HeadGaS: Real-Time Animatable Head Avatars via 3D Gaussian Splatting

Supplementary Material

This supplementary material provides further details about HeadGaS, as well as additional results. Section 1 provides some insights on the learned 3D Gaussians, such as visualization of the learned feature basis as well as a gradual removal of the Gaussians to observe the occluded content. Section 2 provides more implementation details on the methods used in the ablation of the main paper. Further, we provide some time analysis in Section 3. In Section 4 we show more qualitative results on the novel expression task comparing ours against baselines. Finally, in this supplement we kindly refer the reader to a *demo video* which contains method highlights and various result sequences as comparison with state-of-the-art methods, ablation and novel view synthesis.

1 Understanding the Learned 3D Gaussians

Basis visualization Since HeadGaS relies on a feature basis for blending, it would be beneficial to understand what this basis is learning. For this purpose, we utilize expression parameters as one-hot vectors to our model and show the generated image alongside the 3DMM mesh corresponding to that expression in Figure 1. The left section of the figure shows two examples on the NBS data [3], using Face Warehouse [2], while the right part provides two FLAME [4] based examples from the INSTA data [6]. Note that for the FLAME-based tracking we noticed that the optimized neutral expressions were far from a vector of zeros, therefore we normalized the one-hot expression vectors first, by adding the expression weights of a neutral expression from the training set. Since the Face Warehouse basis is more semantic, *i.e.* every expression element corresponds to a more local and interpretable action, such as winking, or opening mouth in surprise, we observe more drastic changes in this base compared to FLAME. The results show that HeadGaS learns a reasonable feature basis that aligns well with the 3DMM expressions. However, this is limited by the level in which an expression is observed in the training data, *e.g.* for subject 2 from the top we observe that winking does not work quite well (both eyes are closing instead of one) due to this reason.

Gradual removal of Gaussians The purpose of this experiment is to reveal what the intermediate Gaussians in a particular frame represent. This is of interest because the proposed method relies on over-representation, *i.e.* multiple Gaussians will represent certain face areas (*e.g.* lips) and they will occasionally become transparent to reveal other areas underneath (*e.g.* teeth) as needed. Therefore we remove the optimized 3D Gaussians gradually, using the camera

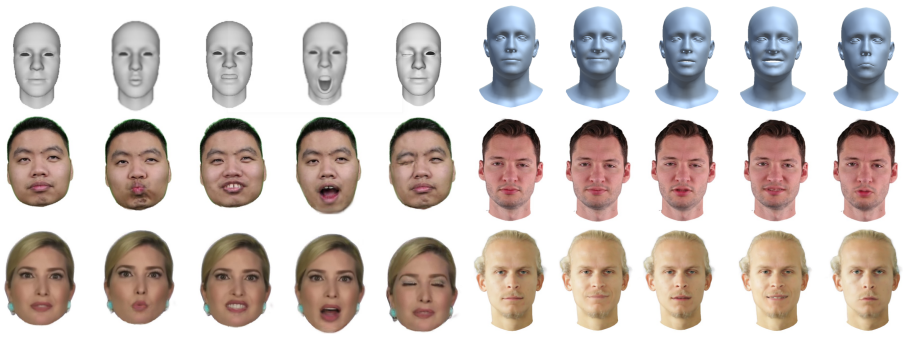

Fig. 1: Learned feature basis visualization. *Left:* Expression parameters from the Face Warehouse model *Right:* Expression parameters from the FLAME model.

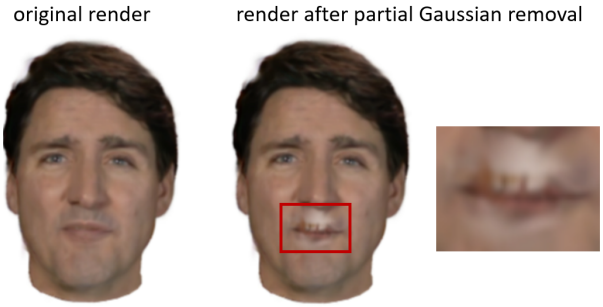

Fig. 2: Removal of frontal 3D Gaussians reveals underlying structures such as teeth, invisible in this frame, but observed in other frames.

view as direction, from near to far. Figure 2 shows the original rendering for a given frame as well as the rendering after we have removed the most frontal Gaussians in the mouth area. As expected, we start seeing teeth in the intermediate layers of visibility, as these structures have been observed from other frames (*e.g.* when the person was talking or laughing). This reflects that, our model does not simply re-color the Gaussians in those areas to accommodate teeth instead of lips, but rather has a separate set of Gaussians to represent the teeth. Additionally, we observe background color (white) when dis-occluding the regions between the nose and the upper lip. This is due to the fact that there is no need to represent the structures underneath, as they are never observed by any frame.

Effect of going outside of the training manifold In Figure 3 (top) we navigate along one 3DMM parameter up to 1.4× of its maximum value in the training set. We notice that colors start slowly to deteriorate after 1.2×max. This observations is in line with the expectations, as we model deformation through changes in color and opacity. Similarly, in Figure 3 (bottom) we see that if the

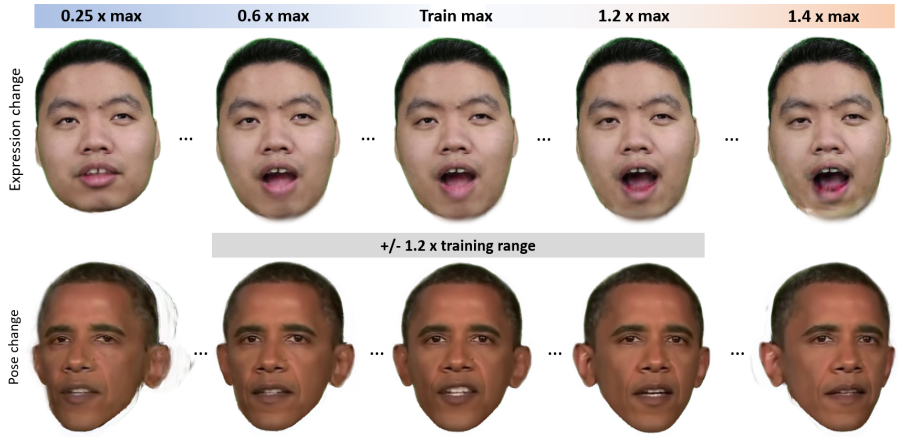

Fig. 3: (Top) Navigation of 3DMM expression space for 1 parameter. (Bottom) Navigation of viewing angle.

viewing angle changes considerably from the range of observed training views, we start noticing some artifacts.

2 Ablation Details

This section provides more details on the ablation methods presented and evaluated in the main paper.

Ours w/o MLP This model does not use learned features for the blending, but rather it blends a basis of colors and opacities directly. In each Gaussian we create these bases as $\boldsymbol{C} \in \mathbb{R}^{B\times 3(k+1)^2}$ and $\boldsymbol{\alpha} \in \mathbb{R}^{B\times 1}$ and use the expression weights to obtain the final color $\boldsymbol{c}_i$ and opacity α_i as a weighted average. Note that summation did not work here, as colors are explicit values and they would add up to high values.

Ours w/ $\boldsymbol{\Delta(\mu, R)}$ This model uses the learned feature basis $\boldsymbol{F}$ of HeadGaS to rather shift the positions $\boldsymbol{\mu}$ and transform the rotations $\boldsymbol{R}$. Thereby we feed the blended feature $\boldsymbol{f}_i$ in to an MLP $\phi'(\cdot)$ that contains the same number of layers and hidden dimensions as $\phi(\cdot)$ from our proposed model. The difference is that, $\phi'(\cdot)$ outputs 3 values of position shift and 4 values of rotation (represented as quaternion) as:

$$\Delta\boldsymbol{\mu}, \boldsymbol{r}_t = \phi'(\boldsymbol{f}_i, \psi(\boldsymbol{\mu})). \tag{1}$$

These outputs are namely used to transform the static parameters of the Gaussian (after converting $\boldsymbol{r}_t$ to a rotation matrix $\boldsymbol{R}_t$) as

$$\boldsymbol{\mu}' = \boldsymbol{\mu} + \Delta\boldsymbol{\mu} \tag{2}$$

and

$$\boldsymbol{R}' = \boldsymbol{R}_t \boldsymbol{R}. \tag{3}$$

To facilitate convergence, we additionally add a regularization term to the estimated position shift, encouraging it to remain in a small range. The final loss then becomes

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_1(I_{\text{r}}, I_{\text{gt}}) + \lambda_s \mathcal{L}_{\text{SSIM}}(I_{\text{r}}, I_{\text{gt}}) + \lambda_p \mathcal{L}_p(I_{\text{r}}, I_{\text{gt}}) + \lambda_\mu \mathcal{L}_1(\Delta\boldsymbol{\mu}). \tag{4}$$

Ours change all This model uses the learned feature basis $\boldsymbol{F}$ of HeadGaS to predict color, opacity as well as a shift of positions $\boldsymbol{\mu}$ and transformation of rotations $\boldsymbol{R}$. Thereby we feed the blended feature $\boldsymbol{f}_i$ to an MLP $\phi''(\cdot)$, which outputs sh colors, opacity, 3 values of position shift and 4 values of rotation (represented as quaternion). The transformations are applied in the same way as in *Ours w/* $\Delta(\mu, R)$. Also here we apply the regularization term in the loss function, as in eq. (4).

Ours w/o blending This model aims to verify that using the expression parameters as a weight for blending Gaussian features works better than using it as a simple condition to the MLP. Therefore, here we do not have a basis of latent features $\boldsymbol{F}$. Instead, the expression vector $\boldsymbol{e}_i$ and the encoded position $\boldsymbol{\mu}$ are fed directly into $\phi(\cdot)$ to predict the color and opacity. We hypothesize that this baseline requires more capacity for the MLP, as, in contrast to our proposed method, it has to learn all dynamics of the face at once. Therefore, we do not restrict our experiments to a small MLP of two layers, but rather extend its capacity until it reaches a plateau. Thus, the MLP here results in five linear layers, each followed by a leaky ReLU.

Ours w/o $\boldsymbol{\mathcal{L}_p}$ This model is the same as the proposed HeadGaS and simply has the perceptual loss disabled

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_1(I_{\text{r}}, I_{\text{gt}}) + \lambda_s \mathcal{L}_{\text{SSIM}}(I_{\text{r}}, I_{\text{gt}}). \tag{5}$$

3 Time analysis

Rendering time vs image size In Figure 4 we plot our models relationship between rendering time and image resolution. For each subject we render our models in 3 different resolutions, namely 512^2, 1024^2 and 2048^2 and collect the run-time statistics. The number of Gaussians range between 21k and 37k, which is one of the main factors affecting the rendering time. The resulting mean rendering time for each resolution is namely 0.004, 0.005 and 0.0086, *i.e.* the rendering time only doubles when we increase resolution by a factor of 4 in both dimensions (*i.e.* 16× more pixels).

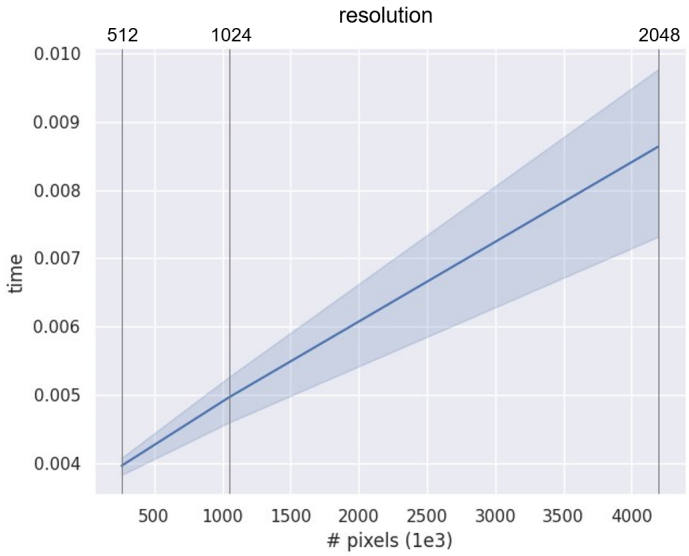

Fig. 4: Rendering time versus resolution for HeadGaS on the INSTA dataset. We show the statistics of all subjects, with number of Gaussians ranging from 21k to 37k.

Ablation on number of 3D Gaussians and rendering time In addition to the image quality metrics evaluated in the main paper, we compare our different ablation models in terms of rendering time and number of Gaussians in Table 1. We notice that the proposed HeadGaS (Ours) results in significantly less Gaussians compared to other models while being the fastest to render. Interestingly, the model that blends explicit parameters directly (Ours w/o MLP) is still slower than ours, despite not employing an MLP computation due to its large number of Gaussians. Another interesting observation is that, even though HeadGaS relies on over-representation, it still leads to significantly less Gaussians compared to the alternative model that transforms Gaussians (Ours w/ $\Delta(\mu, R)$). We believe this is due to the fact that the proposed model is more effective and easier to learn, and therefore it leads to the most efficient representation of space compared to other variants.

Table 1: Ablation methods compared in terms of number of optimized 3D Gaussians and per-frame rendering time. Our method leads to the lowest number of Gaussians (*i.e.* most efficient coverage of space) while having the best PSNR.

Method	# Gaussians ↓	Time (s) ↓	PSNR ↑
Ours w/o blending	135k	0.008	29.38
Ours change all	97k	0.012	29.65
Ours w/ $\Delta(\mu, R)$	234k	0.019	29.83
Ours w/o MLP	295k	0.010	32.08
Ours	**28k**	**0.004**	**32.50**

4 Additional qualitative results

We provide additional qualitative results comparing HeadGaS against the most recent baselines [1, 3, 5, 6] in Figure 5 and Figure 6. We observe that generally our model renders images with less artifacts, higher similarity to the ground truth expression, more noticeable reflecting glasses and skin specularities (Figure 5, row 2).

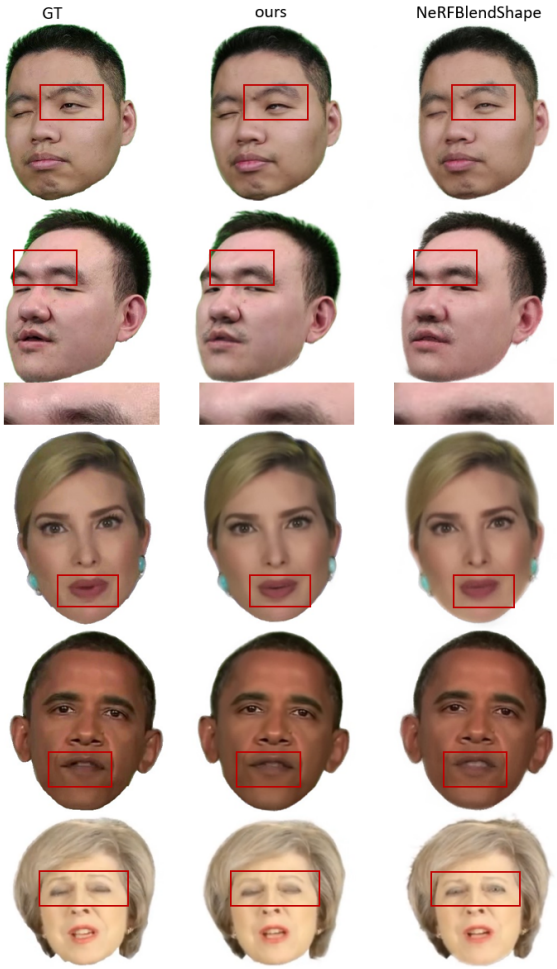

Fig. 5: Additional qualitative results on the NBS data. *Left:* Ground truth, *Center:* HeadGaS (ours), *Right:* NeRFBlendShape.

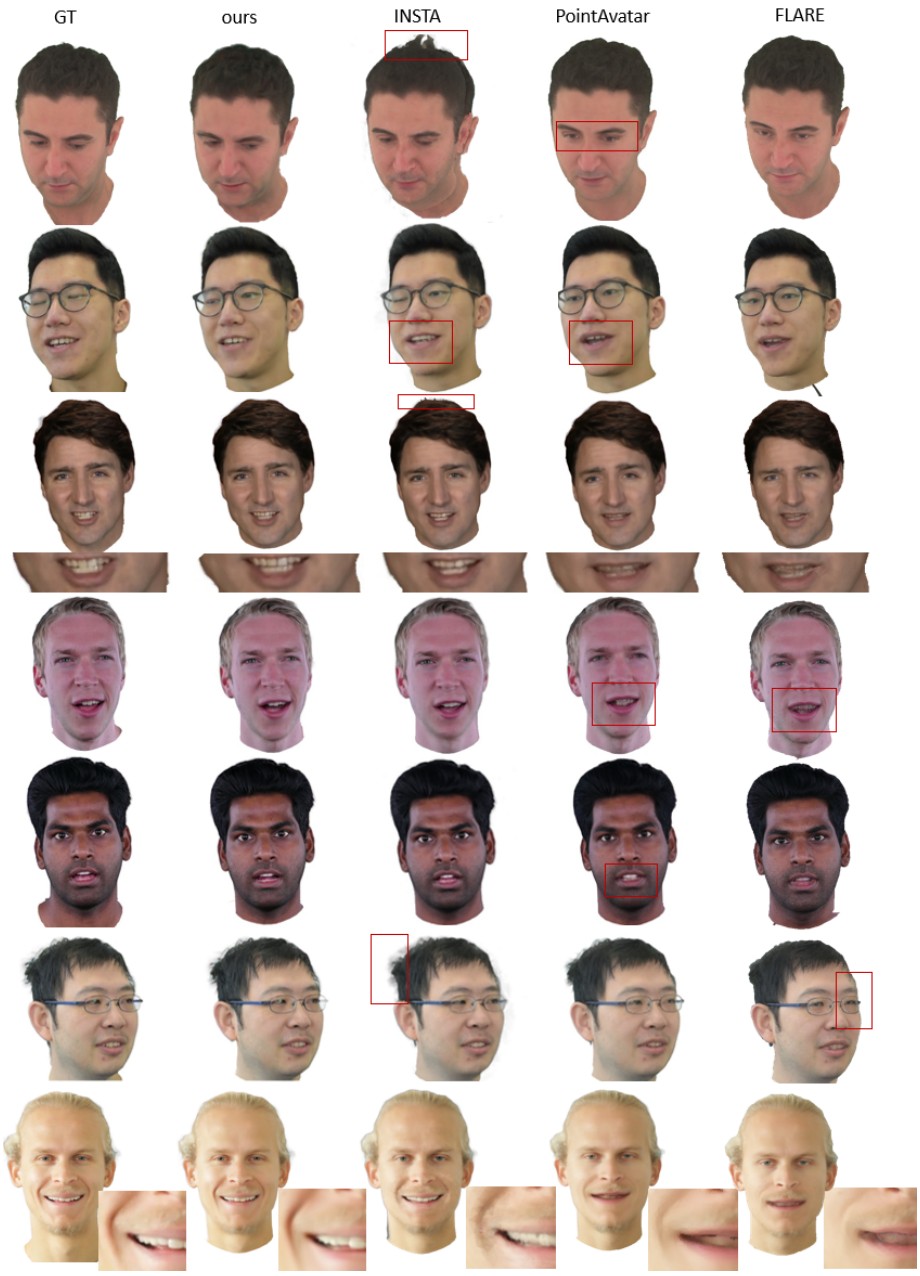

Fig. 6: Additional qualitative results on the INSTA data. From left to right: Ground truth, HeadGaS (ours), INSTA, PointAvatar and FLARE.

References

1. Bharadwaj, S., Zheng, Y., Hilliges, O., Black, M.J., Abrevaya, V.F.: FLARE: Fast learning of animatable and relightable mesh avatars. ACM TOG (2023) 6

2. Cao, C., Weng, Y., Zhou, S., Tong, Y., Zhou, K.: Facewarehouse: A 3d facial expression database for visual computing. IEEE Transactions on Visualization and Computer Graphics (2014) 1
3. Gao, X., Zhong, C., Xiang, J., Hong, Y., Guo, Y., Zhang, J.: Reconstructing personalized semantic facial nerf models from monocular video. ACM TOG (Proceedings of SIGGRAPH Asia) (2022) 1, 6
4. Li, T., Bolkart, T., Black, M.J., Li, H., Romero, J.: Learning a model of facial shape and expression from 4D scans. ACM TOG, (Proc. SIGGRAPH Asia) (2017) 1
5. Zheng, Y., Yifan, W., Wetzstein, G., Black, M.J., Hilliges, O.: Pointavatar: Deformable point-based head avatars from videos. CVPR (2023) 6
6. Zielonka, W., Bolkart, T., Thies, J.: Instant volumetric head avatars. CVPR (2023) 1, 6