DR-GS: Supplementary Material

Jiaxin Li[1,2], Tong Wu[4], Yi Wei[4], Tailin Wu[2], and Li Zhang[1,3,†]

[1] Shanghai Innovation Institute
[2] Department of Artificial Intelligence, Westlake University
[3] School of Data Science, Fudan University
[4] Central Media Technology Institute, Huawei

This supplementary material provides additional methodological details and qualitative results for DR-GS. Sec. 1 presents the mesh-based reinitialization procedure and the associated Gaussian parameterization. Sec. 2 describes the physical simulation details based on MPM, and Sec. 3 provides additional visual results demonstrating parameter disentanglement and deformation robustness.

DR-GS supports reconstruction, relighting, physical simulation, and 3D animation, and more dynamic results are provided in the supplementary video at https://jiaxinlia.github.io/DR-GS/.

1 Details of Mesh-based Reinitialization

We summarize the notation used in Sec. 3.2 of the main paper in Table 1.

Table 1: Notation used in Sec. 3.2 of the main paper.

Symbol	Description
$\mathcal{M} = (\mathcal{V}, \mathcal{F})$	triangular mesh with vertices $\mathcal{V}$ and faces $\mathcal{F}$
f_j	j-th triangle face
$\boldsymbol{v}_{j1}, \boldsymbol{v}_{j2}, \boldsymbol{v}_{j3}$	vertices of face f_j
$\boldsymbol{n}_j$	face normal of f_j
$\boldsymbol{R}_j$	Gaussian rotation aligned with $\boldsymbol{n}_j$
$\boldsymbol{c}_j$	centroid of face f_j
$\tilde{d}_j$	distance to the nearest neighboring face center
r	circumradius of triangle $\{\boldsymbol{v}_a, \boldsymbol{v}_b, \boldsymbol{v}_c\}$
Θ^P	pretrained Gaussian set
Θ^M	mesh-based Gaussian set
$\boldsymbol{p}_i$	center of pretrained Gaussian i
α_i	opacity of Gaussian i
d_j	signed offset along the face normal
$\boldsymbol{\omega}_j$	barycentric coordinates on face f_j
$\boldsymbol{s}_j$	Gaussian log-scale parameter
$\Pi_{\mathcal{M}}$	projection operator mapping a point to its closest mesh face
$\mathcal{I}_j$	set of Gaussians projected onto face f_j
τ	threshold ensuring the projected barycentric coordinates lie inside a triangle
BVH	bounding volume hierarchy used for fast closest-face queries

To establish a geometry-aware parameterization, we reinitialize the pretrained 2D Gaussians on the reconstructed triangular mesh. Given a mesh $\mathcal{M} = (\mathcal{V}, \mathcal{F})$, we attach one Gaussian to each triangle face $f_j \in \mathcal{F}$ and represent it as $\Theta^M = \{d_j, \boldsymbol{\omega}_j, \boldsymbol{s}_j, \boldsymbol{R}_j, \alpha_j\}_{j=1}^{N_f}$, where d_j denotes the signed offset along the face normal, $\boldsymbol{\omega}_j = (\omega_a, \omega_b, \omega_c)$ are barycentric coordinates on the triangle, $\boldsymbol{s}_j$ is the Gaussian log-scale, $\boldsymbol{R}_j$ is the local rotation aligned with the face normal $\boldsymbol{n}_j$, and α_j is the opacity. The Gaussian center is parameterized by barycentric interpolation of the triangle vertices together with a normal-direction offset,

$$\boldsymbol{p}_j = \omega_a \boldsymbol{v}_a + \omega_b \boldsymbol{v}_b + \omega_c \boldsymbol{v}_c + d_j \, r_j \, \boldsymbol{n}_j,$$

where $\{\boldsymbol{v}_a, \boldsymbol{v}_b, \boldsymbol{v}_c\}$ are the vertices of face f_j, $\boldsymbol{n}_j$ is the face normal, and r_j is the circumradius used to normalize the offset magnitude. For initialization, the barycentric coordinates are set to $(1/3, 1/3, 1/3)$ so that the Gaussian center starts at the triangle centroid, and the signed offset is initialized as $d_j = 0$. The Gaussian orientation $\boldsymbol{R}_j$ is aligned with the face normal, while the initial scale is determined by a geometry-adaptive estimate: we compute the centroid $\boldsymbol{c}_j$ of each face and use the distance to its nearest neighboring face center $\tilde{d}_j$ to initialize the log-scale as $\boldsymbol{s}_j = \log(\tilde{d}_j + \epsilon)\mathbf{1}_2$, where ϵ ensures numerical stability.

To transfer appearance information from the pretrained Gaussian set Θ^P, we project each Gaussian center $\boldsymbol{p}_i$ onto the mesh surface using a bounding volume hierarchy (BVH). The projection returns the closest face index f_i, a signed distance, and barycentric coordinates $\boldsymbol{\omega}_i$. To avoid unstable assignments near triangle boundaries, we retain only interior projections satisfying $\min(\boldsymbol{\omega}_i) > \tau$, where τ is a small threshold. For each face f_j, all valid projected Gaussians are collected into an index set $\mathcal{I}_j$, and we aggregate their parameters to initialize the mesh-based Gaussian by averaging barycentric coordinates and opacity,

$$\boldsymbol{\omega}_j = \frac{1}{|\mathcal{I}_j|} \sum_{i \in \mathcal{I}_j} \boldsymbol{\omega}_i, \qquad \alpha_j = \frac{1}{|\mathcal{I}_j|} \sum_{i \in \mathcal{I}_j} \alpha_i.$$

This procedure converts the original free-form Gaussian representation into a mesh-attached parameterization that is explicitly aware of surface geometry and topology, providing a stable initialization for subsequent optimization.

2 Details of Physical Simulation

Our framework is compatible with a wide range of physical simulation methods. For the particle-driven experiments, we adopt the Material Point Method (MPM) [1], a hybrid simulation technique that combines the advantages of Lagrangian particles and Eulerian grids. In MPM, the continuum medium is discretized into a set of material points, where each particle i carries physical quantities including position $\boldsymbol{x}_t(i)$, velocity $\boldsymbol{v}_t(i)$, and elastic deformation gradient $\boldsymbol{F}_t^e(i)$ at timestep t. These particles store the state of the material and move in a Lagrangian manner, while the governing momentum equations are solved on an underlying Eulerian grid to improve numerical stability and computational efficiency.

During each simulation step, particle quantities are first transferred to nearby grid nodes using interpolation weights defined by shape functions. Specifically, particle mass and momentum are accumulated on grid nodes, followed by the computation of internal elastic forces derived from the stress tensor and external forces induced by body forces. The grid velocities are then updated by integrating these forces over the timestep. Subsequently, the updated grid velocities are interpolated back to particles to obtain particle velocities. The deformation gradient of each particle is updated using the velocity gradient estimated from the grid, and particle positions are advanced using explicit time integration. This particle–grid–particle transfer scheme enables MPM to robustly handle large deformations and complex material dynamics, while avoiding mesh distortion issues commonly encountered in purely Lagrangian approaches and the interface tracking difficulties in purely Eulerian formulations. The complete time integration procedure is summarized in Alg. 1.

Algorithm 1 Material Point Method (MPM) Time Integration

Input: Particle positions $\boldsymbol{x}_t(i) \in \mathbb{R}^3$, velocities $\boldsymbol{v}_t(i) \in \mathbb{R}^3$, elastic deformation gradients $\boldsymbol{F}_t^e(i) \in \mathbb{R}^{3\times3}$ for $i = 1, \ldots, N_P$, time step $\Delta t \in \mathbb{R}^+$, external force field $\boldsymbol{b}(\boldsymbol{x}) \in \mathbb{R}^3$
Output: Updated positions $\boldsymbol{x}_{t+1}(i) \in \mathbb{R}^3$, velocities $\boldsymbol{v}_{t+1}(i) \in \mathbb{R}^3$, trial elastic deformation gradients $\boldsymbol{F}_{t+1}^{e,\mathrm{trial}}(i) \in \mathbb{R}^{3\times3}$

Initialize:
  $\boldsymbol{v}_{t+1}(b) \leftarrow \mathbf{0}\ \forall b \in \{1, \ldots, N_G\}$
  $\boldsymbol{f}_t^{\sigma}(b) \leftarrow \mathbf{0}, \boldsymbol{f}_t^{e}(b) \leftarrow \mathbf{0}$
for each batch $\mathcal{B} \subseteq \{1, \ldots, N_G\}$ **do**
  $m_t(b) \leftarrow \sum_{i\in\mathcal{N}_b} N_b(\boldsymbol{x}_t(i))M(i)$ ▷ Mass transfer
  $m_t(b)\boldsymbol{v}_t(b) \leftarrow \sum_{i\in\mathcal{N}_b} N_b(\boldsymbol{x}_t(i))M(i)\boldsymbol{v}_t(i)$ ▷ Momentum transfer
  $\boldsymbol{f}_t^{\sigma}(b) \leftarrow -\sum_{i\in\mathcal{N}_b} \frac{J(\boldsymbol{F}_t^e(i))}{\rho_0}\boldsymbol{\sigma}(\boldsymbol{F}_t^e(i))\nabla N_b(\boldsymbol{x}_t(i))M(i)$ ▷ Internal force
  $\boldsymbol{f}_t^{e}(b) \leftarrow \sum_{i\in\mathcal{N}_b} \frac{J(\boldsymbol{F}_t^e(i))}{\rho_0}\boldsymbol{b}(\boldsymbol{x}_t(i))N_b(\boldsymbol{x}_t(i))M(i)$ ▷ External force
  $\boldsymbol{v}_{t+1}(b) \leftarrow \boldsymbol{v}_t(b) + \frac{\boldsymbol{f}_t^{\sigma}(b)+\boldsymbol{f}_t^{e}(b)}{m_t(b)}\Delta t$ ▷ Grid velocity update
end for
for each particle $i \in \{1, \ldots, N_P\}$ **do**
  $\boldsymbol{v}_{t+1}(i) \leftarrow \sum_{b\in\mathcal{N}_i} N_b(\boldsymbol{x}_t(i))\boldsymbol{v}_{t+1}(b)$ ▷ Velocity interpolation
  $\boldsymbol{F}_{t+1}^{e,\mathrm{trial}}(i) \leftarrow (\boldsymbol{I} + \sum_{b\in\mathcal{N}_i} \boldsymbol{v}_{t+1}(b) \otimes \nabla N_b(\boldsymbol{x}_t(i))\Delta t)\boldsymbol{F}_t^e(i)$ ▷ Deformation update
end for

3 More Results

Parameter disentanglement of DR-GS. Fig. 1 presents qualitative results demonstrating the parameter disentanglement capability of DR-GS across all eight scenes of the GlossySynthetic dataset [2]. For each scene, we show the reconstructed appearance in a novel view, along with the predicted intrinsic material attributes including albedo, metallic, and roughness maps. In addition, the figure visualizes the estimated surface normals and the ray-traced visibility maps derived from the learned 2D Gaussian representation.

The results indicate that DR-GS successfully decomposes complex specular objects into physically meaningful components while maintaining accurate geometry and view-dependent appearance. The predicted albedo maps capture the intrinsic color independent of lighting, while the metallic and roughness maps reflect the spatially varying reflectance properties of different materials. Meanwhile, the reconstructed normal maps remain geometrically consistent with the underlying shape, and the visibility maps correctly capture occlusion patterns required for physically-based shading. Together, these visualizations demonstrate that DR-GS learns a well-disentangled representation of geometry, material, and visibility, enabling faithful rendering and physically interpretable material decomposition.

Physical simulation results for glossy objects. Fig. 2 shows continuous deformation sequences generated by soft-body simulations of objects from the GlossySynthetic dataset [2], rendered from multiple viewing angles. The simulations produce diverse deformation states with significant geometric variations, allowing us to evaluate the robustness of DR-GS under dynamic shape changes. Despite large deformations, DR-GS faithfully reproduces view-dependent reflections and environment lighting effects on glossy surfaces. The results demonstrate that the learned representation remains consistent under complex geometry variations, enabling stable rendering of specular appearance and environment reflections throughout the deformation process.

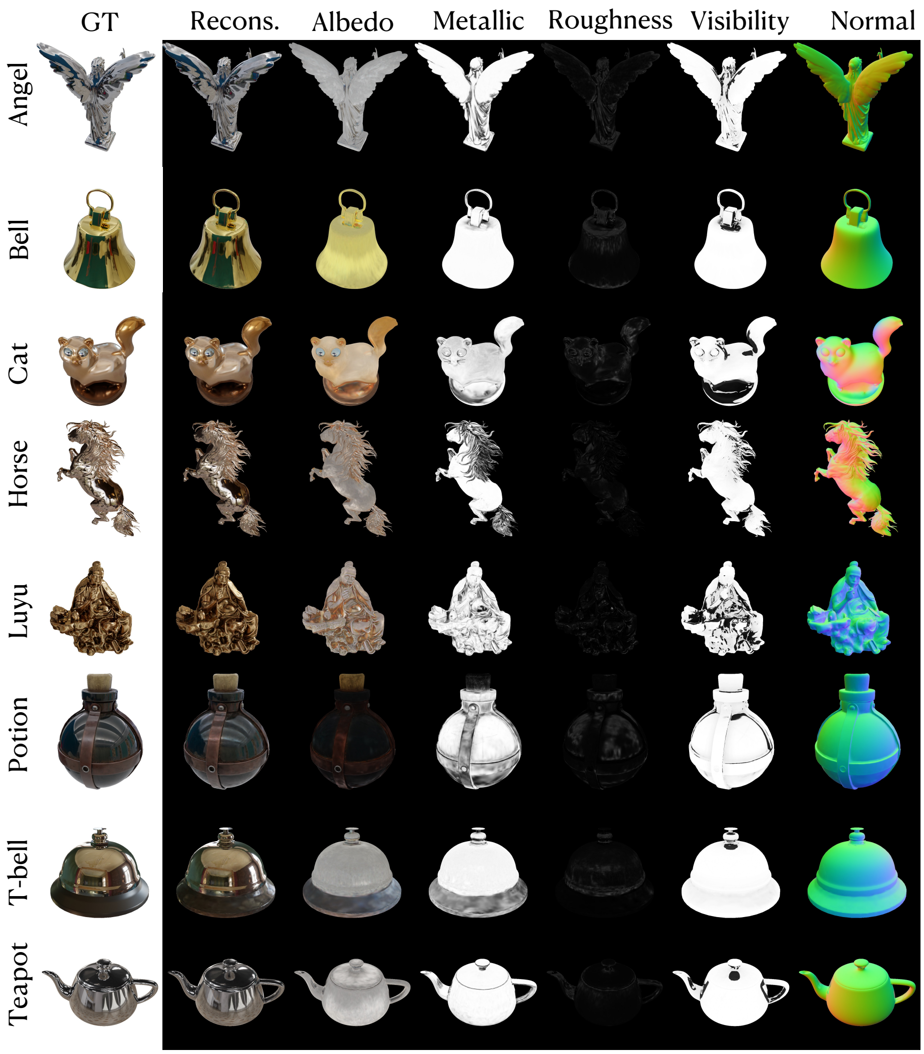

Fig. 1: Parameter disentanglement of DR-GS.

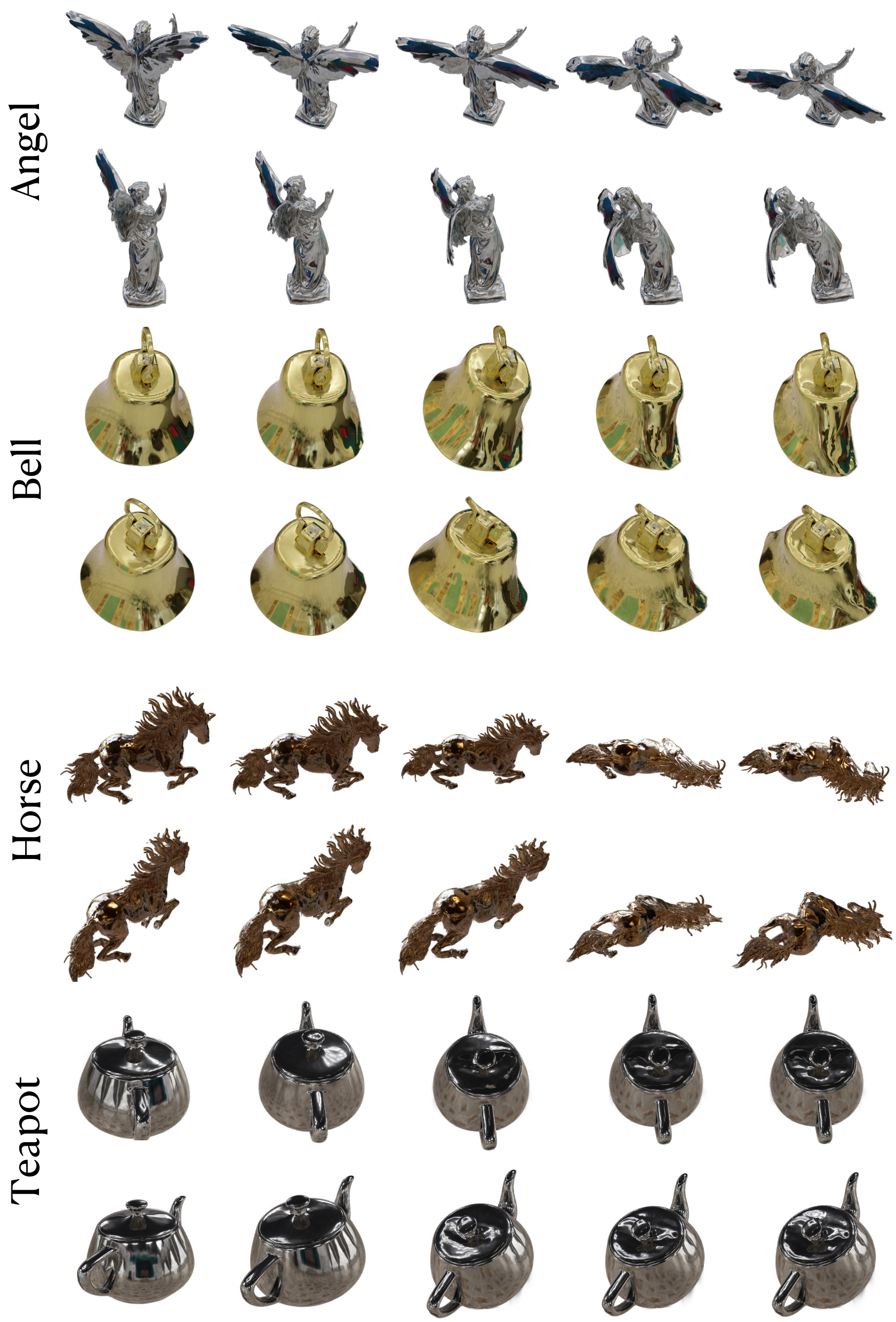

Fig. 2: Physical simulation results for glossy objects.

References

1. Hu, Y., Fang, Y., Ge, Z., Qu, Z., Zhu, Y., Pradhana, A., Jiang, C.: A moving least squares material point method with displacement discontinuity and two-way rigid body coupling. ACM TOG (2018) 2
2. Liu, Y., Wang, P., Lin, C., Long, X., Wang, J., Liu, L., Komura, T., Wang, W.: Nero: Neural geometry and brdf reconstruction of reflective objects from multiview images. In: SIGGRAPH (2023) 4