Human Gaussian Splatting: Real-time Rendering of Animatable Avatars

Supplementary Material

We present further analysis of our method. We invite readers to watch the video that summarizes our contributions and demonstrates real-time rendering, whose details are given in section 7. We further present extensive implementation details and hyperparameters setting in section 8, in order to facilitate the reproducibility of our experiments. Finally, we provide more qualitative results of our method on the THuman4 dataset in section 10.

7. Real-time video rendering

Real-time rendering in the supplementary video The attached video showcases real-time novel pose synthesis on the THuman4 dataset [72] at 60fps. This is done by extending the viewer from 3D-GS to dynamic scenarios in order to render videos.

8. Reproducibility details

We have described our main implementation details in the main manuscript. In this section, we further report the hyperparameter values of our pipeline in Table 6. After that, we provide the additional implementation details of our approach, as described as below.

Linear blend skinning Our implementation for linear blend skinning (LBS) follows SMPL-X [42]. Notably, we do not use pose blend shapes before applying deformations on human joints, because the shape estimation and blend shapes obtained upon that may be inaccurate, thus we design MLP to handle pose-dependent deformations. We also remind that in our case, LBS is applied on canonical gaussians only and thus deforming template vertices is not necessary. The learnable per-gaussian skinning weights vector $\mathbf{w}$ is a parameter optimized through gradient descent which can leads to negative values. We apply a ReLU activation on each $\mathbf{w}_j$ and then normalize the vector such that its components sum to 1 to obtain a well defined skinning weights vector. Finally, the transformations matrices $\mathbf{M}_{j,t}$ that encode the rigid deformation of each body joint j for each training timestep t are precomputed before training to improve efficiency.

Learning rates Similar to 3D-GS [16], we use different learning rates for each set of learnable parameters. We leave the learning rates of the original parameters (position, orientation, scaling, colors and opacity) unchanged. Our MLP and the skinning weights vectors $\mathbf{w}$ are optimized with a constant learning rate that is set to $1e^{-4}$. For latent codes $\mathbf{l}$, we use a learning rate of $2.5e^{-3}$.

Table 6. **Hyperparameters values.**

Parameter name	**Value**
λ_{L_1} (in Eqn. 8)	0.8
λ_{ssim} (in Eqn. 8)	0.2
λ_{lpips} (in Eqn. 8)	0.05
λ_{trans} (in Eqn. 8)	0.01
λ_{rot} (in Eqn. 8)	0.001
λ_{s} (in Eqn. 8)	0.001
λ_{mesh} (in Eqn. 8)	0.1
λ_{skn} (in Eqn. 8)	0.001
Dimension of latent code $\mathbf{l}$	16

9. Concurrent works

Gaussian splatting is currently a very active research topic and while this work was under review, many related drivable avatars based on gaussian splatting have been released. We propose a small discussion and refer to Awesome 3D Gaussian splatting for a complete list of related papers.

Similar to HuGS, most gaussian avatars exploit forward deformation of gaussians with LBS to drive the avatar. Local refinement with a neural network is also a popular choice [12, 14, 21, 27] but different designs have been developed. Notably, SplatArmor [12] uses canonical gaussian parameters as input, Animatable Gaussians [27] and ASH [39] use a 2D CNN. The per-gaussian latent code and the shading components proposed by our method are the main advantages of HuGS compared to these approaches. Regarding skinning weights, most methods rely on the template, with the exception of GART [19] that also optimizes these parameters. In contrast with ours, these learnable skinning weights are not defined per-gaussian but in a voxel grid. Finally, similar to ours, most methods optimize the canonical gaussians jointly with the rest of the pipeline. In contrast, Animatable Gaussians [27] and ASH [39] import a template (such as a SDF) where the body shape as already been fitted on the subject. This design choice adds an expensive pre-processing step but also seems to exhibit very good results. We expect follow-up research to build on this large amount of proposals to push gaussian-based animatable avatars forward.

10. Qualitative analysis

We display in Figure 8 qualitative results of the HuGS method for subject01 and subject02 sequences from the THuman4 dataset [72] for novel pose synthesis. Note that no quantitative comparison is done on these subjects be-

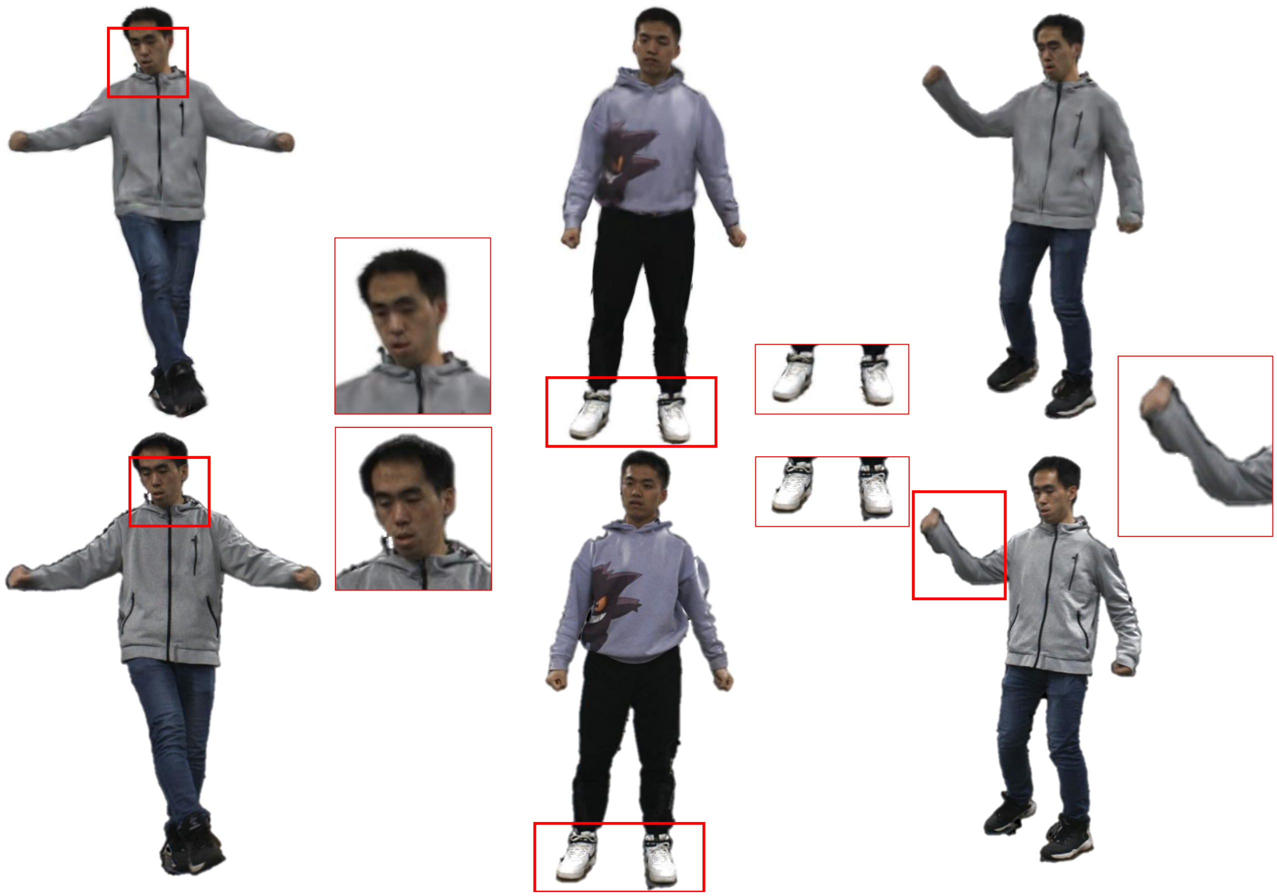

Figure 8. **Qualitative visualization of HuGS novel pose synthesis on THuman4 dataset.**

cause the evaluation setup has not been released by the dataset authors. We observe that our method is able to fit the subjects with precise details, such as the black hood button (left picture) or the shoes (middle), and render the target body pose with high fidelity. However, we also showcase inaccuracies in the dataset caused by segmentation masks and motion blur that are observed regularly on training images and thus create artifacts in the learned model and degrade the overall rendering quality on these subjects.