

PGC: Physics-Based Gaussian Cloth from a Single Pose

Supplementary Material

A. Qualitative Results and Video

Please see the accompanying supplementary video for an introduction and walkthrough of our approach, as well as additional qualitative results. This video shows renders of our simulation-ready garments on unseen motion sequences with outdoor and indoor environment maps. This demonstrates the ability of our method to generalize to novel poses and various lighting conditions. We provide a side-by-side comparison against the original “3D Gaussian Splatting” (3DGS-Only) method. Not only is the shading baked into the garments produced by 3DGS-Only, they are also not relightable. Lastly, we demonstrate 360 relighting, where the garments are dynamically relit with a rotating illumination map.

B. Mesh-Embedded 3DGS Reconstruction

In this section, we provide additional details on the mesh-embedded 3DGS representation and optimization approach.

Mesh Reconstruction: We remesh the garment to 11-18k vertices using Houdini [5], though any remeshing software would suffice. After remeshing, we unwrap the mesh in Blender [1] to obtain the UV map.

Mesh-embedded Gaussian Splat Representation: Once we have sampled Gaussians on the mesh surface, we follow Qian et al. [10] to define each Gaussian in the local coordinate frame of its parent triangle on the mesh. This allows us to deform the Gaussians as the mesh deforms at test time. The origin of the local coordinate frame τ is the centroid of the triangle. The face orientation is defined with three vectors: one of the edges normalized, the normal vector of the triangle, and their cross product. They are concatenated as column vectors to form the rotation matrix $R \in \text{SO}(3)$.

We represent each Gaussian’s rotation quaternion $r \in \mathbb{H}$, position $\mu \in \mathbb{R}^3$, and scale $s \in \mathbb{R}_+^3$ in the local coordinate frame of its parent triangle. To render the Gaussians, we transform the Gaussian parameters into world coordinates with the following equations:

$$r' = Rr, \quad (1)$$

$$\mu' = kR\mu + \tau, \quad (2)$$

$$s' = ks, \quad (3)$$

where $k \in \mathbb{R}_+$ is the scale of the triangle defined as $\frac{B+H}{2}$, where B and H are the base and height of the triangle, respectively.

3DGS Initialization: We initialize our Gaussian model with 1 million splats on the surface of the reconstructed full-body mesh. Note that only the subset of splats that belong

to the garment segmented mesh are kept for inference, so in practice we use much fewer than 1 million splats to model the garment. Following [7] we initialize opacity α to 0.1, and initialize only the first three coefficients of spherical harmonics ϕ using randomly sampled RGB color values, setting the remaining coefficients to zeros. We sample RGB colors uniformly from $c \sim \mathcal{U}(0.5, 0.7)^3$ where $c \in [0, 1]^3$ is the RGB color of the Gaussian splat. The local Gaussian position is computed by transforming the sampled splat positions from world to local coordinates. Following [10], we initialize the local Gaussians with identity rotation and unit scale s .

3DGS Optimization: We apply a mask regularization loss term as described in Sec. 3.2 of the main document. The kernel size K_{fg} is computed as the image height scaled by a mask erosion factor γ_{fg} . The loss is weighted by λ_{fg} .

Our optimization largely follows the settings from [7], with some modifications to the values. The full list of optimization parameters and their values are provided in Tab. 1. In addition to the original implementation, we also provide the hyperparameters to additional modules included in our implementation in Tab. 2.

Table 1. **3DGS Optimization Settings:** We provide the optimization parameters and their values for 3DGS optimization. The parameters are derived from the original 3DGS implementation [7], with some modifications to the values.

Parameter	Setting
num. optimization iterations	30000
optimizer	Adam
position learning rate (init)	0.5e-4
position learning rate (final)	1.0e-7
position learning rate (max steps)	30000
feature learning rate	0.005
opacity learning rate	0.005
scaling learning rate	0.005
rotation learning rate	0.005
SSIM loss weight λ_{SSIM}	0.2
SH increase frequency	500

Caveats: After 3DGS optimization described in Sec. 3.2 of the main document, the mesh is culled to remove triangles deemed to not belong to the garment of interest. Due to the nature of global 3DGS optimization, some Gaussians that contributed to the color of the garment may belong to triangles representing other parts of the reconstructed body mesh. Thus culling triangles after 3DGS optimization subtly reduce the opacity in some areas of the garment (e.g. bot-

Table 2. **Additional 3DGS Optimization Settings:** We provide the optimization parameters and settings for additional modules introduced in our implementation below.

Parameter	Setting
mask erosion factor γ_{fg}	0.03
mask loss weight λ_{fg}	0.1

tom of the T-shirt as visible in Fig. 3 or beside the left pocket of the cardigan in Fig. 1 of the main document). We believe this is a minor issue and not a fundamental limitation of the method. For instance, to improve surface opaqueness for each pixel, we may eliminate contributions from Gaussians on back-facing triangles representing the interior of the garment, thus resulting in a single front facing cloth layer remaining, responsible for the foreground reconstruction during the 3DGS optimization step.

C. PBR Appearance Reconstruction

We provide implementation details for the comparisons with Lambertian and Disney BRDF [2] PBR models in Fig 3. (b) and (c), as well as our PBR model (d) from the main document. We use Mitsuba [6] as our differentiable renderer. We report hyperparameters for optimization in Tab. 3. We use a constant lighting model to approximate the overall lighting employed by our capture system, as described in Sec. 4 of the main document. The base color generated by the neural network (Sec 3.3 of the main document), despite being free from baked-in lighting and shadowing, could still contain colors influenced by the lights. To account for such color scaling, we additionally optimize the RGB radiance of the constant lighting model. We report the initial and optimized parameters per model in Tab. 4 and Tab. 5, respectively. We also report PSNR values for the different PBR models, per garment, in Tab. 6.

Table 3. **Optimization Settings for Shading Reconstruction:** We provide the optimization parameters for shading reconstruction, which are implemented in the Mitsuba [6] differentiable renderer.

Parameter	Setting
num. optimization iterations	3000
optimizer	Adam
learning rate	0.01
samples per pixel	4
integrator type	path
lighting model	constant

Table 4. **PBR Parameter Initialization:** We report the initializations used for each PBR model. Parameters that are not listed use the default values in Mitsuba.

PBR Model	Parameter	Setting
Lambertian	lighting radiance	(1.00, 1.00, 1.00)
Disney BRDF [2]	lighting radiance	(1.00, 1.00, 1.00)
	roughness	0.8
	sheen	1.4
	specular	0.0
Ours	lighting radiance	(1.00, 1.00, 1.00)
	roughness	0.8
	sheen	1.4
	sheen color	(0.75, 0.73, 0.27)
	sheen roughness	0.5
	specular	0.0

D. Gaussian—PBR Hybrid Rendering

Reconstruction Error: Dropping the subscript t for simplicity, we can write the reconstruction error ϵ (when compared to the real world images) as

$$\begin{aligned}
\epsilon &= \mathbf{I} - \hat{\mathbf{I}} \\
&= \mathbf{I} - (h(\mathbf{G}) + l(\mathbf{S})) \\
&= \mathbf{I} - (\mathbf{G} - l(\mathbf{G})) - l(\mathbf{S}) \\
&= \underbrace{l(\mathbf{G}) - l(\mathbf{S})}_{\text{additional error}} + \underbrace{\mathbf{I} - \mathbf{G}}_{\text{3DGS error}}.
\end{aligned} \tag{4}$$

For training frames, given that $\mathbf{I} - \mathbf{G}$ is small, this suggests that the error mostly occurs in PBR reconstruction. It also suggests that it is sufficient for the PBR reconstruction to approximate the ground truth images only in the low frequency mode. At test time, $\mathbf{I} - \mathbf{G}$ is no longer small due to baked-in lighting and shadowing in \mathbf{G} . Intuitively, however, we expect novel shading missing from \mathbf{G} to be reintroduced by $l(\mathbf{S}) - l(\mathbf{G})$. Additional error could be introduced at test time from mismatched geometry since our method does not rely on tracking.

E. Experiments

Optimization and Inference Efficiency: We report performance and runtime efficiency in Tab. 7. All 3DGS rendering is performed on an NVIDIA A100 GPU, while all PBR rendering and simulation is run on an NVIDIA RTX 3080 GPU. Note that while the reported timing is not real time, each of the components have real-time counterparts, making the method compatible with real-time pipelines.

Simulation: We use eXtended Position Based Dynamics (XPBD) [9] as our choice of simulator. The simulation parameters are provided in Tab. 8. In all examples presented in the main paper, the material parameters were

Table 5. **Optimized PBR Parameters:** We report the optimized parameters per PBR-model, for each garment.

PBR Model	Parameter	Garments			
		T-Shirt	Dress	Fleece	Cardigan
Lambertian	lighting radiance	(4.15, 5.16, 5.27)	(1.39, 1.43, 1.44)	(7.07, 6.82, 8.18)	(3.56, 3.59, 5.53)
Disney BRDF [2]	lighting radiance	(3.82, 7.27, 7.56)	(1.32, 1.44, 1.51)	(6.32, 5.97, 6.23)	(3.00, 3.02, 4.02)
	roughness	0.09	0.00	1.00	0.85
	sheen	0.56	4.50	0.57	1.46
	sheen tint	0.66	0.00	0.39	0.56
Ours	lighting radiance	(2.85, 3.34, 3.31)	(1.02, 1.12, 1.24)	(4.86, 4.75, 5.01)	(2.88, 2.90, 4.62)
	roughness	0.97	0.53	1.00	0.99
	sheen	0.35	1.05	0.22	0.40
	sheen color	(0.35, 0.22, 0.15)	(0.51, 0.48, 0.34)	(0.23, 0.24, 0.13)	(0.39, 0.38, 0.13)
	sheen roughness	0.66	0.76	0.62	0.42

Table 6. **Quantitative Results on Shading Reconstruction:** We report PSNR per-PBR model, on each garment.

PBR Model	T-Shirt	Dress	Fleece	Cardigan
Lambertian	34.76	21.94	33.01	26.55
Disney BRDF [2]	37.57	22.26	34.29	27.10
Ours	37.67	22.93	34.40	27.23

Table 7. **Optimization and Inference Efficiency:** We report the efficiency for both optimization and inference of our method.

Stage	Module	Runtime
Optimization	3DGS optimization	3h
	PBR optimization	12m
Inference	XPBD Simulation	11 FPS
	PBR rendering	3.2 FPS
	3DGS rendering	1.6 FPS
	Image filtering	14.4 FPS

kept constant. However, to better approximate the behavior of thicker garments like the fleece and cardigan we set the bending stiffness to be x10 larger than what is used for the t-shirt and dress simulations in Fig. 3.

Table 8. **Simulation Parameters:** We provide the parameters that we use for garment simulation. We use the XPBD simulator [9] with spring constraints for both stretching and bending.

Parameter	Setting
cloth-body offset	0.4 cm
frame rate	30
substeps per frame	30
XPBD iterations	20

Baseline Setup: For Animatable Gaussians, we train on all views with a video sequence of diverse poses containing roughly 8k frames. SCARF is a NeRF-based method that reconstructs animatable clothed humans from monocular video. We train on the same training data as was used for our method, excluding extreme camera poses (because PIXIE, the method for SMPL body estimation, struggles with extreme camera poses [3]). Although it is a method for monocular video, we reframe our multi-view static pose setting as a monocular video by concatenating all the views into a video, as done in [11]. Animatable Gaussians takes 3 days to train. SCARF takes 14 hours.

Additional Metrics: We report additional metrics, structural similarity (SSIM) [13] and peak signal-to-noise ratio (PSNR), comparing with baselines and ablations in Tab. 9. Note that our method does not optimize for the garment rest shape. Instead, we use the reconstructed geometry as the rest shape for simulation. While our results are more crisp and produce more realistic dynamics (see Fig 7. of the main document), our method can produce sagging and in general is not guaranteed to match the geometry from ground truth frames in novel poses. We believe this is the reason why our method (and ablated versions of it) achieve lower scores on metrics that are sensitive to pixel alignment. Nevertheless, we report these metrics for reference.

Novel motion: We evaluate the generalization of our garments to novel motions in Fig. 3. Please see the supplementary video to view the motion sequences.

Lighting 360: We show our garments relit under different rotations of the environment in Fig. 4. Please see the supplementary video for additional renders.

Blur Analysis: We analyze the effect of blur kernel size on training reconstruction for the *cardigan* example. We report feature similarity (FSIM) [12] index of our reconstruction in relation to the blur kernel size in Fig. 1. Large amount of blur reduces the reconstruction error (Eq. (4)) since it is

Table 9. **Quantitative Comparisons:** We report additional metrics comparing with existing work and ablated versions of our method. Note that our method (and ablations) do not optimize for the garment rest shape. We believe this is the reason that baselines outperform on metrics that are sensitive to pixel alignment.

Method	SSIM \uparrow	PSNR \uparrow
SCARF [4]	0.937	25.96
Animatable Gaussians [8]	0.945	29.75
3DGS-Only	0.938	28.62
PBR-Only	0.947	28.65
Ours	0.939	28.37

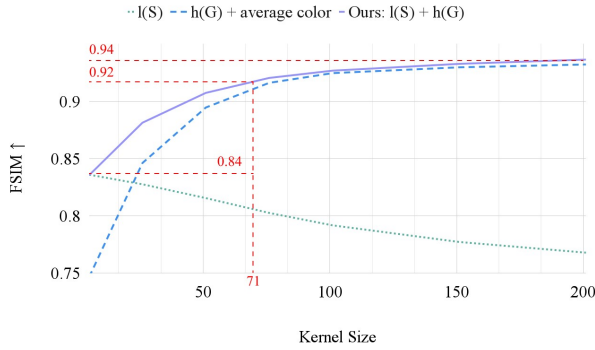


Figure 1. **Blur Analysis.** We analyze the reconstruction performance on the training frame of the *cardigan* garment, across different levels of blur. Compared to no blur, our selected kernel size of 71 pixels (shown in red) improves the FSIM metric by 8%. Increasing the blur kernel further produces no more than a 2% increase in FSIM, while decreasing the generalizability of the appearance model to novel poses. FSIM for $l(S)$ and $h(G)$ +average color are plotted for reference showing that our model produces quantitatively better results at all blur levels.

less challenging for PBR reconstruction to approximate an overly blurred image. However, in the meantime, this introduces more contribution from 3DGS, which prevents us from generalizing to novel poses at testing time. To strike a balance, we select a blur kernel size 71×71 for our training image with resolution 4096×2668 . We use the same kernel size for all our examples, however the optimal blur kernel size ought to vary by garment type. Garments with high-frequency details (thick wool knits) necessitate larger kernels to accurately capture these intricate features. One way to automate the process of choosing a garment-dependent kernel size is by progressively blurring the input image until its gradient magnitudes fall below a threshold, indicating that high-frequency details have been removed. We leave this extension to future work.

ActorsHQ dataset: We show that our method works on the ActorsHQ dataset in Fig. 2 showing the reconstruc-

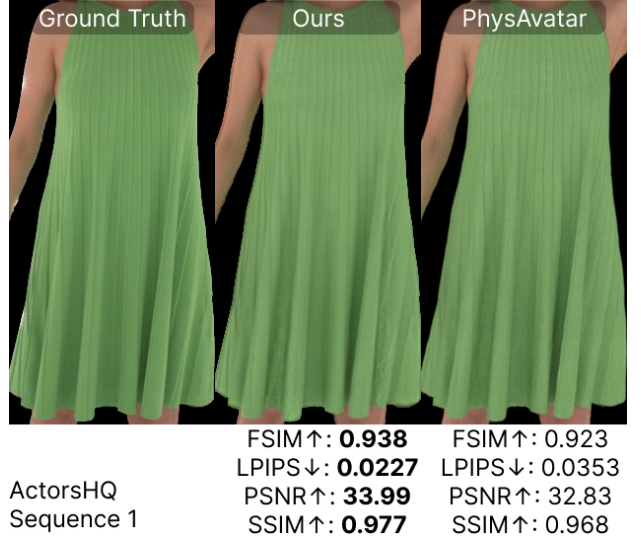


Figure 2. **ActorsHQ example.** This figure demonstrates our method applied to the popular ActorsHQ dataset.

tion for an unseen pose. The quality metrics FSIM=0.938, LPIPS=0.0227, PSNR=33.99, SSIM=0.977 are similar to those in the main paper and slightly outperform PhysAvatar for the selected frame.

References

- [1] Blender. Blender. <https://www.blender.org/>, 2024. Accessed on November 2024. 1
- [2] Brent Burley and Walt Disney Animation Studios. Physically-based shading at disney. In *Acm Siggraph*, pages 1–7. vol. 2012, 2012. 2, 3
- [3] Yao Feng, Vasileios Choutas, Timo Bolkart, Dimitrios Tzionas, and Michael Black. Collaborative regression of expressive bodies using moderation. In *International Conference on 3D Vision (3DV)*, pages 792–804, 2021. 3
- [4] Yao Feng, Jinlong Yang, Marc Pollefeys, Michael J. Black, and Timo Bolkart. Capturing and animation of body and clothing from monocular video. In *SIGGRAPH Asia 2022 Conference Papers*, 2022. 4
- [5] Houdini. Houdini. <https://www.sidefx.com/products/houdini/>, 2024. Accessed on November 2024. 1
- [6] Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, Merlin Nimier-David, Delio Vicini, Tizian Zeltner, Baptiste Nicolet, Miguel Crespo, Vincent Leroy, and Ziyi Zhang. Mitsuba 3 renderer, 2022. <https://mitsuba-renderer.org>. 2
- [7] 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
- [8] Zhe Li, Zerong Zheng, Lizhen Wang, and Yebin Liu. Animatable gaussians: Learning pose-dependent gaussian maps for high-fidelity human avatar modeling. In *CVPR*, 2024. 4
- [9] Miles Macklin, Matthias Müller, and Nuttapong Chentanez. Xpbd: position-based simulation of compliant constrained

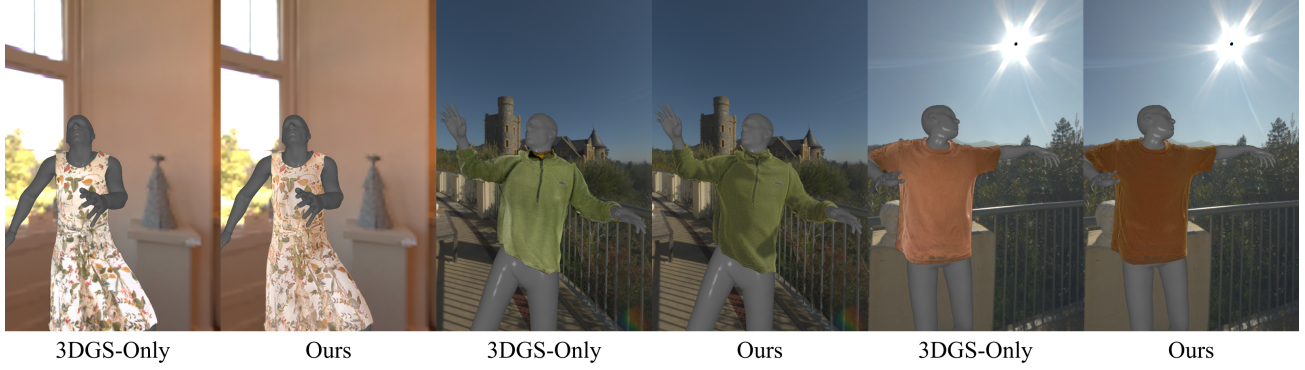


Figure 3. **Novel Motion**. We evaluate our method on novel motion sequences.

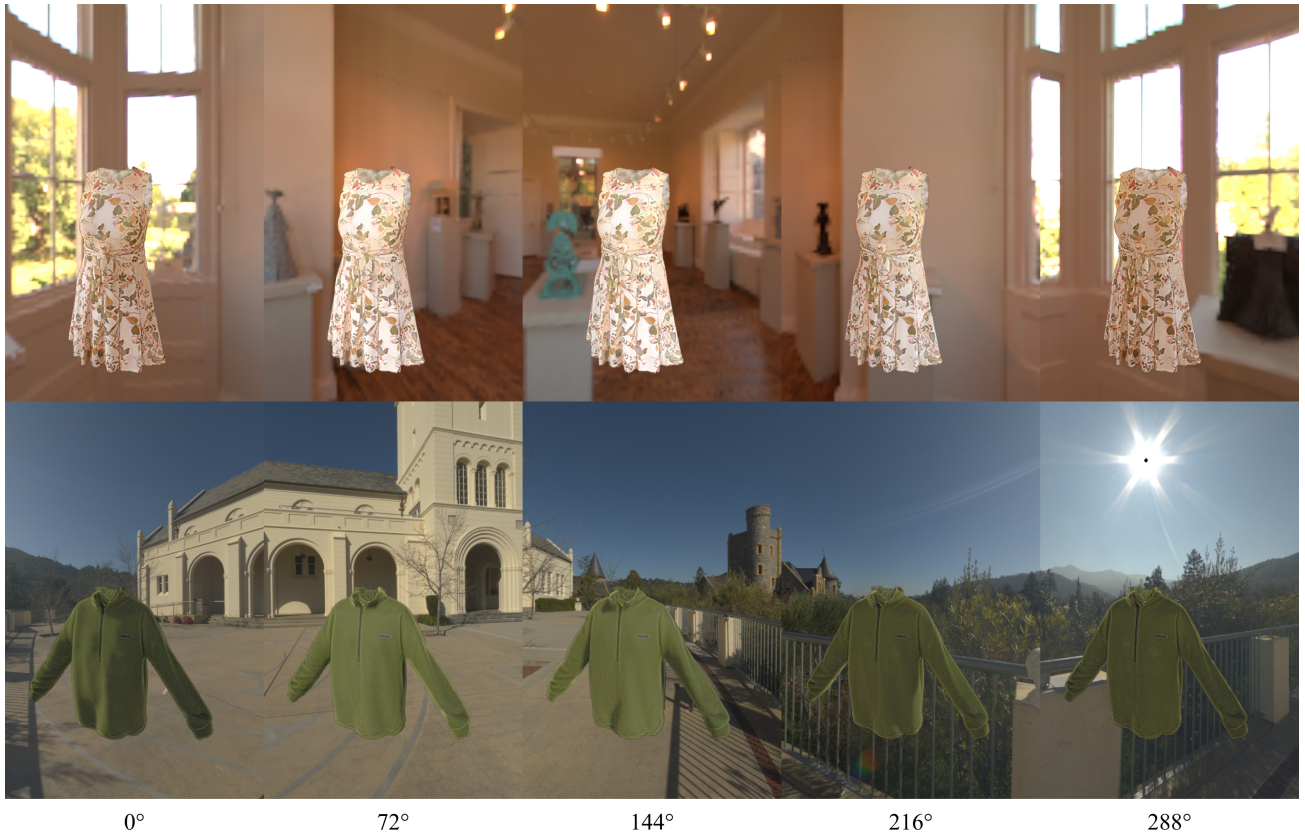


Figure 4. **Lighting 360°**. We show our garments under different rotations of the environment map.

dynamics. In *Proceedings of the 9th International Conference on Motion in Games*, pages 49–54, 2016. 2, 3

- [10] Shenhan Qian, Tobias Kirschstein, Liam Schoneveld, Davide Davoli, Simon Giebenhain, and Matthias Nießner. Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20299–20309, 2024. 1
- [11] Boxiang Rong, Artur Grigorev, Wenbo Wang, Michael J. Black, Bernhard Thomaszewski, Christina Tsalicoglou,

and Otmar Hilliges. Gaussian garments: Reconstructing simulation-ready clothing with photorealistic appearance from multi-view video, 2024. 3

- [12] Lin Zhang, Lei Zhang, Xuanqin Mou, and David Zhang. Fsim: A feature similarity index for image quality assessment. *IEEE Transactions on Image Processing*, 20(8):2378–2386, 2011. 3
- [13] Wang Zhou. Image quality assessment: from error measurement to structural similarity. *IEEE TIP*, 13:600–613, 2004. 3