

GASP: Gaussian Avatars with Synthetic Priors

Supplementary Material

A. Comparison with State-of-the-Art

In Tab. 4, we compare our method against many state-of-the-art methods across a range of important categories. We show that our model is the only one that is real-time, animatable, fast to fit and can model the back of the head using just a single camera for data.

B. Implementation Details

Identity codes, \mathbf{z} , and Gaussian features, \mathbf{f} , are 512 and 8 dimensional, respectively. We initialize with a UV map of 512×512 pixels, resulting in 187,779 Gaussians. The supplementary material details our decoder network \mathcal{D} 's architecture. For all parameters, we optimize using the Adam optimizer [24]. The canonical Gaussians are optimized using the learning rates from the original implementation of 3D Gaussian Splatting [21]. We optimize \mathbf{z} and \mathcal{D} with a learning rate 0.0002. We set the values of the loss function weights as follows: $\lambda_1 = 0.8$, $\lambda_{SSIM} = 0.1$, $\lambda_\alpha = 0.2$, $\lambda_{percept} = 0.5$, $\lambda_\sigma = 0.01$ and $\lambda_\mu = 0.01$ for the face and 0.0001 for the scalp. Prior network training took 4 days and was performed using 4xA100'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.

C. 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. 9 and Fig. 11. Despite having never seen the back of an actual person's head, our model produces plausible results. We also show cross-identity reenactment, taking a video from one Avatar to animate several others. This is demonstrated in Fig. 9 and Fig. 10. Video versions of these results are also shown in our supplemental video.

D. 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 Fig. 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 [15] 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 Fig. 12, where each column demonstrates one of the features we control.

We also show the effects of each stage of the fitting process in Fig. 13, as discussed in Sec. 5.5.

E. 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 visualize this architecture in Fig. 14



Figure 9. **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.

Method	Real-time	Animatable	Single Camera	$\pm 90^\circ$ Rendering	Back of the Head	Fast Fitting	Models Hair
Athar et al. [1]	✓	✓	✓	✓	✗	✓	✗
Cao et al. [6]	-	✓	✓	✓	✗	✗	✗
Mihajlovic et al. [31]	✗	✓	✗	✓	✗	✗	✓
Grassal et al. [13]	-	✓	✓	✗	✗	✗	✓
Zheng et al. [55]	✗	✓	✓	✗	✗	✗	✓
Bharadwaj et al. [2]	✗	✓	✓	✓	✗	✓	✓
Zielonka et al. [57]	-	✓	✓	✗	✗	✓	✓
Hong et al. [19]	✓	✓	✓	✗	✗	✓	✓
Xiang et al. [47]	✓	✓	✓	✗	✗	✓	✓
Zheng et al. [54]	✗	✓	✓	✓	✗	✗	✓
Xu et al. [50]	✓	✓	✓	✓	✗	✓	✓
Buehler et al. [4]	✓	✓	✓	✓	✗	✓	✓
Ours	✓	✓	✓	✓	✓	✓	✓

Table 4. A Table detailing the comparisons between our work and related state of the art works. Our model is the only one that is real-time, animatable, fast to fit and can model the back of the head using just a single camera for data. A ✓ means that a model meets a given criteria, a ✗ that it doesn’t and a – that it is not stated. We define real-time as over 25fps and fast fitting as under an hour.

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

F. 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-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



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**.

222200038. These are the cameras we used in training.

Multi Camera. For each training subject, we used the all 16 Cameras from 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. In order to reduce the size of these datasets, we sub-sampled 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 training sets. We used cameras as shown in Tab. 5. 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.

G. Three Frame Model

Some other few-shot Avatar models (e.g., [4, 5]) 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. 16. It can be seen here that our model

**Single
Image**



**Monocular
Video**



**Multi
Camera**



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.

performs somewhat better on novel expressions from one of the training views (the left and right columns) and has significantly fewer artifacts on a novel view (middle column).

H. 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. 5. Each participant is shown each combination of method and training setting (Monocular, Single Frame and Multi-Camera) for an individual subject, meaning a total of 13 images

per user (Four methods times three settings plus the Single Frame setting for ROME [22]). Images are shown in a grid of two-by-two using each of the four camera angles. We do 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 with 40 participants using Amazon’s Mechanical Turk. The results are shown in Tab. 1 and Tab. 2.

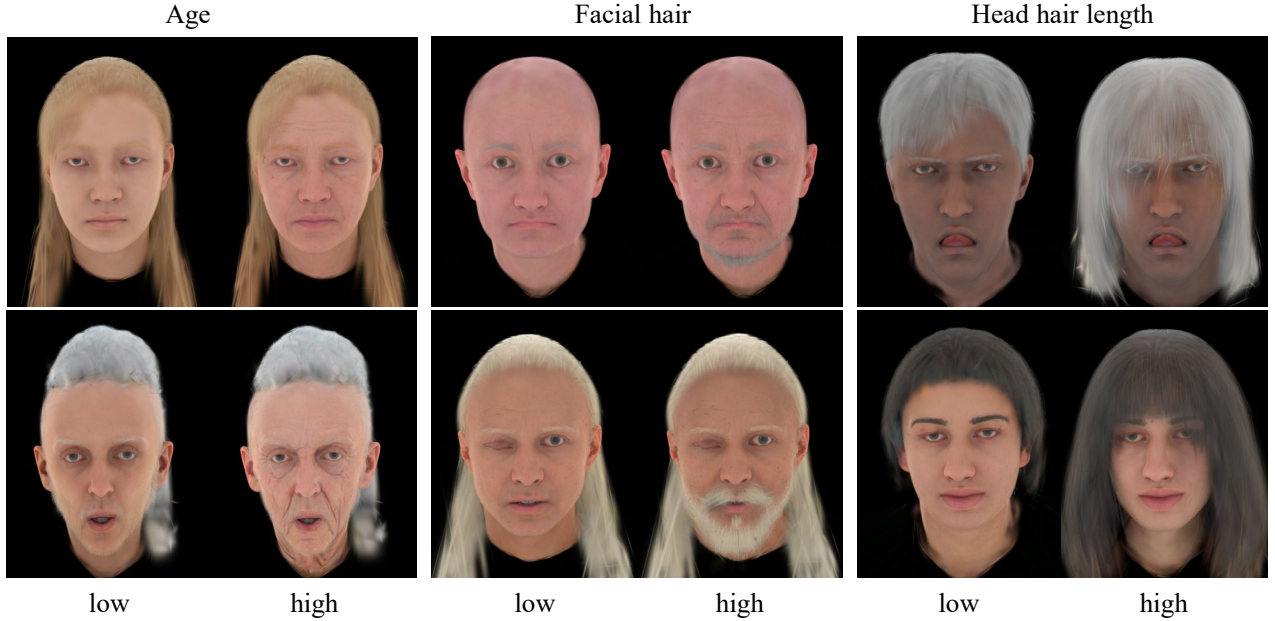


Figure 12. We demonstrate that the latent space learned by our prior model is controllable by finding directions in it that correspond to semantic features such as age, facial hair and hair length.



Figure 13. 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.

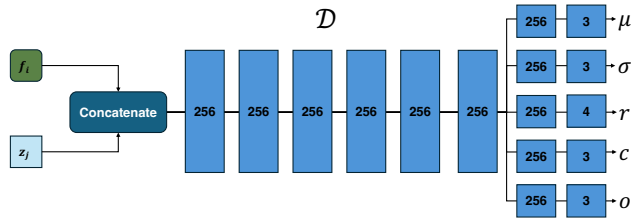


Figure 14. The architecture of our MLP decoder \mathcal{D} . \mathbf{f} and \mathbf{z} are concatenated and passed through 6 linear layers with output size 256. The network then splits into per-attribute branches. Each block represents a linear layer followed by ReLU and using weight normalization.

I. Ablations

We use subjects A, B, and C for our ablation study. We consider the monocular setup described in Appendix F. In addition to the qualitative results displayed in Tab. 3, we also show the results of our ablation study qualitatively. Figure 15 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 15 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 64×64 (2926 Gaussians), 128×128 (11, 758 Gaussians),



Figure 15. **Ablations:** We show the qualitative effect of using differing numbers of subjects to train the prior (top) and different numbers of Gaussians (bottom).



Figure 16. 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.

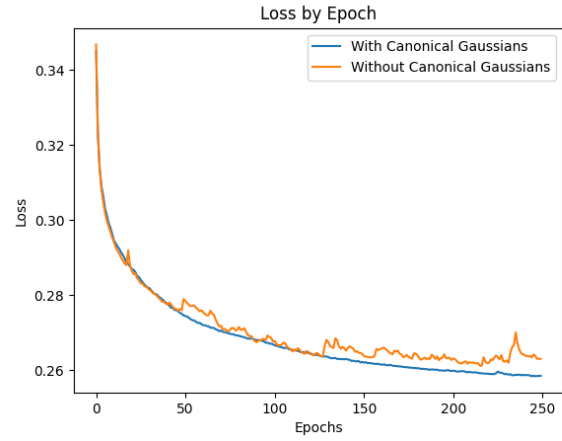


Figure 17. The training loss curves for $\lambda_{pix}L_{pix} + \lambda_{percep}L_{percep}$ with (blue) and without (orange) the canonical Gaussians. Note the improved training stability and better overall loss.

256 × 256 (46,928 Gaussians) and our full model using 512 × 512 (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. 17.

J. Ethical Concerns

We recognize the potential for misuse of our model. We feel strongly about preventing this. We are actively researching watermarking methods for avatars and metadata labeling



Figure 18. A comparison of our method (Right) compared to Cafca [4] (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.

methods, such as the C2PA Initiative. We are also considering systems for likeness management, for example, only 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.

K. Comparison to Cafca

Cafca [4] 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 modeling 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 detail. A comparison can be seen in Fig. 18. As Cafca is not publicly available, we take their results directly from their project page.

L. Acknowledgement

We want to thank the rest of the team at Microsoft for their productive discussions on this project. The UKRI has supported Jack Saunders under grant EP/S023437/1 during his PhD studies.