their webcam or smartphone camera. The problem of fitting an avatar to this data is ill-posed; the extreme sides and back of the head are not visible, leading to artifacts in these unseen regions. In order to overcome this, we require a prior model that is able to “fill in the gaps” left by missing data. Such a model has been shown to be effective in other limited data human-centric models, such as visual dubbing [31] and static NeRF models [2]. Ideally, we would train such a model on a large, multi-view, perfectly annotated and diverse dataset. However, very few multi-view face datasets exist. Those that do either lack full coverage, particularly around the back of the head [19], or have only a small number of subjects [39]. Furthermore, annotations such as camera calibrations and 3D morphable model (3DMM) parameters associated with these datasets have to be estimated using imperfect methods and are a significant source of error.

We propose **GASP: Gaussian Avatars with Synthetic Priors**. We use a large, diverse dataset of *synthetic* humans [12, 38] to overcome the difficulties associated with training a prior on real data. This data is generated using computer graphics and has perfectly accurate annotations, including exact correspondence to the underlying 3DMM. This enables the large-scale training of a prior for Gaussian Avatars for the first time. However, the use of synthetic data introduces a domain gap problem. We address this by learning per-Gaussian features with semantic correlations. By learning these correlations on synthetic data and then maintaining them when fitting to real data, using a three-stage fitting process, we can cross this domain gap. Our method even enables rendering the back of the head, having fit to only a single front-facing image or video; see Fig. 1.

To summarize, we propose a novel system for creating realistic, real-time animatable avatars from a webcam or smartphone enabled by the following contributions:

- A prior model over Gaussian Avatar parameters trained using purely synthetic data.
- A three-stage fitting process, combined with learned per-Gaussian correlations to overcome the synthetic-to-real domain gap and allow for 360° rendering.
- Real-time rendering enabled through use of neural networks only during training and fitting, and not at inference time.

2. Related Work

2.1. Photorealistic Animatable Avatars

A significant number of works have attempted to build photorealistic 3D Avatars that can be animated. Most of these works use an existing animatable model, known as a 3D morphable model (3DMM) [1, 20]. Earlier works improve the realism of a 3DMM in image space using compositing [35], a CNN model [17] or pixel-level MLPs [24]. Some work [32, 37] improve the CNN models by adding a learnable latent texture known as a neural texture [36] and evaluating this with a deferred neural renderer. Other works make use of volumetric rendering, either in the form of a point-based representation [46], or a NeRF [9, 26, 43, 47]. Each of these methods has shown great potential but is too slow or too prone to artifacts to provide a complete solution.

Gaussian Splatting [15] has allowed for unprecedented photorealism and real-time capabilities in volumetric rendering. Unsurprisingly, this technology has been adapted for applications in the photorealistic avatar space. We refer to this class of methods as Gaussian Avatars. Most Gaussian Avatar methods have built upon 3DMMs as a coarse representation of the geometry and Gaussian Splatting for finer geometry and appearance. Qian et al. [28] do this by binding Gaussians individual triangles in the mesh. Xiang et al. [40] initialize Gaussians by sampling from a UV map and moving the Gaussians by barycentric interpolation of the posed meshes. They add a dynamic MLP that learns to introduce wrinkles based on the 3DMM expression blend weights. Chen et al. [4] learn an extension to the LBS function of FLAME [20] by extending the blendshape basis to apply per Gaussian. Xu et al. [41] model deformations and dynamics with MLPs and apply a CNN-based super-resolution network. Giebenhain et al. [8] do similar but attach the Gaussians to an implicit geometry model using cycle consistency. These methods can produce photorealistic avatars with real-time rendering and impressive quality, but either train on a single camera and evaluate on the same camera or allow novel view synthesis but only within a small range and rely on multiple cameras. A similar method to our work is the recent Gaussian Morphable Model of Xu et al. [42], which can fit to a single input image. However, as most of the training data used is from the front half of the head, it cannot produce results in the back of the head and as a new task-specific expression basis is learned, control with an existing 3DMM is impossible.

2.2. Few-Shot Avatars

Several other works have attempted to address a similar problem to ours, in which the goal is to create a photorealistic avatar from limited amounts of data. In each case, the solution is to leverage some form of data-driven prior. Preface [3] uses a large-scale dataset to train an identity-

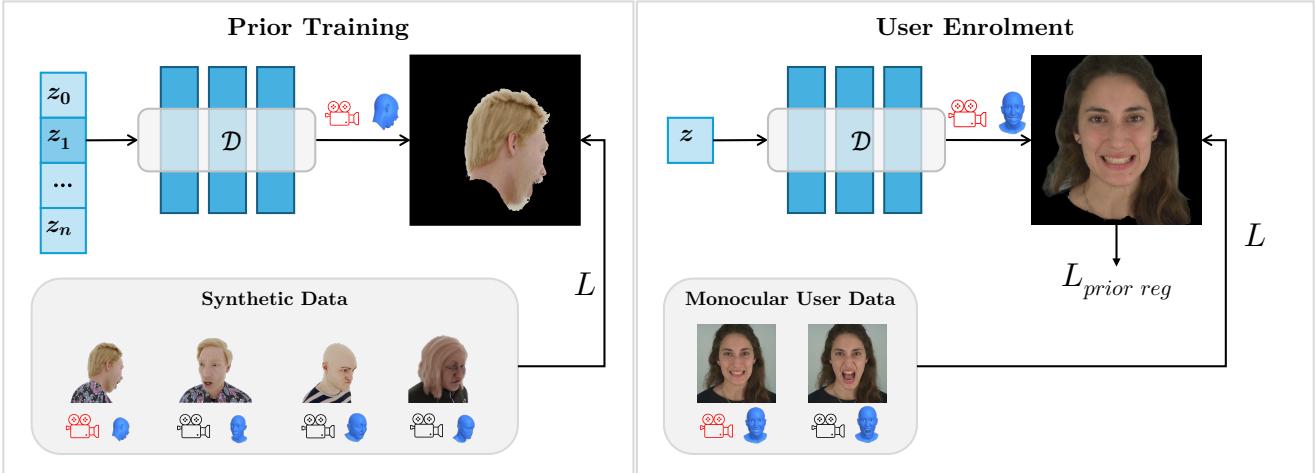


Figure 2. The overview of our model. In the first stage, we train an autodecoder prior model on Synthetic data to predict the parameters of a mesh attached Gaussian Avatar. We can then adapt this model to user enrollment data, either a single image or short monocular video. We leverage the prior to improve the quality in unseen regions and achieve free-viewpoint rendering.

conditioned NeRF prior model in an auto-decoder fashion. Cafca [2] also seeks to train this model on large-volume synthetic data. While high quality, the results are static and cannot be animated, and being NeRF-based, are also slow to render, taking over 20s per frame. Some works use powerful 2D image-space models as a prior, exploiting a small amount of data to enable control over the larger model with a 3DMM. StyleRig [34] first achieves control over StyleGAN2 [14] in this way, and DiffusionRig [5] obtains even better results using a DDPM [13] as a prior. Dubbing for Everyone [31] uses a StyleGAN-based UNET with personalized Neural Textures, which allows for better adaptation. ROME [16] takes a similar approach, with neural textures predicted from images. However, as they operate at the image level, they cannot model the back and sides of the head.

3. Method

3.1. Background: Gaussian Splatting

3D Gaussian Splatting is a method for reconstructing a volume from a set of images with corresponding camera calibrations. It involves using a collection of Gaussian primitives, represented by a position μ in 3D space, an anisotropic covariance matrix Σ , a color c and an opacity α . Kerbl et al. [15] proposed a system to optimize these parameters to fit the evidence provided by the images by decomposing the covariance Σ into the scale, σ , and rotation, r , components, represented as a vector and quaternion respectively. Following projection by the camera and depth sorting, each pixel color P is computed as:

$$P = \sum_{i=1}^{N_G} c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j) \quad (1)$$

Since the whole process is differentiable, the Gaussian Attributes can be optimized to match the given images and camera parameters.

3.2. Background: Mesh Attached Gaussians

Gaussian Splatting is excellent at reconstructing static scenes but, in its basic form, cannot model animation dynamics. Multiple works [28, 33] make the observation that, given a sufficiently good coarse approximation of geometry in the form of a mesh, the problem can be reduced to an approximately static scene. By attaching each Gaussian, \mathcal{G}_i , to a specific triangle, t , in the mesh, the Gaussian is assumed to remain static relative to that triangle's pose. There are several successful formulations of this posing transformation:

$$\mu, \sigma, r = \mathcal{T}_{local \rightarrow global}(\mu', \sigma', r' | t) \quad (2)$$

For our purposes, we use the definition of Qian et al. [28], where the origin of each triangle's system is assumed to be the center, the orthonormal basis is determined by one edge, the triangle's normal and their cross-product, and the isotropic scale by the mean of the length of one edge and its perpendicular in the triangle. This allows us to define a Gaussian Avatar, \mathcal{G} , as a collection of static Gaussian primitives in a triangle-local space.

$$\mathcal{G} = \{\mathcal{G}_i : 1 \leq i \leq N_G\}, \mathcal{G}_i = \{\mu'_i, \sigma'_i, r'_i, c_i, o_i\} \quad (3)$$

As these are static, we can optimize them using the same procedures as in the original formulation [15].

3.3. Prior Model Training

We train our prior model as a generative model over identities. Following previous work [7, 27, 29, 42], we train this prior as an auto-decoder model. We jointly learn a per-subject identity code, $\mathbf{z}_j \in \mathbb{R}^{512}, j \in \{1, \dots, N_{id}\}$, and an MLP decoder, $\mathcal{D}(\mathbf{z})$. One may naively think of training this model to directly output the Gaussian Attributes, \mathcal{A} , with a single MLP. However, such a method quickly becomes intractable. As a typical model with 100,000 Gaussians may have millions of attributes, the number of parameters in \mathcal{D} would be too large. Instead, we augment each Gaussian with a learnable feature vector, $\mathbf{f}_i \in \mathbb{R}^8, i \in \{1, \dots, N_G\}$. This feature is analogous to a positional encoding with additional semantic meaning. We then train a network to map these per-Gaussian features to Gaussian attributes, with each Gaussian processed independently and in parallel.

To make optimization more stable, we learn a Canonical Gaussian Template, \mathcal{C} , and model the per-person variation as offsets from this template. The Canonical Template can be considered the mean Avatar. The i -th Gaussian of the avatar for the subject j is given by:

$$\mathcal{A}_{i,j} = \mathcal{C}_{i,j} + \mathcal{D}(\mathbf{f}_i, \mathbf{z}_j) \quad (4)$$

This is best understood by following Fig. 3. To train this model, we jointly optimize \mathcal{C} , \mathcal{D} , $\{\mathbf{z}_j\}_{1 \leq j \leq N_{id}}$, $\{\mathbf{f}_i\}_{1 \leq i \leq N_G}$ to minimize the following loss function:

$$\mathcal{L} = \lambda_{pix} L_{pix} + \lambda_\alpha L_\alpha + \lambda_{percep} L_{percep} + L_{reg} \quad (5)$$

Where L_{pix} is a pixel level loss consisting of L_1 , the ℓ_1 difference between the real and predicted images, and L_{SSIM} which is the differentiable SSIM loss, weighted by λ_1 and λ_{SSIM} respectively. L_{percep} is a perceptual loss based on LPIPS [45], L_α is the ℓ_1 distance between the real and predicted alpha masks, and L_{reg} is a regularization loss acting on the Gaussians. We regularize scale and displacement:

$$L_{reg} = \lambda_\sigma ||\max(0.6, \sigma')||_2 + \lambda_\mu ||\mu'||_2 \quad (6)$$

Unlike previous methods, our 3DMM does not capture course hair, meaning the Gaussians must model it. We, therefore, reduce λ_μ by a factor of 100 for Gaussians bound to faces in the scalp region, which we manually define.

3.4. Initialization

Using just one Gaussian per triangle face of the 3DMM leads to an under-parameterised model that lacks sufficient detail. To overcome this, we use the initialization strategy of Xiang et al. [40]. We generate a UV map of a given resolution for our mesh and take each pixel's corresponding face and barycentric coordinates. The face is used for Gaussian binding. We use the barycentric coordinates to position the origin of each Gaussian's local coordinate system.

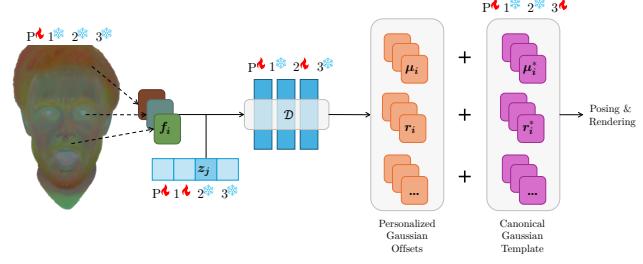


Figure 3. The architecture of our prior model. A latent vector for identity is used to transform learnable per-Gaussian features into Gaussian Attributes, which offset a canonical template. Our training process has four stages: the prior training, P, and three user-specific fitting steps. We freeze some layers and train others at each stage, as indicated.

3.5. Fitting Process

Given input data ranging from a single image, to a short video from a single monocular camera, we aim to produce a high-quality avatar that can be viewed from any angle. We have three stages to this fitting process, visualized in Fig. 3:

1. We find the best in-prior Gaussian Avatar by randomly initializing an identity latent vector, \mathbf{z} , and optimizing this with everything else frozen; we call this inversion.
2. We fine-tune the MLP, \mathcal{D} , with the rest of the model frozen.
3. We refine the resulting Gaussians using the standard Gaussian Splatting optimization procedure [15] to best fit the data.

To motivate this three-step process, we can consider two extremes. On the one hand, we could perform inversion only. This relies heavily on the prior. If we had perfectly diverse real-human data and a perfect prior, this may be all we would need to do. However, our prior training was on synthetic data, so we could only generate synthetic-looking avatars with this method. On the other hand, we could use the prior for initialization and then optimize the resulting Gaussians. This would achieve similar results to the existing state-of-the-art but with the unseen regions looking synthetic.

We can extract more value from our prior model by considering correlations in the per-Gaussian features, \mathbf{f} . Our network is forced to map these to Gaussian attributes and learns to associate similar Gaussians with similar features. Fig. 4 shows a PCA decomposition of the Gaussian features, demonstrating that these features have learned semantic meaning. By freezing the features in the fitting process, Gaussians with similar semantic features will be mapped to have similar attributes. For example, if \mathcal{D} learns to make a Gaussian representing hair at the front of the head blonde, it will also update an unseen one at the back of the head.

To prevent stages 2 and 3 from diverging too far from the

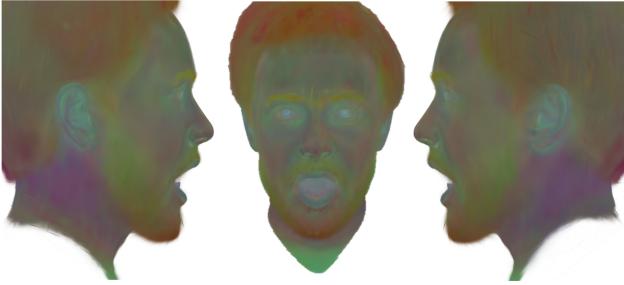


Figure 4. Visualization of the first three components of a PCA decomposition of the Gaussian features f , displayed using the geometry of a random subject. Note the semantic relationships.

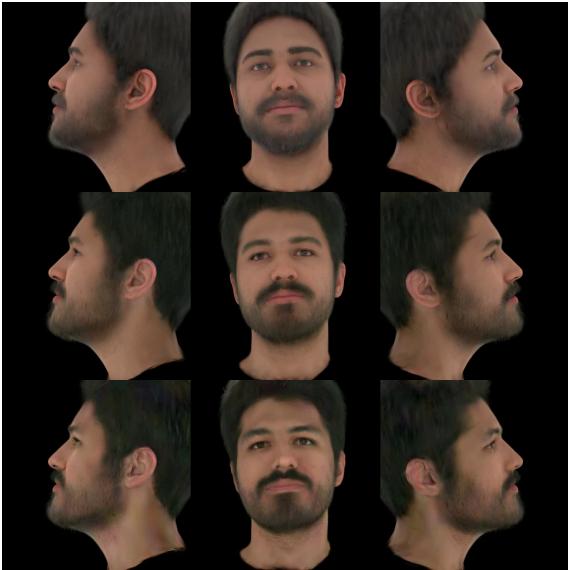


Figure 5. Examples showing how the three stages in our fitting process resolve the domain gap of the synthetic prior. Stage 1 (Top) optimizes within the prior, Stage 2 (Middle) finetunes the MLP, \mathcal{D} , and Stage 3 (Bottom) refines the individual Gaussians. Note the beard and eyes.

prior, we introduce an additional regularization term, L_{prior} , to the loss, \mathcal{L} , during these stages. L_{prior} is defined as the ℓ_2 distance between each Gaussian attribute and its corresponding value from the prior (i.e., after stage 1). This is particularly important when regularizing unseen Gaussians. Results after each stage of fitting are shown in Fig. 5.

4. Dataset

We require calibrated multi-camera data of the same subject performing a wide range of expressions to train our prior model. Collecting such data would require complex and expensive camera rigs. Instead, we leverage a synthetic data generation pipeline of Hewitt et al. [12]. This allows us to generate highly diverse and perfectly calibrated image data



Figure 6. Examples from our synthetic dataset. We generate a large and diverse set of synthetic subjects rendered from many views to train our prior model.

with pixel-perfect annotations. We generate 1000 identities (random face shape, texture, upper body clothing, hairstyle, hair color, and eye color). We illuminate the scene using uniform white lighting to simplify model training. We pose those faces with random expressions and sample a virtual camera uniformly from a hemisphere ($[-180, +180]$ degrees azimuth and $[-20, +45]$ degrees elevation) to render 50 images per identity. Examples of the data used in training our prior is shown in Fig. 6.

5. Implementation Details

Identity codes, z , and Gaussian features, f , are 512 and 8 dimensional, respectively. We initialize with a UV map of 512×512 pixels, resulting in 187,779 Gaussians. The supplementary details our decoder network \mathcal{D} 's architecture. For all parameters, we optimize using the Adam optimizer [18]. The canonical Gaussians are optimized using the learning rates from the original implementation of 3D Gaussian Splatting [15]. We optimize z and \mathcal{D} with a learning rate 0.0002. Prior network training took 4 days and was performed using 4timesA100's with a batch size of 8 for 250 epochs. The fitting process uses 500 steps for stages 1 and 2. We use 100 steps for stage 3. The whole fitting process takes 10 minutes on an NVIDIA Geforce 4090 RTX GPU.

6. Results

We conduct all of our evaluations on the NeRSemble Dataset [19]. This dataset contains multiple subjects performing dozens of facial expression sequences, including one freeform sequence, across 16 cameras. For each sequence, we preprocess each video using an off-the-shelf background removal [21] and face segmentation tool [44] to get the head region only. We obtain Morphable Model parameters in the format of Wood et al. [38] using the method of Hewitt et al. [12]. We consider three experimental settings using this data; please refer to the supplementary ma-

Method	Monocular Video						Single Image					
	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	FID \downarrow	ID-SIM \uparrow	QUAL \uparrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	FID \downarrow	ID-SIM \uparrow	QUAL \uparrow
FlashAvatar	17.25	0.603	0.450	351	0.234	2.08	13.26	0.490	0.519	367	0.057	2.05
GaussianAvatars	17.39	0.601	0.428	366	0.179	2.08	14.80	0.474	0.475	385	0.000	2.03
ROME*	-	-	-	-	-	-	15.78	0.543	0.441	136	0.408	3.38
DiffusionRig	19.67	0.343	0.436	155	0.302	2.98	16.87	0.316	0.541	183	0.239	3.15
Ours	21.34	0.712	0.333	117	0.568	3.68	20.73	0.677	0.348	119	0.526	3.80

Table 1. Quantitative Evaluations: We compare our method with three state-of-the-art models. We evaluate on two scenarios, for the Monocular scenario we on a single camera and then evaluate on the four most extreme. For single image we do the same but using only the first image from the Monocular sequence. In each case the evaluation sequence is unseen in the training set. We take the average PSNR, SSIM and LPIPS scores for each frame of each avatar. We also ask for user ratings of the quality of each method and report the mean scores out of 5 (QUAL). (*) ROME only supports single image use cases. We highlight the Best and Second Best for each metric.

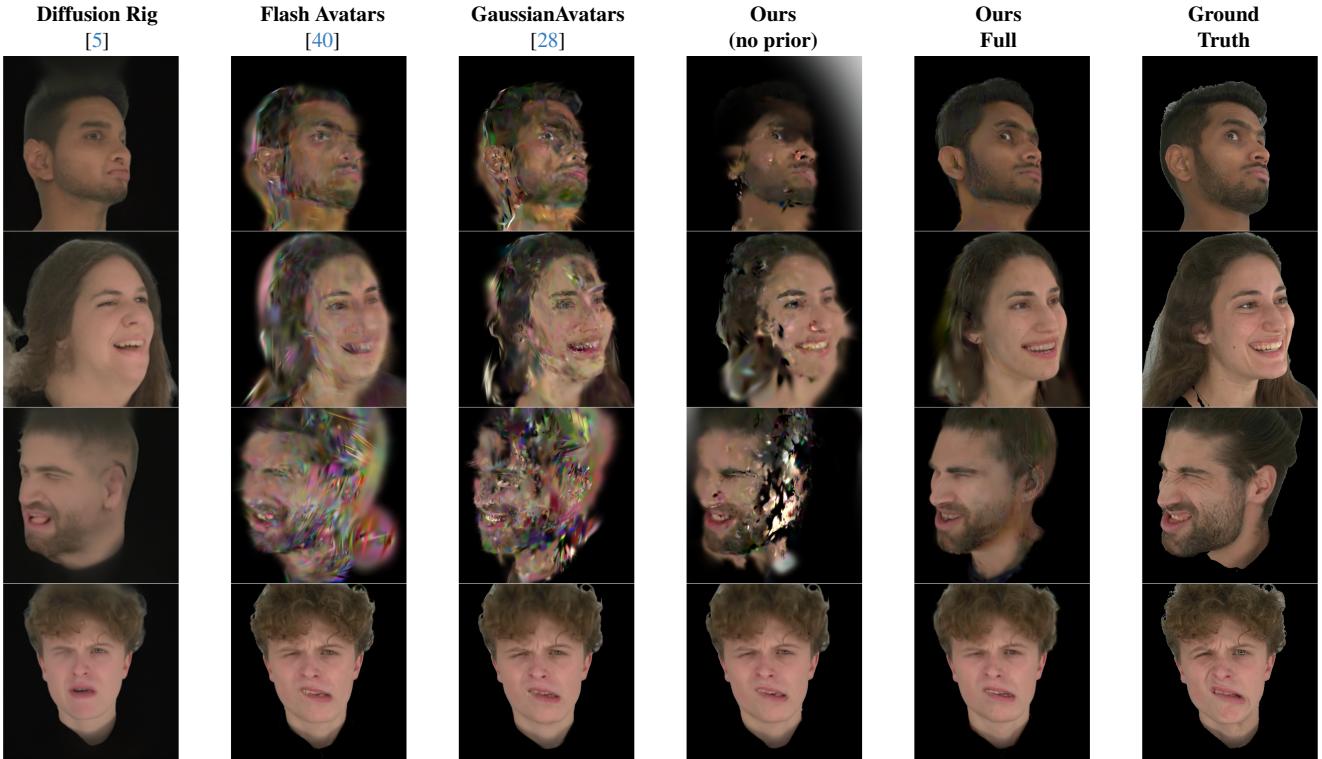


Figure 7. Qualitative comparisons of our method with existing state-of-the-art in the Monocular Setting. We train on a monocular camera and evaluate on unseen camera poses (top three rows) and an unseen sequence from the training view (bottom row). Our model captures identity better than Diffusion Rig [5] and suffers from fewer artifacts than other Gaussian Avatar models ([28, 40], ours without a prior)

Method	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	FID \downarrow	ID-SIM \uparrow	QUAL \uparrow
FlashAvatar	24.73	0.815	0.253	125	0.767	3.70
GaussianAvatars	23.73	0.812	0.285	113	0.773	3.65
DiffusionRig	19.42	0.377	0.425	155	0.302	3.00
Ours	23.44	0.786	0.261	101	0.734	3.80

Table 2. Multi-Camera: We run comparisons using 16 training cameras. We report the mean user ratings out of 5 (QUAL). We highlight the Best and Second Best for each metric.

terial for the cameras and sequences used:

Monocular: To best replicate our desired setting, we en-

roll all avatars using a single frontal camera. We use a subset of the expression sequences for fitting and evaluate them using the unseen freeform sequence. We use the four most extreme view cameras for evaluation, as determined by manual inspection, to test the model’s ability to produce good results on regions unseen at training.

Multi-Camera: To confirm that our model does not sacrifice performance when more data is available, we also enroll avatars using the same configuration above but with all cameras used for input.

Single Image: To test the limits of our model, we also

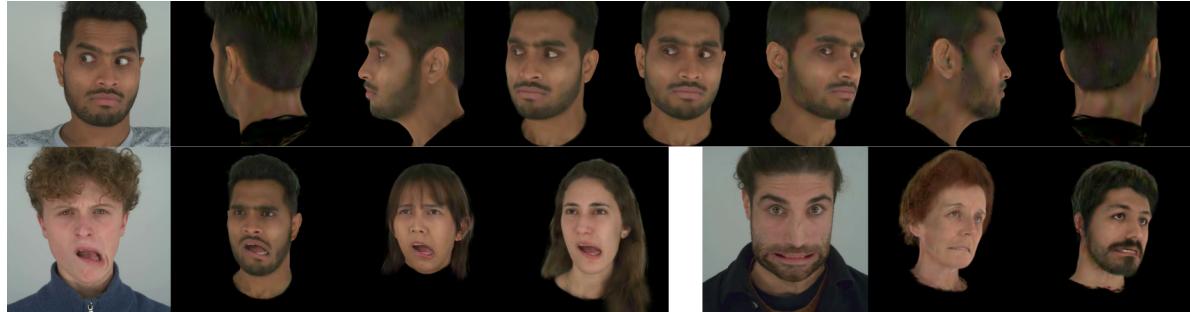


Figure 8. Self/Cross Reenactment: We show examples of our model for self-reenactment (top) and cross-identity (bottom). The model is fit using a frontal view video only (frame with a gray background). Despite never seeing the back of a real person’s head, we still obtain good-quality results (frames with a black background). More examples are in Fig. 1 and the supplementary.

experiment with just a single image as input, selecting the first frame from the Monocular setting as input.

To evaluate visual quality, we use the standard metrics PSNR, SSIM, LPIPS [45] and FID [11]. We find that PSNR and SSIM prefer solutions that match low-frequency detail, e.g. a flat sheet of hair. While FID is better at capturing high-frequencies. We also conducted a user study to measure perceived quality most accurately. We ask users to rate each video out of five and report the mean scores; we denote this QUAL. More details can be found in the supplementary.

6.1. Baselines

We compare our model to state-of-the-art methods. For the first set of methods, we look at Gaussian Avatar models: Gaussian Avatars [28], which is designed for ultra-high quality rendering when trained on multiple views, and Flash Avatars [40], which is designed to be trained and evaluated on monocular data. We train these using the same morphable model, 3DMM fitting process and dataset preprocessing as our method. In addition to Gaussian Avatar models, we look at models designed for few-shot animatable avatar synthesis. We select the publicly available implementations of ROME [16] and DiffusionRig [5].

6.2. Monocular Training

The results of this experiment can be found in Table 1. Our model significantly outperforms state-of-the-art across all metrics, including user-perceived quality. Our model produces significantly fewer artifacts in novel views compared to other Gaussian Avatar methods [28, 40]. This is because our prior helps prevent the model from overfitting to the training camera view. Diffusion Rig [5] does not show any visible artifacts, but struggles to preserve the identity of the subject, this is best seen in Fig. 7.

6.3. Multi-Camera Training

Our model is competitive with the state-of-the-art in the Multi-Camera setting (Tab. 2). We expect our model to per-

form worse than other Gaussian Avatar methods [28, 40] as the prior regularizes the model towards a synthetic solution, and we do not model lighting or dynamic expressions. Despite this, our model performs similarly to the state-of-the-art, suggesting it can effectively use all available data. Furthermore, using the prior allows our model to converge in fewer steps than other Gaussian Avatar models, making it cheaper and more efficient to train. Our model performs better on all metrics compared to Diffusion Rig [5].

6.4. Single Image Training

The results of the single image setting are shown in Tab. 1. With such limited data, other Gaussian Avatar methods overfit and perform poorly. Even on the same camera view as the input image, Gaussian Avatar methods struggle with artefacts; see Fig. 9. Our method also outperforms ROME [16], which is designed to work with a single image.

6.5. Ablations

We perform an ablation study to demonstrate our model’s effectiveness. The results are shown in Tab. 3. We use three subjects in the monocular setting. More details, as well as additional qualitative results, are in the supplementary.

No Prior: To validate the use of the prior, we fit person-specific models using our MLP without any prior. We also ablate the use of the prior regularization loss term. It can be seen that the absence of the prior dramatically reduces the quality according to all metrics, while not regularizing towards the prior leads to slightly better ID reconstruction but worse quality according to all other metrics.

Number of Subjects: We compare the model quality using priors trained on differing numbers of subjects. The more subjects we have, the better the quality. Interestingly, using one subject in the prior performs worse than not using a prior. This contrasts with the findings of Buehler et al. [2].

Number of Gaussians: We consider the effect of initializing with fewer Gaussians. We use texture maps ranging from 64×64 up to the full 512×512 . We see that the

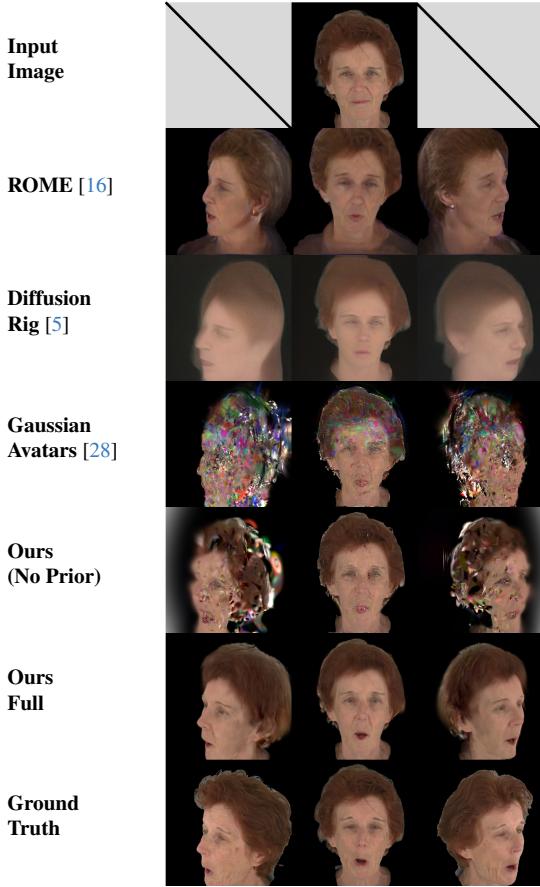


Figure 9. Qualitative comparisons of our method with existing state-of-the-art in the **Single Image Setting**, using the top image only for the fitting process.

highest resolution model performs best. Although the gain is small, it is notable visually (see supplementary).

Fitting Stages: We show the importance of each stage of the fitting process. Without stage 1 (optimizing for \mathbf{z}), our model performs worse on all metrics; this is also true for stage 2, although less pronounced. Without stage 3 our model performs similarly, or slightly better, visually, but suffers from a significant drop in ID reconstruction. The stages are seen visually in Fig. 5.

6.6. Runtime

After fitting a user’s Avatar using the prior, we can generate the mesh attached Gaussian Avatar parameters \mathcal{A} . Combined with the triangle face bindings and barycentric coordinates, this fully specifies an Avatar. No neural networks, including \mathcal{D} , are required for inference. A user’s Avatar can be stored as an approximately 15MB file. Without any runtime optimizations, the complete inference pass, from Morphable Model parameters to the final rendered image runs at 70fps on an NVIDIA 4090 RTX GPU. The posing

Method	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	FID \downarrow	ID-SIM \uparrow
w/o prior	19.42	0.670	0.391	212	0.478
w/o prior regularization	20.31	0.701	0.344	122	0.620
w/o stage 1	19.56	0.678	0.364	127	0.588
w/o stage 2	20.33	0.704	0.347	118	0.585
w/o stage 3	20.47	0.711	0.343	113	0.441
1 Prior Subject	15.86	0.550	0.459	274	0.365
10 Prior Subjects	19.98	0.678	0.367	146	0.538
100 Prior Subjects	20.39	0.703	0.347	129	0.577
64×64 Gaussians	20.41	0.709	0.363	155	0.493
128×128 Gaussians	20.45	0.704	0.353	127	0.567
256×256 Gaussians	20.43	0.702	0.350	117	0.575
Full (1k Subjects, 512×512)	20.67	0.716	0.340	108	0.589

Table 3. **Ablations:** We ablate several components of the model. We evaluate the absence of the prior, the effect of fewer subjects and fewer Gaussians. We also ablate each stage of the fitting process. We highlight the **Best** and **Second Best** for each metric.

of the Gaussians can run at 67fps on a 3rd Gen Intel(R) Core(TM) i9-13900K CPU, suggesting improvements in Gaussian Splatting may allow real-time CPU inference.

7. Limitations and Future Work

While our model is able to achieve high-quality 360° rendering, it has some limitations. For some regions, such as the back of the head, the model produces synthetic-looking results. We would like to address this issue by looking into 2D image-based priors [22, 23] based on diffusion models [30]. To reduce artefacts introduced by overfitting to the monocular view, we used only flat RGB colour and did not model lighting, reducing our model’s realism. In future, we may include a lighting model in our prior, enabled by a diverse set of lighting conditions in our synthetic data. As can be seen in our supplementary, our prior serves as a generative model with good interpretability. Given sufficient resources and a good camera/morphable model registration pipeline, we would like to use the findings of this work to train a similar generative prior using real data.

8. Conclusion

We have presented **GASP**, a novel method enabling 360°, high-quality Avatar synthesis from limited data. Our model builds a prior over Gaussian Avatar parameters to “fill in” missing regions. To bypass issues associated with collecting a large-scale real dataset, such as the need for full coverage and perfect annotation, we use synthetic data. Learned semantic Gaussian features and a three-stage fitting process enable us to cross the domain gap, while fitting to real data, to create realistic avatars. Our model outperforms the state-of-the-art in novel view and expression synthesis with Avatars trained from a single camera (e.g., a webcam or phone camera) using a short video or a single image while retaining the ability to animate and render in real-time.

References

- [1] Volker Blanz and Thomas Vetter. *A Morphable Model For The Synthesis Of 3D Faces*. Association for Computing Machinery, New York, NY, USA, 1 edition, 2023. 2
- [2] Marcel C. Buehler, Gengyan Li, Erroll Wood, Leonhard Helminger, Xu Chen, Tanmay Shah, Daoye Wang, Stephan Garbin, Sergio Orts-Escalano, Otmar Hilliges, Dmitry Lagun, Jérémie Rivière, Paulo Gotardo, Thabo Beeler, Abhimitra Meka, and Kripasindhu Sarkar. Cafca: High-quality novel view synthesis of expressive faces from casual few-shot captures. In *ACM SIGGRAPH Asia 2024 Conference Paper*. 2024. 1, 2, 3, 7, 6
- [3] Marcel C Bühler, Kripasindhu Sarkar, Tanmay Shah, Gengyan Li, Daoye Wang, Leonhard Helminger, Sergio Orts-Escalano, Dmitry Lagun, Otmar Hilliges, Thabo Beeler, et al. Preface: A data-driven volumetric prior for few-shot ultra high-resolution face synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3402–3413, 2023. 1, 2, 3
- [4] Yufan Chen, Lizhen Wang, Qijing Li, Hongjiang Xiao, Shengping Zhang, Hongxun Yao, and Yebin Liu. Monogaussianavatar: Monocular gaussian point-based head avatar. *arXiv*, 2023. 1, 2
- [5] Cecilia Ding, Zheng ans Zhang, Zhihao Xia, Lars Jebe, Zhiwen Tu, and Xiuming Zhang. Diffusionrig: Learning personalized priors for facial appearance editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 3, 6, 7, 8
- [6] Guy Gafni, Justus Thies, Michael Zollhöfer, 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 (CVPR)*, pages 8649–8658, 2021. 1
- [7] Simon Giebenhain, Tobias Kirschstein, Markos Georgopoulos, Martin Rünz, Lourdes Agapito, and Matthias Nießner. Learning neural parametric head models. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 4
- [8] Simon Giebenhain, Tobias Kirschstein, Martin Rünz, Lourdes Agapito, and Matthias Nießner. Npga: Neural parametric gaussian avatars. In *SIGGRAPH Asia 2024 Conference Papers (SA Conference Papers '24), December 3-6, Tokyo, Japan*, 2024. 1, 2
- [9] Yudong Guo, Keyu Chen, Sen Liang, Yongjin Liu, Hujun Bao, and Juyong Zhang. Ad-nerf: Audio driven neural radiance fields for talking head synthesis. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1, 2
- [10] Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28, 1998. 1
- [11] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 6629–6640, Red Hook, NY, USA, 2017. Curran Associates Inc. 7
- [12] Charlie Hewitt, Fatemeh Saleh, Sadegh Aliakbarian, Lohit Petikam, Shideh Rezaifar, Louis Florentin, Zafirah Hoseinne, Thomas J Cashman, Julien Valentin, Darren Cosker, and Tadas Baltrušaitis. Look ma, no markers: holistic performance capture without the hassle. *ACM Transactions on Graphics (TOG)*, 36(6), 2024. 2, 5
- [13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arxiv:2006.11239*, 2020. 3
- [14] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*, 2020. 3
- [15] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023. 1, 2, 3, 4, 5
- [16] Taras Khakhulin, Vanessa Sklyarova, Victor Lempitsky, and Egor Zakharov. Realistic one-shot mesh-based head avatars. In *European Conference of Computer vision (ECCV)*, 2022. 3, 7, 8, 4
- [17] Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Nießner, Patrick Pérez, Christian Richardt, Michael Zollöfer, and Christian Theobalt. Deep video portraits. *ACM Transactions on Graphics (TOG)*, 37(4):163, 2018. 2
- [18] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 5
- [19] Tobias Kirschstein, Shenhan Qian, Simon Giebenhain, Tim Walter, and Matthias Nießner. Nersemble: Multi-view radiance field reconstruction of human heads. *ACM Trans. Graph.*, 42(4), 2023. 1, 2, 5
- [20] Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6):194:1–194:17, 2017. 2
- [21] Shanchuan Lin, Linjie Yang, Imran Saleemi, and Soumyadip Sengupta. Robust high-resolution video matting with temporal guidance, 2021. 5
- [22] Minghua Liu, Ruoxi Shi, Linghao Chen, Zhuoyang Zhang, Chao Xu, Xinyue Wei, Hansheng Chen, Chong Zeng, Jiayuan Gu, and Hao Su. One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion. *arXiv preprint arXiv:2311.07885*, 2023. 8
- [23] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9264–9275, 2023. 8
- [24] S. Ma, T. Simon, J. Saragih, D. Wang, Y. Li, F. La Torre, and Y. Sheikh. Pixel codec avatars. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 64–73, Los Alamitos, CA, USA, 2021. IEEE Computer Society. 2
- [25] 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. In *The European Conference on Computer Vision (ECCV)*, 2020. 1
- [26] 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, 2020. 2
- [27] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deep sdf: Learning continuous signed distance functions for shape representation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 4
- [28] Shenhan Qian, Tobias Kirschstein, Liam Schoneveld, Davide Davoli, Simon Giebenhain, and Matthias Nießner. Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians. *arXiv preprint arXiv:2312.02069*, 2023. 1, 2, 3, 6, 7, 8
- [29] Pramod Rao, Mallikarjun B R, Gereon Fox, Tim Weyrich, Bernd Bickel, Hanspeter Pfister, Wojciech Matusik, Ayush Tewari, Christian Theobalt, and Mohamed Elgharib. Vorf: Volumetric relightable faces. In *33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21–24, 2022*. BMVA Press, 2022. 4
- [30] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 8
- [31] Jack Saunders and Vinay Namboodiri. Dubbing for everyone: Data-efficient visual dubbing using neural rendering priors. *arxiv*, 2024. 2, 3
- [32] Jack Saunders and Vinay P. Namboodiri. Read avatars: Realistic emotion-controllable audio driven avatars. In *arxiv*, 2023. 2
- [33] 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 (CVPR)*, 2024. 1, 3
- [34] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt. Stylerig: Rigging stylegan for 3d control over portrait images. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6141–6150, 2020. 3
- [35] Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: real-time face capture and reenactment of rgb videos. *Commun. ACM*, 62(1):96–104, 2018. 2
- [36] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: image synthesis using neural textures. *ACM Trans. Graph.*, 38(4), 2019. 2
- [37] Justus Thies, Mohamed Elgharib, Ayush Tewari, Christian Theobalt, and Matthias Nießner. Neural voice puppetry: Audio-driven facial reenactment. *ECCV 2020*, 2020. 2
- [38] Erroll Wood, Tadas Baltrušaitis, Charlie Hewitt, Sebastian Dziadzio, Thomas J Cashman, and Jamie Shotton. Fake it till you make it: face analysis in the wild using synthetic data alone. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3681–3691, 2021. 2, 5
- [39] Cheng-hsin Wuu, Ningyuan Zheng, Scott Ardisson, Rohan Bali, Danielle Belko, Eric Brockmeyer, Lucas Evans, Timothy Godisart, Hyowon Ha, Xuhua Huang, Alexander Hypes, Taylor Koska, Steven Krenn, Stephen Lombardi, Xiaomin Luo, Kevyn McPhail, Laura Millerschoen, Michal Perdoch, Mark Pitts, Alexander Richard, Jason Saragih, Junko Saragih, Takaaki Shiratori, Tomas Simon, Matt Stewart, Autumn Trimble, Xinshuo Weng, David Whitewolf, Chenglei Wu, Shou-I Yu, and Yaser Sheikh. Multiface: A dataset for neural face rendering. In *arXiv*, 2022. 2
- [40] Jun Xiang, Xuan Gao, Yudong Guo, and Juyong Zhang. Flashavatar: High-fidelity head avatar with efficient gaussian embedding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1, 2, 4, 6, 7
- [41] 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 (CVPR)*, 2024. 1, 2
- [42] Yuelang Xu, Lizhen Wang, Zerong Zheng, Zhaoqi Su, and Yebin Liu. 3d gaussian parametric head model. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. 2, 4
- [43] Zhenhui Ye, Ziyue Jiang, Yi Ren, Jinglin Liu, Jinzheng He, and Zhou Zhao. Geneface: Generalized and high-fidelity audio-driven 3d talking face synthesis. *arXiv preprint arXiv:2301.13430*, 2023. 2
- [44] Changqian Yu, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen, and Nong Sang. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *Int. J. Comput. Vision*, 129(11):3051–3068, 2021. 5
- [45] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 4, 7
- [46] 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 (CVPR)*, 2023. 2
- [47] Wojciech Zielenka, Timo Bolkart, and Justus Thies. Instant volumetric head avatars. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4574–4584, 2022. 1, 2

GASP: Gaussian Avatars with Synthetic Priors

Supplementary Material

A. Further Results

We show further examples of self-reenactment, wherein we take an unseen video of the subject and use it to drive their avatar. We show full 360° renderings of the head. The results are shown in Fig. 11. Despite having never seen the back of an actual person’s head, our model produces plausible results. We also show cross-identity re-enactment, taking a video from one Avatar to animate several others. This is demonstrated in Fig. 10. Video versions of these results are also shown in our supplemental video.

B. Latent space controllability

To demonstrate that our prior model learns a controllable latent space, we propose a simple method for finding directions \mathbf{d}_k in the latent space that are semantically meaningful. We then demonstrate that adding or subtracting those direction from a given identity’s latent vector \mathbf{z}_j leads to the desired changes in the person’s appearance. The results of this process are shown in Figure 12.

To learn \mathbf{d}_k for a given semantic feature, we group our training data into samples that have this feature and samples that do not have it. As our training data is synthetic and extensively labeled, doing so is a matter of checking the metadata of the samples. We then take a pre-trained prior model and extract the \mathbf{z}_j for each training sample. Finally, we train a Linear Support Vector Machine [10] that classifies the training data samples into ones that have the semantic feature and ones that do not have it, given the sample’s \mathbf{z}_j . The direction \mathbf{d}_k estimated by the Linear SVM is a vector orthogonal to the hyperplane that separate the two groups in the latent space of the prior model. Thus, adding \mathbf{d}_k to a sample’s latent vector \mathbf{z}_j should move it closer to samples that have the feature, and subtracting it should have the opposite effect.

We evaluate this approach on three features:

1. Age - this corresponds to the age of the person whose facial texture was used in the training data sample. The SVM here was learned to classify age ≥ 45 into a separate group from age < 45 .
2. Facial hair - here, the SVM was learned to classify samples with facial hair separately from samples with no facial hair.
3. Head hair - here, the SVM separated samples with long hair from samples with short hair.

The results of the evaluation are shown in the supplementary video as well as in Figure 12, where each column demonstrates one of the features we control.

Subject	Test Cameras	Subject	Test Cameras
36	221501007	37	221501007
	222200040		222200040
	222200044		222200044
	222200046		222200045
57	221501007	74	221501007
	222200040		222200040
	222200044		222200042
	222200046		222200044
100	221501007	145	221501007
	222200039		222200042
	222200042		222200044
	222200045		222200045
165	221501007	251	221501007
	222200042		222200042
	222200044		222200044
	222200045		222200045

Table 4. Cameras selected as the most extreme view for each subject. The selection was performed empirically.

C. MLP Architecture

Here, we give more detail about the architecture of our MLP Decoder \mathcal{D} . The network takes each 8-dimensional Gaussian feature \mathbf{f}_i as input and concatenates them with the 512-dimensional vector \mathbf{z}_j for the identity of the Avatar. This gives a 512-dimensional vector. These inputs are then passed through six linear layers with an output dimensionality 256. After this, the network separates into separate branches for position μ , scale σ , rotation r , color c and opacity o . Each branch has one linear layer with output dimension 256, followed by a final linear projection to the relevant shape for that attribute. Each linear layer, except the final projection, is followed by the ReLU activation function. Weight normalization is used on each layer. We visualise this architecture in Fig. 13

D. Experimental Setup

Here, we discuss the exact setup of the experiments in the main paper. Recall we consider three experimental setups: Monocular, Single Frame and Multi Camera.

Monocular For each training subject, we used the following sequences as training data: EMO-1-shout+laugh, EMO-2-surprise+fear, EMO-3-angry+sad, EMO-4-disgust+happy, EXP-2-eyes, EXP-3-cheeks+nose, EXP-



Figure 10. **Additional Cross Reenactment Results.** We show several more examples of cross-reenactment. We use the input image on the left to drive the avatars on the right. Each Avatar is trained in the **Monocular Setting**.

4-lips, EXP-5-mouth, EXP-6-tongue-1, EXP-7-tongue-2, EXP-8-jaw-1, EXP-9-jaw-2. For all subjects except 57, the camera 222200037 was selected as the most frontal, for subject 57 this was 222200038. These are the cameras we used in training. We subsample every other frame.

Single Image For each training subject, we used the first frame of the sequence EMO-1-shout+laugh for training data. For all subjects except 57, the camera 222200037 was selected as the most frontal, for subject 57 this was 222200038. These are the cameras we used in training.

Multi Camera For each training subject, we used the following sequences as training data: EMO-1-shout+laugh, EMO-2-surprise+fear, EMO-3-angry+sad, EMO-4-disgust+happy, EXP-2-eyes, EXP-3-cheeks+nose, EXP-4-lips, EXP-5-mouth, EXP-6-tongue-1, EXP-7-tongue-2, EXP-8-jaw-1, EXP-9-jaw-2. We used all 16 Cameras. In order to reduce the size of these datasets, we subsampled every 10th frame, effectively taking each video at 7fps.

Testing Testing on all subjects was performed using the FREE sequence, which has no overlap with any of our train-

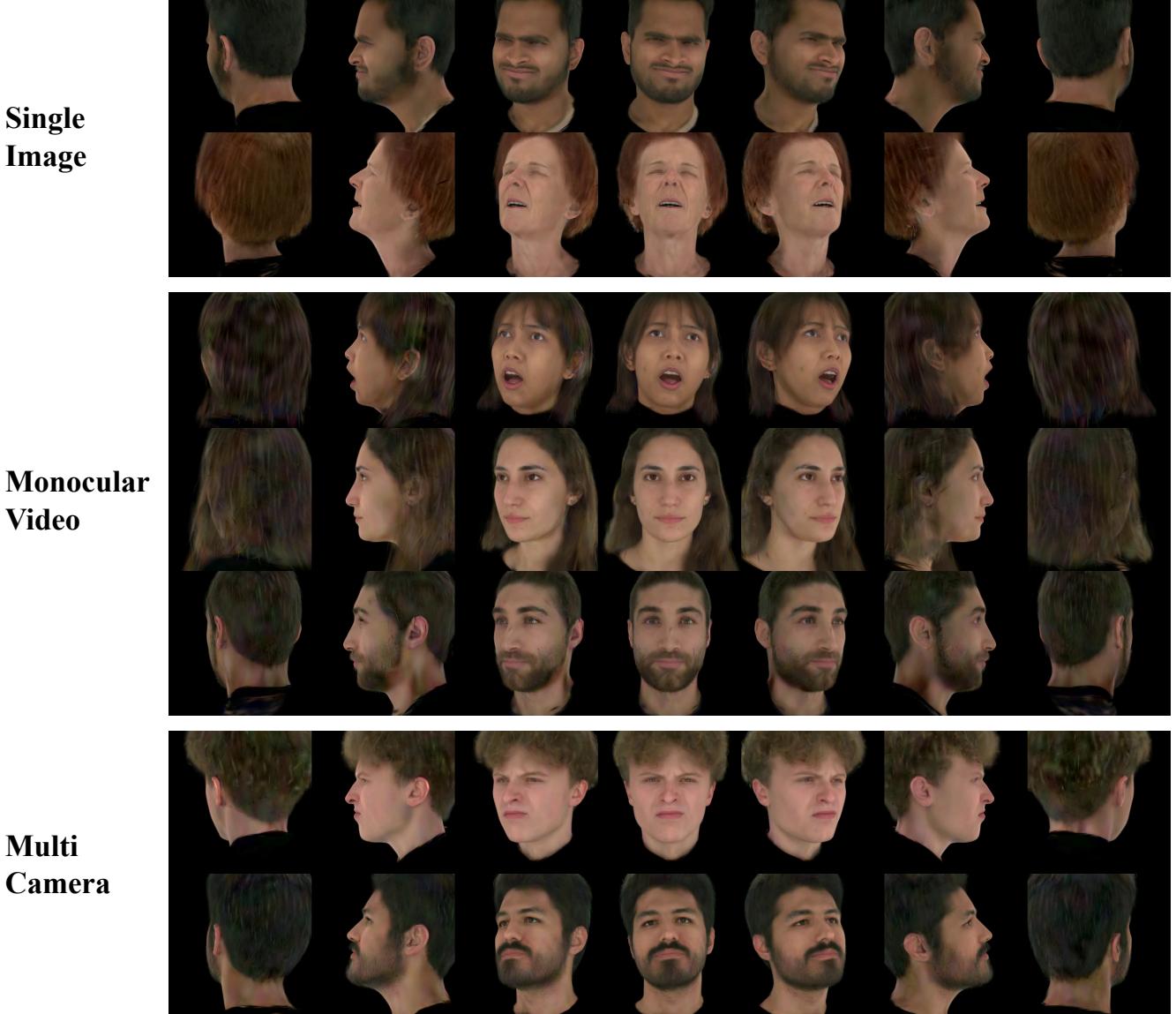


Figure 11. Additional Self Reenactment Results. We show several more examples of self-reenactment with 360° rendering. We show models fit to a single image (Top), a monocular video (Middle) and multiple views (Bottom). In each case, the back of the head is never included in the fitting data.

ing sets. We used cameras as shown in Tab. 4. For the main quantitative results, we subsample every 5th frame to reduce computational overhead. For generating the qualitative videos we use every frame of the FREE sequence.

E. Three Frame Model

Some other few-shot Avatar models (e.g., [2, 3]) address a related but different experimental setup using three frames, one frontal facing, one from the left and one from the right. While these models are not available for comparison, we replicate their setup here. For this, we select one image

from the front, left and right of a model. We show some of the results in Fig. 15. It can be seen here that our model performs somewhat better on novel expressions from one of the training views (the left and right columns) and has significantly fewer artefacts on a novel view (middle column).

F. User Study

For our user study, we ask participants to rate the quality of each method. We show each method the FREE sequence played from the four extreme test cameras in Tab. 4. Each participant is shown each combination of method and train-

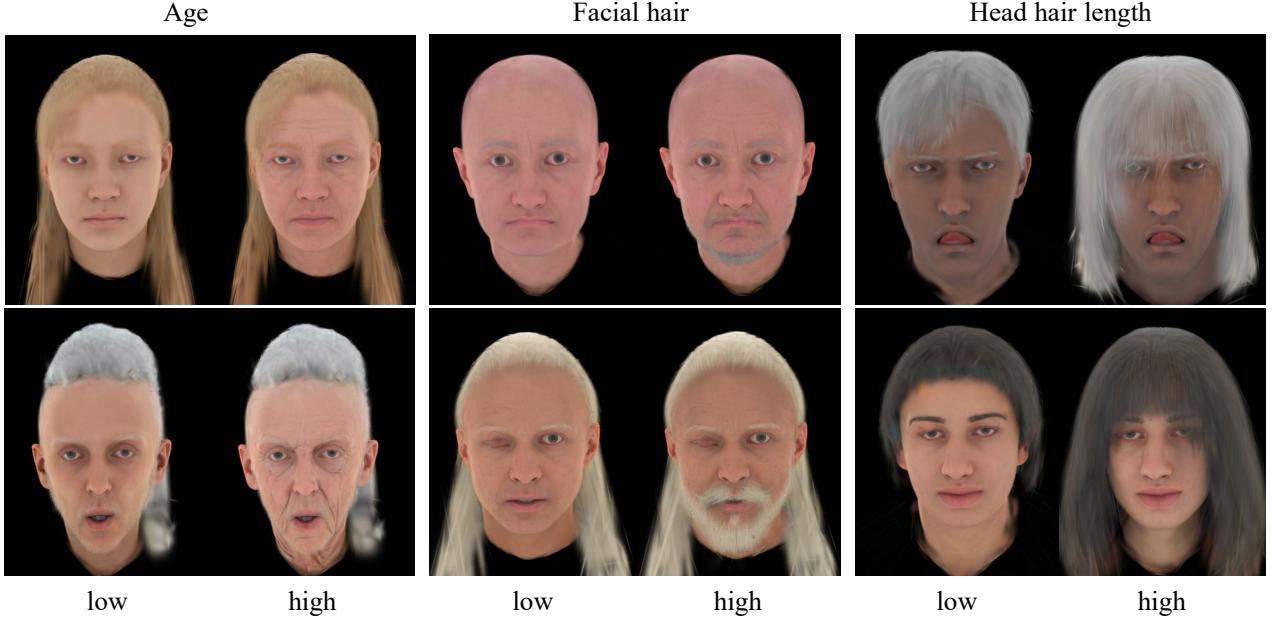


Figure 12. We demonstrate that the latent space learned by our prior model is controllable by finding directions in it that correspond to semantic features such as age, facial hair and hair length.

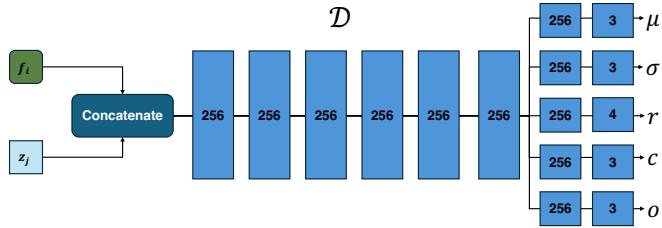


Figure 13. The architecture of our MLP decoder \mathcal{D} . f and \ddagger are concatenated and passed through 6 linear layers with output size 256. The network then splits into per-attribute branches. Each block represents a linear layer followed by ReLU and using weight normalization.

ing setting (Monocular, Single Frame and Multi-Camera) for an individual subject, meaning a total of 13 images per user (Four methods times 3 settings plus the Single Frame setting for ROME [16]). Images are shown in a grid of two-by-two using each of the four camera angles. We perform this for each of the eight test subjects we have run evaluation on, with users being assigned one of these subjects at random. Video order is also randomized to prevent bias. We conducted the user study using Amazon’s Mechanical Turk. In total, 40 users completed the user study. The results are shown in Tab. 1 and Tab. 2.

G. Ablations

We use subjects A, B, and C for our ablation study. We consider the monocular setup described in Appendix D. In addition to the qualitative results displayed in Tab. 3, we also show the results of our ablation study qualitatively. Figure 14 shows the effect of training our prior on differing numbers of subjects, ranging from using no prior, to using the complete 1k subjects. In each case, we select all frames from the first N training subjects in the synthetic training dataset for a prior with N subjects. Figure 14 also shows the effect of using a different number of Gaussian primitives in the model. Here, we use varying UV map resolutions for the initialization (see Sec. 3.4); we consider maps of resolution 64x64 (2926 Gaussians), 128x128 (11,758 Gaussians), 256x256 (46,928 Gaussians) and our full model using 512x512 (187,776 Gaussians).

Canonical Gaussians: In addition to the ablations shown in the main paper, we also validate our claim that canonical Gaussians improve training stability. To show this, we plot the image space loss curves for $\lambda_{pix}L_{pix} + \lambda_{percep}L_{percep}$ in Fig. 16.

H. Ethical Concerns

We recognise the potential for misuse of our model. We feel strongly about preventing this. We are actively researching watermarking methods for avatars and metadata labelling methods, such as the C2PA Initiative. We are also considering systems for likeness management, for example, only



Figure 14. **Ablations:** We show the qualitative effect of using differing numbers of subjects to train the prior (top) and different numbers of Gaussians (bottom).

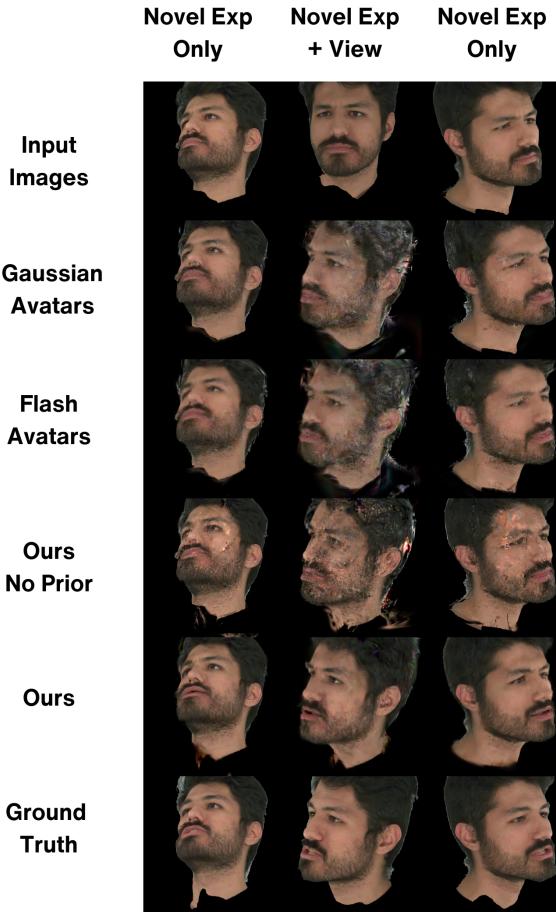


Figure 15. Qualitative comparisons of our method with existing state-of-the-art in the **Three Image Setting**, using the top 3 images as input. We show both novel expression and novel view synthesis in this setup.

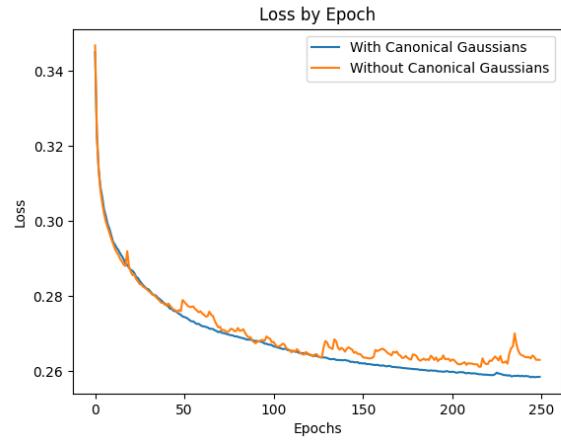


Figure 16. The training loss curves for $\lambda_{pix}L_{pix} + \lambda_{percep}L_{percep}$ with (blue) and without (orange) the canonical Gaussians. Note the improved training stability and better overall loss.

allowing a single account to operate an avatar. Before deploying any avatar system using our method, we will consult a wide range of stakeholders to mitigate the possibility of harm through our model.

Our model has advantages over others that have built priors over non-synthetic data. If we expose our prior to a user training their avatar, we do not run the risk of dataset distillation attacks. This means that there is no risk of privacy violations wherein an adversary could obtain personal data about subjects that have been used to train the prior. This also helps avoid legal issues around GDPR and consent. There is no chance of a subject withdrawing consent and requiring our prior to be retrained or detained.



Figure 17. A comparison of our method (Right) compared to Cafca [2] (Middle), using the input image on the left. Our model performs better on the side of the head, such as on the ear, while being thousands of times faster to render. Our model can also be animated, while Cafca cannot.

I. Comparison to Cafca

Cafca [2] is a NeRF-based synthetic prior model that shares several similarities with our work. However, there is some crucial differences. Their method is only capable of modelling static expressions and cannot be animated. Furthermore, rendering for Cafca takes 20 seconds per frame on a 4 TPU machine. Our model, conversely can be freely animated and rendered at 70fps on a much more available NVIDIA 4090 RTX GPU. Despite our models much faster rendering time, we are able to achieve a similar level of quality, with our model better capturing the ear and back of head detail, but not quite getting as much high-frequency details. A comparison can be seen in Fig. 17. As Cafca is not publically available, we take their results directly from their project page.