High-Fidelity Mobile Avatars with Pruned Local Blendshapes

Supplementary Material

1. Attribute PCA Analysis

To demonstrate that the attributes of local Gaussians are highly correlated, we perform PCA on Gaussian attributes. We use color as the example. We collect the output RGB color attributes from AnimatableGS [3]. There are $F = 1500$ frames and the model has $N_g = 370K$ Gaussians, so all colors form a large matrix. We subtract each Gaussian's mean color to obtain the color corrective matrix X with size $[F, 3N_g]$.

For SqueezeMe [2] and TaoAvatar [1], they use a global pose feature to combine all the blendshapes. That is, they represent the full color matrix X using a blendshape matrix of size $[N_B, 3N_g]$, where N_B is the dimension of the blendshape space used in their paper. We use PCA on all the Gaussians (global PCA) to measure how large the approximation error of this representation is. We reduce the matrix X to $[N_B, 3N_g]$, using the first N_B dimensions, then project it back to original space $[F, 3N_g]$ as a reconstruction. To measure the reconstruction error, we compute the L1 error between X and its reconstruction. We test with $N_B = 16$ and 64 and report the error in Tab. 1. We note that $N_B = 16$ is the parameter used in our paper and $N_B = 64$ is the same setting used in SqueezeMe [2].

For our design, we group Gaussians by their positions according to the different parts of the body, and use local pose feature to combine the local blendshape within each group. To measure the approximation error of this representation, we group the Gaussians from AnimatableGS [3] in the same way as we do in the paper. Then we form many color matrices $\{X_k\}$, one for each group, perform PCA on each group (local PCA) to reduce the dimension, then project them back to the original space as reconstructions. We also compute the L1 error between the original color matrices and their reconstructions. The results are shown in Tab. 1. The local PCA produces lower error, indicating that grouping Gaussians by positions yields better reconstruction error.

We also report the explained variance ratio to show how concentrated the principal components are for local and global PCA. In PCA, the explained variance ratio is the fraction of the dataset's total variance that a principal component accounts for. A higher explained variance ratio means the component carries more of the original data's information. The ratio for local PCA is the average explained variance ratios computed for all matrices $\{X_k\}$. We present the results in Fig. 1. The ratios of local PCA are more concentrated in the first few components, further demonstrating that local Gaussians' attributes are more correlated and can be captured with fewer blendshapes to capture the total variance.

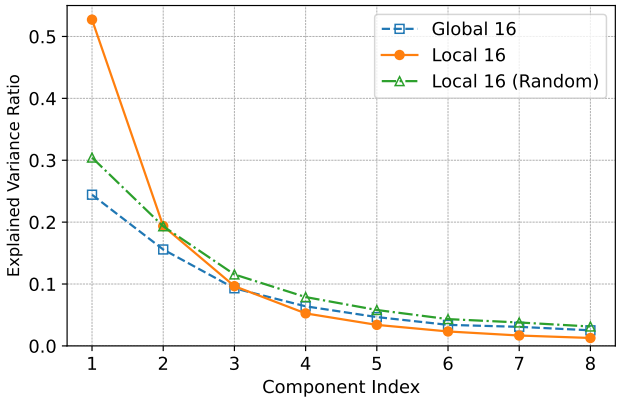

Figure 1. Explained variance ratio for local and global PCA.

	Global 16	Global 64	Local 16	Local 16(Random)
L1 Error	0.0159	0.0092	0.0042	0.0158

Table 1. Error analysis for global and local PCA.

We further emphasize that this correlating property can only be captured by grouping based on locality. To demonstrate this, we group Gaussians to random groups, perform local PCA and compute errors, as shown in Tab. 1. We also report the explained variance ratios in Fig. 1. This random grouping design produces large errors and poorly concentrated information in the principal components.

The above experiments shows that using local blendshapes to model Gaussians can produce lower error. Unlike SqueezeMe [2], which obtains blendshapes by distilling the output of a convolutional network, we allow our model to learn the local blendshapes during optimization.

2. Pruning Visualization

We visualize the Gaussians on the body that keep color blendshapes across different identities, as shown in Fig. 2. These Gaussians are mainly located in regions with rich pose-dependent appearance, such as the clothing wrinkles and arms. Other body parts, like lower legs and shoes, show little appearance change along with pose change. Therefore, the blendshapes of Gaussians at these places can be pruned. We note that as the avatar's arm poses change, the lighting and appearance of the arms also change. Therefore many Gaussians' blendshapes are retained on the arms to model this variation.

Figure 2. Visualization of Gaussians that contain the color blendshapes. The green points are the Gaussians whose color blendshapes have not been pruned.

3. Implementation Details

MLP Architecture. The MLP on each part of the body contains three hidden layer. Each layer's size is 64. For 256 MLPs, all the parameters take about 7MB storage with float16 quantization.

Novel Pose Animation. We follow AnimatableGS [3] and mmlphuman [5] to use PCA to project the novel pose and novel expression into the distribution of the training poses and expressions to prevent unpleasant rendering results.

4. Partition Number Ablation

We ablate the number of body partitions N_G to evaluate its effect on quality, speed, and model size. Fig. 3 shows how PSNR, FPS, and model size vary with the partition count. The data is measured on the training pose and novel view of avatarrex_zzr sequence. Rather than dividing the body into 24 parts based on the SMPL joints, we partition it into more parts to capture finer-grained local structure and improve reconstruction quality. However, increasing the number of partitions also increases model size and reduces rendering speed. A balanced choice among model size, rendering speed, and visual quality is $N_G = 256$ partitions.

5. Pruning Threshold N_P Ablation

We ablate the pruning threshold N_P that controls how many Gaussians retain their blendshapes after pruning. Fig. 4 shows how PSNR, FPS, and model size vary with N_P. To balance quality, model size, and rendering speed, we empirically choose $N_P = 20K$ in our experiments. We note that

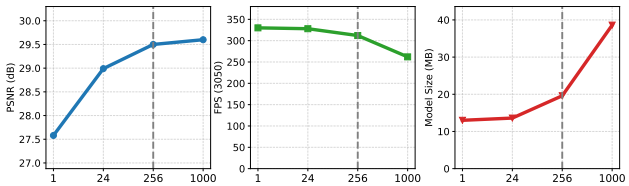

Figure 3. Ablation on the number of body partitions N_G.

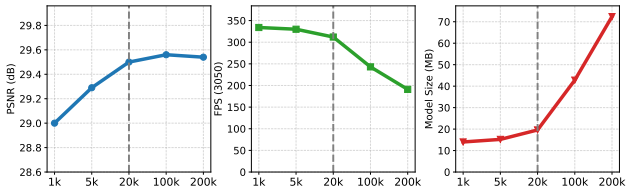

Figure 4. Ablation on the pruning threshold N_P.

N_P is applied separately to each attribute type, i.e., the rotation, color, and scale attributes will each retain 20K blendshapes after pruning.

6. Mobile Performance Analysis

To evaluate the stability of rendering performance on mobile devices, we measure the FPS over a continuous 20-minute session, as shown in Fig. 5. On the Snapdragon 8 Elite device, the rendering maintains 120 FPS for the first 10 minutes before thermal throttling reduces it to approximately 90 FPS for the remainder of the session. We also test on mid-range phones with Snapdragon 8 Gen 1 and Google Tensor G3 chips (released 4 and 3 years ago respectively), both achieving 60 FPS, demonstrating that our method is

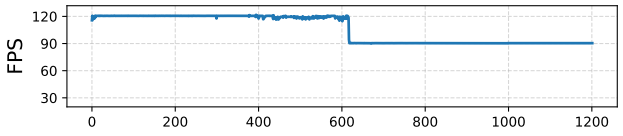

Figure 5. FPS over 20 minutes of continuous rendering on a mobile device (Snapdragon 8 Elite).

Figure 6. Failure case of the pruning strategy.

not limited to high-end devices.

7. SplattingAvatar Speed Comparison

SplattingAvatar [4] cannot model pose-dependent appearance so we didn't include in the quality evaluation. For speed, SplattingAvatar achieves 50 FPS_{3050}, while our method achieves 312 FPS_{3050}. When not predicting dynamic Gaussians, our method reaches 410 FPS_{3050} (dynamic-Gaussian-prediction time / total time = 0.695 ms / 3.053 ms per frame), indicating that predicting dynamic appearance does not add significant overheads.

8. Limitation

For some challenging cases, *e.g.*, an avatar wearing a loose garment, the pruned results may be worse than the unpruned ones, as shown in Fig. 6. This may be because 20K blendshapes are still insufficient to represent such complex appearance. Keeping different numbers of blendshapes per dataset based on the variance magnitude might address this problem.

Additionally, since our application relies on WebGPU, whose support may be incomplete across different devices and browsers, we find that the application fails to run on some older devices, limiting broader application. Adopting a more mature graphics standard to implement may help address this issue.

References

[1] Jianchuan Chen, Jingchuan Hu, Gaige Wang, Zhonghua Jiang, Tiansong Zhou, Zhiwen Chen, and Chengfei Lv. Taoavatar: Real-time lifelike full-body talking avatars for augmented reality via 3d gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10723–10734, 2025. 1

[2] Forrest Iandola, Stanislav Pidhorskyi, Igor Santesteban, Divam Gupta, Anuj Pahuja, Nemanja Bartolovic, Frank Yu, Emanuel Garbin, Tomas Simon, and Shunsuke Saito. Squeezeme: Mobile-ready distillation of gaussian full-body avatars. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pages 1–11, 2025. 1

[3] Zhe Li, Zerong Zheng, Lizhen Wang, and Yebin Liu. Animatable gaussians: Learning pose-dependent gaussian maps for high-fidelity human avatar modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19711–19722, 2024. 1, 2

[4] 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. 3

[5] Youyi Zhan, Tianjia Shao, Yin Yang, and Kun Zhou. Real-time high-fidelity gaussian human avatars with position-based interpolation of spatially distributed mlps. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26297–26307, 2025. 2