MGSR: 2D/3D Mutual-boosted Gaussian Splatting for High-fidelity Surface Reconstruction under Various Light Conditions

Supplementary Material

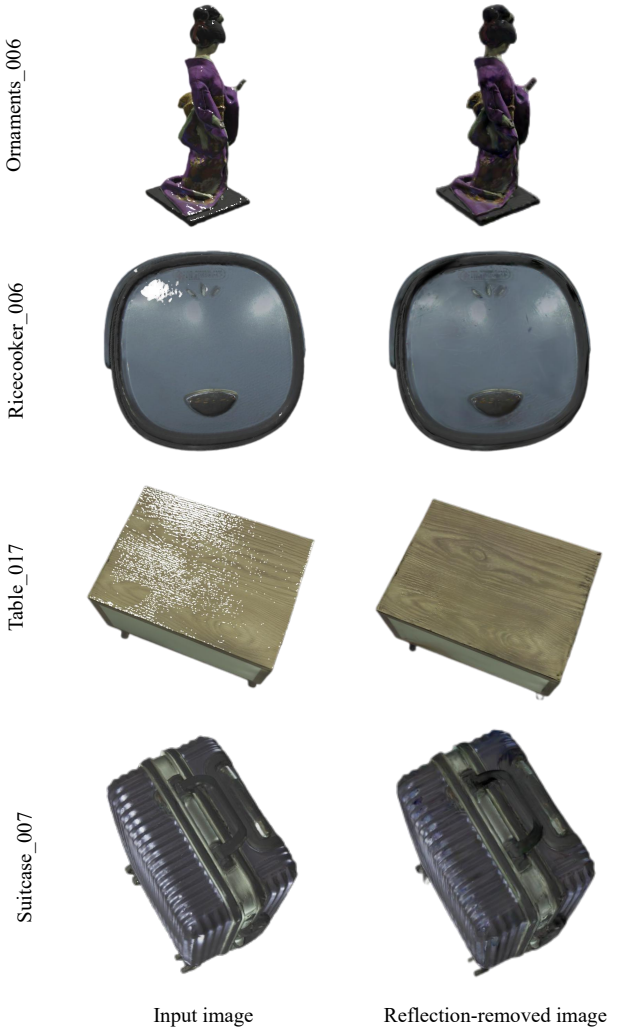


Figure 8. Reflection-removed rendering results of MGSR on OmniObject3D dataset.

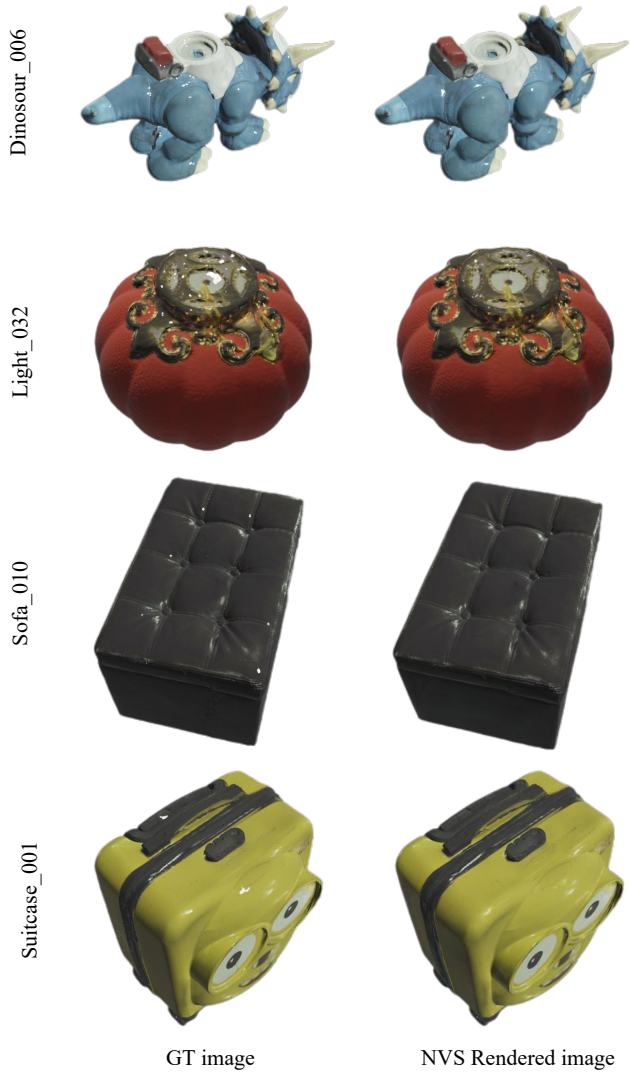


Figure 9. NVS results of MGSR on OmniObject3D dataset.

7. Preliminaries

3D Gaussian Splatting [9] is an explicit scene representation that utilizes numerous 3D anisotropic spheres to model a radiance field. Each of the anisotropic sphere is represented by a 3D Gaussian distribution, as illustrated in Equation 16,

$$G(X) = e^{-\frac{1}{2}\mathcal{M}^\top \Sigma^{-1}\mathcal{M}}, \tag{16}$$

where $\mathcal{M} \in \mathbb{R}^3$ denotes the mean of the Gaussian distribution, Σ stands for the covariance matrix, $\theta \in \mathbb{R}$ represents the opacity of sphere, and $\mathcal{C} \in \mathbb{R}^k$ contains the spherical harmonics (SH) for anisotropic representation where k refers to the degrees of freedom. Considering the inherent physical significance of the covariance matrix, which challenges to update conventionally in back-propagation, the covariance matrix is decomposed into rotation matrix $\mathbf{R}$ and scaling matrix $\mathbf{S}$ by Equation 17,

$$\Sigma = \mathbf{RSS}^\top \mathbf{R}^\top, \tag{17}$$

where $R \in \mathbb{R}^4$ represents quaternions, and $S \in \mathbb{R}^3$ denotes to scaling factors.

View rendering of 3DGS is achieved by projecting 3D Gaussian primitives to 2D splats [34], followed by fast α-blending for rasterization that is computed in parallel on the GPU. Specifically, as shown in Equation 18,

$$\Sigma' = \mathbf{JW}\Sigma\mathbf{J}^\top\mathbf{W}^\top, \tag{18}$$

3D Gaussian sphere is converted into camera coordinates using the world-to-camera transformation matrix W, and it is then projected onto the image plane through an affine transformation J. Note that, to splat a 3D Gaussian to 2D, it is only necessary to skip the third row and column [34]. Subsequently, the color of each pixel is computed with α-blending in depth order, as demonstrated in Equation 19,

$$C = \sum_{k=1}^{K} c_k \alpha_k \prod_{j=1}^{k-1} (1 - \alpha_j), \tag{19}$$

where k denotes to the index of the Gaussian sphere, α_k is the opacity θ multiplied by the density of the projected 2D Gaussian at the pixel location, and c_k represents the view-dependent color.

2D Gaussian Splatting [6] employs flat 2D Gaussian disks for the representation of 3D scenes, aiming to extract surface meshes, which are characterized by central points p_k, two principal tangential vectors t_u and t_v, and a scaling vector $S = (s_u, s_v)$. Similar to 3DGS, each 2D Gaussian comprises the opacity and a view-dependent color represented by SH. A 2D Gaussian is parameterized as Eq. 20,

$$P(u, v) = p_k + s_u t_u u + s_v t_v v = \mathbf{H}(u, v, 1, 1)^\top, \tag{20}$$

where $\mathbf{H}$ is a homogeneous transformation matrix characterizing the geometry of the 2D Gaussian. The normal of a 2D Gaussian primitive is defined as the direction of the steepest change in density. The advantage of the 2DGS representation lies in its 2D, flattened Gaussian primitives being better aligned with the surfaces of objects, leading to more accurate geometry reconstruction.

2DGS utilizes an explicit ray-splat intersection method to splat 2D Gaussian onto the image space, and employs the same rasterization method as 3DGS.

8. Implementation

8.1. Implementation details

MGSR is implemented with PyTorch 2.0.0 and CUDA 11.8. The pseudocode for training MGSR is shown in Algorithm 1 and Algorithm 2.

For the compared methods, all GS-based approaches, except for R3DG, are trained for 30k iterations. R3DG is trained for 40k iterations, and NeuS2 is trained for 15k iterations. On DTU dataset, 2DGS uses the median depths instead of the mean depths, which was proved to be better [6]. For unbounded and in-the-wild scenes, GOF is implemented with the surface extraction technique based on tetrahedra-grids instead of the TSDF [27]. All of the experiments are conducted on NVIDIA A100 GPU(s).

8.2. Parameters

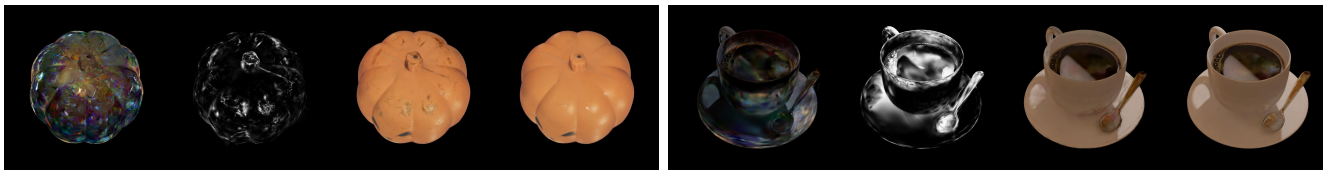
During the training of MGSR, we followed most of the previous GS-based works to set the weights of different loss terms. In the render loss, the weight λ_1 balances the SSIM loss and L1 loss is set to 0.8, while the weight λ_3 of the normal loss is set to 0.05. The weights for all TV losses, λ_2, λ_4, and λ_5, are set to 1. λ_6 of the render loss of mutual-boosted optimization is set to 0.5. Based on our ablation study, to achieve a balance between NVS and SR, the weights are set as follows: w_{2D} is set to 0.5, w_{3D} is set to 0.5, and $w_{\text{depth-mutual}}$ is set to 0.01.

9. Revisiting baselines

In Section 4.2 of the main text, MGSR is compared with three GS-based illumination decomposition methods, three mesh extraction methods, and vanilla 3DGS. Table 5 compares the technical details of the baseline methods, which will be discussed in detail in the following two subsections. "Reflective NVS (Reflective SR)" indicates that the method is capable of NVS (SR) for reflective objects under varying lighting conditions. "Preprocessing Data" refers to the need for data preprocessing before optimization, such as incorporating additional pre-trained methods for depth estimation or performing exposure compensation on the images. "Synthetic Data" denotes that the method supports synthetic datasets. "Additional GT" implies that the method requires extra GT, such as additional GT depth. "Additional Integration" signifies that the method integrates other general and multi-parameterized module beyond Gaussian Splatting, such as BRDF, SDF, etc. "Real-time" indicates that the method preserves the advantage of real-time rendering, while "Time-saving" refers to the total runtime of the method on an object in the OmniObject3D dataset being under 20 minutes (where vanilla 3DGS only takes 7 minutes).

9.1. Illumination decomposition

For illumination decomposition methods, all compared methods [4, 8, 12, 23] integrate BRDF into 3DGS, which leads to slower optimization. GS-IR [12] does not seem to be suitable for rendering scenes with lighting variations and reflective surfaces. R3DG [4] is built upon NEILF [25] and 3DGS, and it is unable to perform real-time rendering. Additionally, R3DG integrates Vis-MVSNet [28] to preprocess the data and perform depth estimation as pseudo GT. Although GShader [8] is dedicated to rendering specular and reflective surfaces, its runtime is a major limitation. Normal-GS [23] is the latest SOTA work, but since it



(a) reflected color (b) reflection map (c) transmitted color (d) rendered (a) reflected color (b) reflection map (c) transmitted color (d) rendered

Figure 10. Visualization of illumination decomposition.

Algorithm 1: The pseudocode for the warm-up of MGSR. • denotes operations in 3DGS branch, while • presents operations on 2DGS branch.

Input: images under various light conditions C_{GT}, the max iterations of auto-stop warm-up $i_{\text{warm-up}}$, 2D Gaussians G_{2D}, 3D Gaussians G_{3D}.
Output: 2D Gaussians G_{2D}, 3D Gaussians G_{3D}.
$L_{\text{2D}} = 0; L_{\text{3D}} = 0$
for i *from* 1 *to* $i_{\textit{warm-up}}$ **do**
  $v = \text{Viewpoint}(i)$ # Select a viewpoint
  $C_{\text{tran}}, C_{\text{ref}}, W = \text{Render\_3D}(G_{\text{3D}}, v)$
  $C_{\text{2D}}, \text{Depth}_{\text{2D}}, \text{Normal} = \text{Render\_2D}(G_{\text{2D}}, v)$
  $C_{\text{3D}} = C_{\text{tran}} + W \times C_{\text{ref}}$
  $L_{\text{render\_3D}} = \text{Render\_loss}(C_{\text{3D}}, C_{\text{GT}})$
  $L_{\text{render\_2D}} = \text{Render\_loss}(C_{\text{2D}}, C_{\text{GT}})$
  $L_{\text{n}} = \text{Normal\_loss}(\text{Normal}, \text{Depth}_{\text{2D}})$
  $L_{\text{trans-TV}} = \text{TV\_loss}(C_{\text{tran}}, C_{\text{GT}})$
  $L_{\text{d-TV}} = \text{TV\_loss}(\text{Depth}_{\text{2D}}, C_{\text{GT}})$
  $L_{\text{n-TV}} = \text{TV\_loss}(\text{Normal}_{\text{2D}}, C_{\text{GT}})$
  $L_{\text{3D}} \mathrel{+}= (L_{\text{render\_3D}} + \lambda_2 L_{\text{trans-TV}})$
  $L_{\text{2D}} \mathrel{+}=$
   $(L_{\text{render\_2D}} + \lambda_3(L_{\text{n}} + \lambda_4 L_{\text{n-TV}}) + \lambda_5 L_{\text{d-TV}})$
  $G_{\text{3D}} \leftarrow \text{Update\_3D\_Gaussian}(G_{\text{3D}}, L_{\text{3D}})$
  $G_{\text{2D}} \leftarrow \text{Update\_2D\_Gaussian}(G_{\text{2D}}, L_{\text{2D}})$
  if $i ==$ *iter_densification* **then**
    $G_{\text{3D}} \leftarrow \text{Densificate\_3D\_Gaussian}(G_{\text{3D}})$
    $G_{\text{2D}} \leftarrow \text{Densificate\_2D\_Gaussian}(G_{\text{2D}})$
  end
  if Convergence(L_{3D}) **then**
    $\text{Stop\_optimization}(G_{\text{3D}})$
  end
  if Convergence(L_{2D}) **then**
    $\text{Stop\_optimization}(G_{\text{2D}})$
  end
end
return $G_{\text{3D}}, G_{\text{2D}}$

Algorithm 2: The pseudocode for the mutual-boosted optimization of MGSR. • denotes operations in 3DGS branch, while • presents operations on 2DGS branch.

Input: images under various light conditions C_{GT}, iterations of mutual-boosted optimization i_{mutual}, warm-up 2D Gaussians G_{2D}, warm-up 3D Gaussians G_{3D}.
Output: 2D Gaussians G_{2D}, 3D Gaussians G_{3D}, total loss L_{total}.
$L_{\text{total}} = 0; L_{\text{2D}} = 0; L_{\text{3D}} = 0$
for i *from* 1 *to* $i_{\textit{mutual}}$ **do**
  $v = \text{Viewpoint}(i)$ # Select a viewpoint
  $C_{\text{tran}}, C_{\text{ref}}, W, \text{Depth}_{\text{3D}} = \text{Render\_3D}(G_{\text{3D}}, v)$
  $C_{\text{2D}}, \text{Depth}_{\text{2D}}, \text{Normal} = \text{Render\_2D}(G_{\text{2D}}, v)$
  $C_{\text{3D}} = C_{\text{tran}} + W \times C_{\text{ref}}$
  $L_{\text{render\_3D}} = \text{Render\_loss}(C_{\text{3D}}, C_{\text{GT}})$
  $L_{\text{render\_2D}} = \text{Render\_loss}(C_{\text{2D}}, C_{\text{GT}})$
  $L_{\text{render\_mutual}} = \text{Render\_loss}(C_{\text{2D}}, C_{\text{tran}})$
  $L_{\text{render\_2D}} = (1 - \lambda_6) L_{\text{render\_mutual}} + \lambda_6 L_{\text{render\_2D}}$
  $L_{\text{n}} = \text{Normal\_loss}(\text{Normal}, \text{Depth}_{\text{2D}})$
  $L_{\text{depth-m}} = \text{L2\_loss}(\text{Depth}_{\text{3D}}, \text{Depth}_{\text{2D}})$
  $L_{\text{d-TV-m}} = \text{TV\_loss}(\text{Depth}_{\text{2D}}, C_{\text{tran}})$
  $L_{\text{n-TV-m}} = \text{TV\_loss}(\text{Normal}_{\text{2D}}, C_{\text{tran}})$
  $L_{\text{2D}} \mathrel{+}=$
   $(L_{\text{render\_2D}} + \lambda_3(L_{\text{n}} + \lambda_4 L_{\text{n-TV-m}}) + \lambda_5 L_{\text{d-TV-m}})$
  $L_{\text{total}} \mathrel{+}= (w_{\text{2D}} L_{\text{2D}} + w_{\text{3D}} L_{\text{3D}} + w_{\text{Mutual}} L_{\text{depth-m}})$
  $G_{\text{3D}} \leftarrow \text{Update\_3D\_Gaussian}(G_{\text{3D}}, L_{\text{total}})$
  $G_{\text{2D}} \leftarrow \text{Update\_2D\_Gaussian}(G_{\text{2D}}, L_{\text{total}})$
  if $i ==$ *iter_densification* **then**
    $G_{\text{3D}} \leftarrow \text{Densificate\_3D\_Gaussian}(G_{\text{3D}})$
    $G_{\text{2D}} \leftarrow \text{Densificate\_2D\_Gaussian}(G_{\text{2D}})$
  end
end
return $G_{\text{3D}}, G_{\text{2D}}, L_{\text{total}}$

requires GT normal maps from synthetic datasets as additional supervision for the optimization of 3DGS, we do not consider it a fair comparison for experiments on reflective surfaces and have not included it in the baseline.

9.2. Surface reconstruction

For surface reconstruction methods, although NeuS2 [22] achieves the fastest speed and the highest CD metric, its surface exhibits visible irregularities. This is why we recommend using Normal Consistency, as the CD metric measures the proximity of two point clouds without considering

Table 5. Detailed technique comparisons on all GS-based baselines. Negative technical metrics are highlighted in red.

Methods	NVS	SR	Reflective NVS	Reflective SR	Preprocessing Data	Synthetic Data	Additional GT	Additional Integration	Real-time	Time-saving
3D-GS	✔	✘	✘	✘	✘	✔	✘	✘	✔	✔
GS-IR	✔	✘	✘	✘	✘	✔	✘	✔	✔	✘
R3DG	✔	✘	✔	✘	✔	✔	✘	✔	✘	✘
GShader	✔	✘	✔	✘	✘	✔	✘	✔	✔	✘
Normal-GS	✔	✘	-	✘	✘	✔	✔	✔	✔	-
2D-GS	✔	✔	✘	✘	✘	✔	✘	✘	✔	✔
GOF	✔	✔	✔	✘	✘	✔	✘	✘	✔	✔
PGSR	✔	✔	-	-	✔	✘	✘	✘	✔	-
GSDF	✔	✔	✘	✘	✘	✔	✘	✔	✘	✘
MGSR (Ours)	✔	✔	✔	✔	✘	✔	✘	✘	✔	✔

surface quality. Due to the lack of consideration for lighting factors, both GOF [27] and 2DGS [6] fail in certain scenes, as shown in Table 6. To ensure a fair and reasonable comparison in the main text, we have excluded the metrics of the failed objects from the average calculation. Although GOF establishes Gaussian Opacity Fields, its optimization time does not increase significantly. Therefore, we also consider it time-saving. PGSR [1] incorporates lighting factors into the training process and uses exposure compensation to adjust the lighting of input data. However, since it is not applicable to synthetic datasets, comparisons cannot be conducted. GSDF [26] is an interesting dual-branch method that introduces SDF into 3DGS. However, since it failed to produce SR results on all objects in Shiny Blender, with many extra faces scattered outside the objects, comparisons with GSDF are not included in the experiments. Additionally, introducing SDF inevitably increases computation time, with GSDF's optimization taking approximately 90 minutes.

10. Revisiting illumination decomposition in MGSR

Decomposing a scene into transmitted and reflected components is inherently an under-constrained problem. Humans can correctly identify reflected virtual images because we have an understanding of real-world geometry. To address this challenge, a geometry-guided illumination decomposition module is introduced in MGSR, where depth information from the 2DGS branch assists the 3DGS branch in identifying reflective regions. To further refine the depth information from the 2DGS branch, we incorporate an iterative refinement process in which the transmitted component of the 3DGS branch supervises the depth estimation during alternating optimization. Moreover, TV loss is applied to the transmitted component to ensure it retains only the low-frequency components of the illumination decomposition. Figure 10 provides an example of illumination decomposition in MGSR.

11. Additional rendering experiments

11.1. Experiments on reflection-removed rendering

In Section 3.2 of the main text, a 3DGS-based illumination decomposition method is introduced. As shown in Figure 8, to validate the illumination decomposition, we present the rendering results of reflection removal on OmniObject3D dataset. MGSR is able to remove reflections and overexposure without supervision effectively.

11.2. Experiments on novel view synthesis

Table 1 and Table 2 in the main text present the NVS results on OmniObject3D dataset. As a supplement, visual results of MGSR (Figure 9) are provided. Considering that reflections and strong overexposure impact the GT image, we aim for the NVS results to retain reflections while removing the overexposure, addressing the view-dependent overexposure caused by camera flashes, which do not belong to the actual scenes. As shown in Figure 11 and Figure 12, previous GS-based methods have encountered challenges of NVS in certain scenes. For detailed results, instance-level NVS results on Omni3DObject dataset are provided in Table 7.

12. Additional SR experiments

The instance-level qualitative and quantitative results of SR on Omni3DObject and DTU datasets are provided. Figure 13-20 are qualitative results on Omni3DObject dataset, Table 6 shows the quantitative on Omni3DObject dataset, and Figure 21-25 are qualitative results on DTU dataset.

As shown in these results, NeuS2 produces inferior surfaces. Under extreme lighting conditions, NeuS2 reconstructs surfaces that deviate significantly from the actual ones (especially on Ricecooker_007, Sofa_011, Table_023, Table_026), and the surface colors are relatively unsatisfactory (especially on Ricecooker_007, Table_009, Table_023). 2DGS tends to generate redundant meshes, particularly beneath objects. In most scenes, a small number of redundant meshes can be observed at the bottom of objects, while in some cases (Light_031, Light_039, Ricecooker_004, Suitcase_003, Suitcase_005, Suitcase_007, Table_019), a large

Figure 11. Challenging scenes in OmniObject3D and the NVS results compared with non-BRDF methods.

redundant plane occurs. To make a fair comparison, scenes with redundant planes were excluded from the quantitative evaluation on 2DGS. GOF results in broken surfaces under various lighting conditions (especially on Light_039, Ornaments_007, Ricecooker_004, Table_017, Table_019, Table_020), and fails to reconstruct Sofa_007. MGSR delivers superior visual performance, but its limitation lies in retaining details in extremely dark scenes.

13. Additional ablations

13.1. Ablations on non-interactive branches

In Section 5 of the main text, by adjusting the weights of the total loss, an ablation study is conducted on the independent learning of the 2DGS and 3DGS branches without interaction. When the weight of the 2DGS branch is set to 0 (Model A), the 3DGS branch learns independently, supporting only NVS. Conversely, when the 3DGS branch weight is set to 0 (Model B), only the 2DGS branch supports SR. In non-interactive optimization, since both branches are optimized simultaneously, the warm-up time remains unchanged, while the total time is shortened by approximately 2 minutes. Table 8 presents the instance-level ablation results, and Model E achieves the best performance.

13.2. Additional ablation on TV loss

As introduced in Section 5 of the main text, TV losses are incorporated as smoothness terms in both branches of MGSR. An additional ablation study is performed to demonstrate the effectiveness of TV loss. Model N denotes the method that does not apply TV losses, while all other parameters and modules are consistent with the best model (Model E). Table 9 presents the instance-level ablation results, and Model E (with TV loss) achieves the best performance.

13.3. Instance-level results of ablations

Moreover, ablation studies are conducted on loss weights (Models A-F), iterations of mutual-boosted optimization (Models G-J), bidirectional BP and auto-stop warm-up strategy (Models K-L) on OmniObject3D dataset. Table 10 presents the instance-level ablation study of loss weights of mutual-boosted optimization involving both branches (Model C-F), with Model E achieving the best performance.

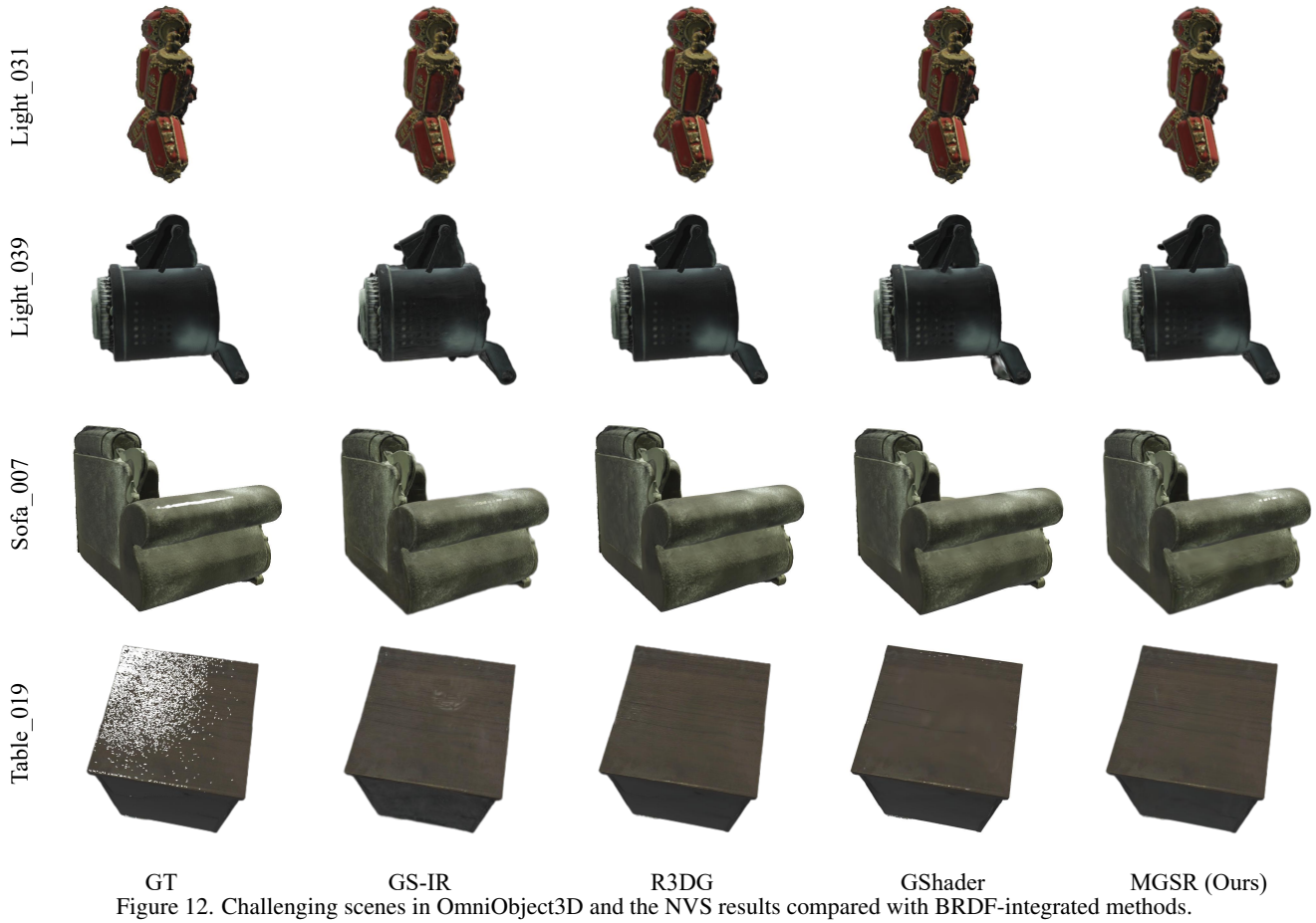


Figure 12. Challenging scenes in OmniObject3D and the NVS results compared with BRDF-integrated methods.

It is found that when the branch responsible for either the NVS or SR task is assigned a greater weight in the total loss, it ultimately achieves better NVS or SR metrics. Specifically, Model C, which assigns a greater weight to the 3DGS branch, achieves higher NVS metrics, while Model D, with a greater weight on the 2DGS branch, delivers better SR metrics. For our task, it is essential to ensure the quality of both SR and NVS, so the weights of the two branches are kept equal. Table 11 presents the instance-level ablation study of iterations of mutual-boosted optimization (Model G-J), with Model I achieving the best performance. Model J demonstrates results similar to Model I but requires more time and computational resources.

Table 12 presents the instance-level ablation study of bidirectional BP and auto-stop warm-up strategy (Model K-M), with Model M achieving the best performance.

Table 6. Instance-level SR results on Omni3DObject dataset. NC is multiplied by 10^2, CD is multiplied by 10^3. Methods marked with * fail on certain objects, which are excluded from the average metric values presented in this table.

Object	NeuS2		2D-GS*		GOF*		MGSR	
	NC↑	CD↓	NC↑	CD↓	NC↑	CD↓	NC↑	CD↓
Dinosaur_006	90.1569	0.0725	81.8403	0.0962	88.6009	0.0650	90.4567	0.0746
Dinosaur_013	95.5248	0.0189	87.1373	0.1064	91.6784	0.0543	94.0746	0.0440
Light_009	96.5897	1.0651	94.0576	1.1344	95.9383	1.8106	97.1616	2.0088
Light_031	88.3717	0.1655	-	-	89.9315	0.0174	88.0956	0.1581
Light_032	95.2516	0.1663	90.3772	0.1789	92.2662	0.3032	90.2715	0.3159
Light_039	83.5021	0.7149	-	-	90.7382	0.2957	90.1561	0.3062
Ornaments_006	88.1233	0.0589	79.7084	0.1391	88.8116	0.0787	89.0818	0.0989
Ornaments_007	74.0552	1.1813	73.0130	0.3775	86.9837	0.3697	87.8159	0.4503
Ricecooker_004	85.9650	0.4691	-	-	91.5693	0.8835	93.2487	0.9029
Ricecooker_006	88.3785	0.8975	89.3856	1.0538	93.1174	1.9190	94.3153	2.2044
Ricecooker_007	91.9614	0.3791	-	-	93.8866	1.1480	94.5260	1.3389
Ricecooker_008	90.8250	0.1334	85.8226	0.3651	92.4282	0.7426	92.9716	0.8535
Ricecooker_009	91.9463	1.4914	92.3986	1.0975	92.6477	1.7290	93.6561	1.9475
Sofa_006	87.5499	0.3712	86.5336	1.0148	88.8350	1.2254	89.4616	1.2888
Sofa_007	84.3413	0.8330	83.3484	1.1001	-	-	89.9815	2.0386
Sofa_009	85.3473	0.4424	84.5947	1.5745	85.8566	1.7763	87.1883	1.9120
Sofa_010	90.3916	0.4457	89.7465	1.6659	91.4509	1.5579	90.6918	1.7355
Sofa_011	90.6936	0.2745	85.6295	0.2766	89.8381	0.3298	89.7134	0.4064
Suitcase_001	90.8942	0.1191	86.9193	0.6705	88.7701	0.8664	89.7266	0.9542
Suitcase_003	85.9519	0.1217	-	-	88.8916	0.2161	88.7826	0.1370
Suitcase_004	77.5440	0.3384	-	-	83.0572	0.3942	81.2096	0.3125
Suitcase_005	82.8952	0.1780	-	-	87.7643	0.2180	86.3998	0.1478
Suitcase_006	88.4846	0.2626	86.2197	0.7476	92.7503	0.5140	92.1355	0.5843
Suitcase_007	79.7047	0.4638	-	-	86.2734	0.3973	85.5167	0.2656
Table_009	90.1395	0.8906	85.0342	0.9401	88.3946	1.4133	89.4704	1.4223
Table_017	95.6205	0.1915	88.6482	2.0993	91.1475	2.3482	92.1693	2.2957
Table_019	87.3016	1.3056	-	-	86.8282	2.3097	90.7717	1.5752
Table_020	92.9152	0.1895	89.0018	0.9170	87.1417	1.0857	92.3414	0.9811
Table_023	92.0970	0.1956	88.4267	0.0345	94.0275	0.1637	93.8091	0.1192
Table_026	83.7329	0.8739	85.2499	0.2125	92.2478	0.1719	92.8703	0.1750
Mean	88.2085	**0.4770**	86.3378*	0.7525*	90.0646*	0.8415*	**90.6024**	0.9018

Table 7. Instance-level NVS results on Omni3DObject dataset.

Object	3D-GS		GS-IR		R3DG		GShader		2D-GS		GOF		MGSR	
	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑
Dinosaur_006	0.9927	36.4287	0.9846	35.6511	0.9923	37.6052	0.9933	38.3139	0.9937	38.4870	0.9934	38.6770	0.9932	38.6247
Dinosaur_013	0.9916	33.6264	0.9874	35.2058	0.9923	35.9954	0.9934	36.0291	0.9936	35.9326	0.9931	36.4862	0.9914	36.1045
Light_009	0.9877	31.7295	0.9778	32.6627	0.9884	34.4062	0.9872	31.4642	0.9879	31.4275	0.9890	33.8423	0.9858	33.3903
Light_031	0.9934	34.9704	0.9872	37.6736	0.9926	39.2882	0.9900	34.3910	0.9747	27.4650	0.9937	40.6299	0.9920	40.2280
Light_032	0.9864	32.8026	0.9721	33.3569	0.9872	34.7429	0.9843	32.4283	0.9852	32.5503	0.9860	34.9471	0.9825	34.6889
Light_039	0.9819	26.9721	0.9801	34.9665	0.9905	38.7602	0.9802	26.7486	0.9664	24.1437	0.9906	37.8889	0.9895	38.0387
Ornaments_006	0.9926	34.7640	0.9881	35.3220	0.9931	36.4150	0.9935	36.3201	0.9937	36.4055	0.9931	36.5107	0.9928	36.3158
Ornaments_007	0.9884	32.6647	0.9774	32.6472	0.9848	32.4016	0.9892	33.8028	0.9899	33.8669	0.9893	34.0256	0.9886	33.9089
Ricecooker_004	0.9884	33.6033	0.9776	33.5642	0.9879	34.5135	0.9862	33.6316	0.9859	33.3472	0.9878	35.5263	0.9838	35.1740
Ricecooker_006	0.9797	32.4254	0.9631	32.5921	0.9829	36.1250	0.9766	32.2463	0.9777	32.1262	0.9806	35.0519	0.9746	34.7367
Ricecooker_007	0.9893	34.2847	0.9813	34.5967	0.9875	36.2659	0.9859	32.1802	0.9875	33.6553	0.9912	37.7394	0.9884	37.2851
Ricecooker_008	0.9861	33.9829	0.9812	33.7846	0.9900	36.2274	0.9870	34.5241	0.9883	34.4347	0.9900	36.3168	0.9878	36.4294
Ricecooker_009	0.9874	34.3698	0.9424	31.6037	0.9886	37.0272	0.9848	34.2798	0.9853	34.3124	0.9896	38.1752	0.9862	37.9426
Sofa_006	0.9889	38.0209	0.9825	37.9212	0.9892	40.1741	0.9852	37.8428	0.9866	37.6375	0.9915	40.4490	0.9860	40.0727
Sofa_007	0.9653	32.2157	0.9704	33.8311	0.9783	36.0560	0.9695	35.0136	0.9718	35.1502	0.9852	36.6120	0.9696	35.9044
Sofa_009	0.9849	34.7215	0.9759	33.6432	0.9836	37.2150	0.9817	34.4730	0.9826	34.5866	0.9874	35.4817	0.9821	35.3156
Sofa_010	0.9892	34.7421	0.9716	34.6905	0.9850	37.0198	0.9871	34.7184	0.9873	34.4099	0.9885	37.7966	0.9857	37.3777
Sofa_011	0.9943	37.1184	0.9861	37.9595	0.9932	40.9331	0.9922	36.0972	0.9942	37.4036	0.9950	41.9851	0.9940	41.6715
Suitcase_001	0.9858	34.5283	0.9722	34.8769	0.9861	37.4308	0.9823	34.7406	0.9824	34.6655	0.9848	37.1346	0.9784	36.5393
Suitcase_003	0.9832	29.3838	0.9760	35.7282	0.9897	38.8572	0.9829	29.6674	0.9846	30.2111	0.9891	39.0383	0.9863	38.5199
Suitcase_004	0.9855	29.3198	0.9714	36.9130	0.9869	39.1367	0.9770	27.7403	0.9820	27.8992	0.9903	40.8235	0.9836	39.5490
Suitcase_005	0.9932	34.6974	0.9797	36.4900	0.9915	39.4079	0.9889	33.3183	0.9717	30.0767	0.9936	40.9601	0.9906	40.3685
Suitcase_006	0.9849	32.1548	0.9847	36.7334	0.9912	37.7130	0.9771	29.4960	0.9821	29.3216	0.9922	39.0869	0.9904	38.5725
Suitcase_007	0.9899	31.2892	0.9738	35.5182	0.9889	37.9873	0.9839	29.0931	0.9877	28.9368	0.9929	40.4452	0.9903	39.9557
Table_009	0.9875	34.9191	0.9768	37.0736	0.9881	38.6299	0.9868	38.8087	0.9878	40.1807	0.9895	40.8316	0.9870	40.4433
Table_017	0.9662	34.1205	0.9542	34.0561	0.9852	39.8399	0.9659	37.7021	0.9665	38.6527	0.9679	38.8205	0.9663	38.3512
Table_019	0.9715	33.5099	0.9568	34.8068	0.9760	35.9826	0.9710	37.4496	0.9708	36.2160	0.9738	38.5274	0.9708	38.2598
Table_020	0.9803	34.3233	0.9582	34.7215	0.9779	35.9871	0.9771	37.3925	0.9779	38.1759	0.9818	39.3978	0.9771	38.8922
Table_023	0.9964	38.6613	0.9905	42.1495	0.9924	43.6015	0.9948	39.0959	0.9972	45.5292	0.9978	46.2803	0.9921	42.5329
Table_026	0.9843	32.7922	0.9738	33.9553	0.9811	35.3654	0.9843	35.2430	0.9854	35.5367	0.9859	35.8765	0.9840	35.5904
Mean	0.9859	33.6381	0.9752	35.1565	0.9874	37.3704	0.9840	34.1418	0.9836	34.0915	**0.9885**	**38.1788**	0.9850	37.6928

Table 8. Ablation studies of non-interactive branches on OmniObject3D dataset. NC is multiplied by 10^2, CD is multiplied by 10^3. Method marked with * fails on certain objects, which are excluded from the average metric values presented in this table.

Object	Model A*		Model B		**Model E**			
	SSIM↑	PSNR↑	NC↑	CD↓	SSIM↑	PSNR↑	NC↑	CD↓
Dinosaur_006	0.9913	38.1125	90.5925	0.0748	0.9932	38.6247	90.4567	0.0746
Dinosaur_013	0.9859	34.6946	92.1297	0.0604	0.9914	36.1045	94.0746	0.0440
Light_009	0.9770	31.9619	96.5244	2.0609	0.9858	33.3903	97.1616	2.0088
Light_031	0.9750	35.8377	88.4982	0.1489	0.9920	40.2280	88.0956	0.1581
Light_032	0.9605	31.8998	90.1979	0.2893	0.9825	34.6889	90.2715	0.3159
Light_039	0.9805	35.2980	89.9830	0.2753	0.9895	38.0387	90.1561	0.3062
Ornaments_006	0.9911	35.9259	88.9423	0.1004	0.9928	36.3158	89.0818	0.0989
Ornaments_007	0.9866	33.6558	87.7405	0.4575	0.9886	33.9089	87.8159	0.4503
Ricecooker_004	0.9732	32.3026	93.8493	0.9389	0.9838	35.1740	93.2487	0.9029
Ricecooker_006	0.9647	33.1932	94.1782	2.1032	0.9746	34.7367	94.3153	2.2044
Ricecooker_007	0.9841	36.2126	94.1524	1.2862	0.9884	37.2851	94.5260	1.3389
Ricecooker_008	0.9832	34.5028	92.6104	0.9185	0.9878	36.4294	92.9716	0.8535
Ricecooker_009	0.9794	36.3032	93.2737	2.0498	0.9862	37.9426	93.6561	1.9475
Sofa_006	0.9802	38.4267	88.4430	1.3269	0.9860	40.0727	89.4616	1.2888
Sofa_007	0.9566	34.5499	90.0589	2.1387	0.9696	35.9044	89.9815	2.0386
Sofa_009	0.9748	33.8555	87.7013	1.7785	0.9821	35.3156	87.1883	1.9120
Sofa_010	0.9739	34.5919	90.1309	1.9343	0.9857	37.3777	90.6918	1.7355
Sofa_011	0.9801	35.5484	88.6660	0.4271	0.9940	41.6715	89.7134	0.4064
Suitcase_001	0.9566	33.1515	88.5754	1.0165	0.9784	36.5393	89.7266	0.9542
Suitcase_003	0.9769	36.0147	88.4477	0.1363	0.9863	38.5199	88.7826	0.1370
Suitcase_004	0.9605	36.1530	81.2667	0.3178	0.9836	39.5490	81.2096	0.3125
Suitcase_005	0.9851	38.3858	86.2360	0.1334	0.9906	40.3685	86.3998	0.1478
Suitcase_006	0.9873	37.6382	91.5944	0.6160	0.9904	38.5725	92.1355	0.5843
Suitcase_007	0.9793	37.3233	84.2799	0.2765	0.9903	39.9557	85.5167	0.2656
Table_009	-	-	87.2876	1.3349	0.9870	40.4433	89.4704	1.4223
Table_017	0.9503	31.1000	91.6864	2.3485	0.9663	38.3512	92.1693	2.2957
Table_019	-	-	89.9166	1.6810	0.9708	38.2598	90.7717	1.5752
Table_020	0.9681	36.0674	92.0774	0.9877	0.9771	38.8922	92.3414	0.9811
Table_023	-	-	93.5767	0.1001	0.9921	42.5329	93.8091	0.1192
Table_026	-	-	92.1997	0.1542	0.9840	35.5904	92.8703	0.1750
Mean	0.9755	35.1041	90.1606	0.9157	**0.9850**	**37.6928**	**90.6024**	**0.9018**

Table 9. Ablation studies of the TV loss on OmniObject3D dataset. NC is multiplied by 10^2, CD is multiplied by 10^3. Method marked with * fails on certain objects, which are excluded from the average metric values presented in this table.

Object	Model N*				**Model E**			
	SSIM↑	PSNR↑	NC↑	CD↓	SSIM↑	PSNR↑	NC↑	CD↓
Dinosaur_006	0.9933	38.6872	90.4634	0.0746	0.9932	38.6247	90.4567	0.0746
Dinosaur_013	0.9920	36.2217	93.8064	0.0525	0.9914	36.1045	94.0746	0.0440
Light_009	0.9866	33.3829	96.9765	1.9267	0.9858	33.3903	97.1616	2.0088
Light_031	0.9915	40.0097	87.7747	0.1397	0.9920	40.2280	88.0956	0.1581
Light_032	0.9809	34.6044	90.4378	0.3193	0.9825	34.6889	90.2715	0.3159
Light_039	0.9881	36.7589	86.2017	0.4096	0.9895	38.0387	90.1561	0.3062
Ornaments_006	0.9929	36.3431	89.2538	0.0937	0.9928	36.3158	89.0818	0.0989
Ornaments_007	0.9886	33.9144	87.6355	0.4423	0.9886	33.9089	87.8159	0.4503
Ricecooker_004	0.9833	34.9910	93.4426	0.9013	0.9838	35.1740	93.2487	0.9029
Ricecooker_006	0.9747	34.7858	93.9987	2.2837	0.9746	34.7367	94.3153	2.2044
Ricecooker_007	0.9887	37.2113	94.2233	1.2972	0.9884	37.2851	94.5260	1.3389
Ricecooker_008	0.9880	36.5234	93.1555	0.8789	0.9878	36.4294	92.9716	0.8535
Ricecooker_009	0.9862	37.7828	93.6198	2.0441	0.9862	37.9426	93.6561	1.9475
Sofa_006	0.9858	40.0083	89.3410	1.2751	0.9860	40.0727	89.4616	1.2888
Sofa_007	0.9717	36.0763	90.0827	2.1480	0.9696	35.9044	89.9815	2.0386
Sofa_009	0.9830	35.4634	87.2521	1.7226	0.9821	35.3156	87.1883	1.9120
Sofa_010	0.9859	37.2347	90.5786	1.7111	0.9857	37.3777	90.6918	1.7355
Sofa_011	0.9938	41.5324	89.2901	0.4105	0.9940	41.6715	89.7134	0.4064
Suitcase_001	0.9783	36.4412	90.3672	1.0035	0.9784	36.5393	89.7266	0.9542
Suitcase_003	0.9864	38.7144	88.8497	0.1412	0.9863	38.5199	88.7826	0.1370
Suitcase_004	0.9825	39.2706	80.1943	0.3659	0.9836	39.5490	81.2096	0.3125
Suitcase_005	0.9918	40.6811	86.4862	0.1313	0.9906	40.3685	86.3998	0.1478
Suitcase_006	0.9906	38.6354	91.9379	0.5754	0.9904	38.5725	92.1355	0.5843
Suitcase_007	0.9902	40.0736	84.5554	0.2673	0.9903	39.9557	85.5167	0.2656
Table_009	0.9870	40.5000	89.4792	1.4272	0.9870	40.4433	89.4704	1.4223
Table_017	0.9659	38.0879	92.3002	2.4423	0.9663	38.3512	92.1693	2.2957
Table_019	-	-	90.3206	1.6275	0.9708	38.2598	90.7717	1.5752
Table_020	0.9756	38.7866	92.4123	0.9851	0.9771	38.8922	92.3414	0.9811
Table_023	-	-	93.6877	0.1217	0.9921	42.5329	93.8091	0.1192
Table_026	0.9798	34.9358	92.9960	0.1774	0.9840	35.5904	92.8703	0.1750
Mean	**0.9851**	37.4164	90.3707	0.9132	0.9850	**37.6928**	**90.6024**	**0.9018**

Table 10. Ablation studies of loss weights on OmniObject3D dataset. NC is multiplied by 10^2, CD is multiplied by 10^3.

Object	Model C				Model D				**Model E**				Model F			
	SSIM↑	PSNR↑	NC↑	CD↓	SSIM↑	PSNR↑	NC↑	CD↓	SSIM↑	PSNR↑	NC↑	CD↓	SSIM↑	PSNR↑	NC↑	CD↓
Dinosaur_006	0.9935	38.7076	90.5023	0.0784	0.9924	38.4711	90.2034	0.0785	0.9932	38.6247	90.4567	0.0746	0.9903	37.7020	90.6364	0.0749
Dinosaur_013	0.9916	36.1224	94.1225	0.0525	0.9906	35.9385	94.0279	0.0506	0.9914	36.1045	94.0746	0.0440	0.9900	35.7224	94.0528	0.0526
Light_009	0.9869	33.4895	97.1528	2.0368	0.9827	32.9686	97.1905	1.9354	0.9858	33.3903	97.1616	2.0088	0.9789	32.1022	97.0563	1.9512
Light_031	0.9932	40.6310	88.0013	0.1543	0.9883	39.0731	88.0980	0.1428	0.9920	40.2280	88.0956	0.1581	0.9878	38.6098	87.3143	0.1352
Light_032	0.9835	34.7910	90.3095	0.3105	0.9786	34.3320	90.4992	0.3082	0.9825	34.6889	90.2715	0.3159	0.9783	34.1355	90.3524	0.2738
Light_039	0.9901	38.2049	90.0410	0.2809	0.9879	37.6014	90.0103	0.2777	0.9895	38.0387	90.1561	0.3062	0.9853	36.8469	90.1714	0.3152
Ornaments_006	0.9930	36.3896	88.7504	0.1007	0.9921	36.2037	89.0907	0.0898	0.9928	36.3158	89.0818	0.0989	0.9923	36.2403	88.9554	0.1001
Ornaments_007	0.9887	33.8634	87.6954	0.4502	0.9879	33.8269	87.7090	0.4407	0.9886	33.9089	87.8159	0.4503	0.9876	33.7858	87.3093	0.4595
Ricecooker_004	0.9845	35.2523	93.5480	0.8993	0.9811	34.4836	93.4950	0.9647	0.9838	35.1740	93.2487	0.9029	0.9743	33.2737	93.6942	0.9364
Ricecooker_006	0.9761	34.8970	94.2851	2.1977	0.9720	34.3524	94.1818	2.2963	0.9746	34.7367	94.3153	2.2044	0.9686	33.4684	94.3494	2.2106
Ricecooker_007	0.9893	37.5324	94.0571	1.2318	0.9863	36.7675	94.6029	1.2492	0.9884	37.2851	94.5260	1.3389	0.9795	34.7943	94.2734	1.3091
Ricecooker_008	0.9883	36.5371	92.9274	0.9158	0.9867	36.0597	92.7839	0.9368	0.9878	36.4294	92.9716	0.8535	0.9826	34.7000	93.2196	0.8852
Ricecooker_009	0.9872	37.9975	93.5243	2.0460	0.9835	37.2980	93.6399	1.9359	0.9862	37.9426	93.6561	1.9475	0.9764	34.6591	93.7341	2.0912
Sofa_006	0.9874	39.9959	89.5067	1.2683	0.9838	39.7737	89.4717	1.2743	0.9860	40.0727	89.4616	1.2888	0.9831	38.9391	89.6677	1.3087
Sofa_007	0.9755	36.3484	90.0072	2.1749	0.9627	35.3764	90.1160	2.1653	0.9696	35.9044	89.9815	2.0386	0.9510	34.3506	89.9760	1.9597
Sofa_009	0.9839	35.5753	87.5711	1.8709	0.9797	34.9001	87.3417	1.8188	0.9821	35.3156	87.1883	1.9120	0.9764	34.1961	87.2809	1.8166
Sofa_010	0.9866	37.6033	90.9955	1.6919	0.9836	36.5962	91.0662	1.8383	0.9857	37.3777	90.6918	1.7355	0.9809	35.5376	91.0651	1.7339
Sofa_011	0.9938	41.6951	89.3621	0.4246	0.9936	41.4389	89.4910	0.4082	0.9940	41.6715	89.7134	0.4064	0.9920	40.2625	89.5397	0.4473
Suitcase_001	0.9803	36.6937	90.1120	1.0062	0.9719	35.5736	90.0518	0.9975	0.9784	36.5393	89.7266	0.9542	0.9708	35.3707	90.0329	0.8898
Suitcase_003	0.9865	38.6728	88.7559	0.1284	0.9843	38.0164	88.7584	0.1521	0.9863	38.5199	88.7826	0.1370	0.9837	37.7332	88.7584	0.1419
Suitcase_004	0.9862	39.9987	81.1906	0.3063	0.9783	38.6967	81.3494	0.3175	0.9836	39.5490	81.2096	0.3125	0.9789	38.3876	81.3336	0.3402
Suitcase_005	0.9914	40.6171	86.2041	0.1392	0.9893	39.9424	87.1172	0.1351	0.9906	40.3685	86.3998	0.1478	0.9890	39.6695	87.1191	0.1328
Suitcase_006	0.9913	38.9944	91.9900	0.6081	0.9882	37.7774	91.7508	0.5987	0.9904	38.5725	92.1355	0.5843	0.9852	36.6509	91.8762	0.6232
Suitcase_007	0.9909	40.0387	84.5505	0.2810	0.9878	39.2398	85.3193	0.2819	0.9903	39.9557	85.5167	0.2656	0.9877	39.0710	85.2697	0.2838
Table_009	0.9876	40.5444	89.3742	1.4930	0.9848	39.9106	89.3460	1.5431	0.9870	40.4433	89.4704	1.4223	0.9823	38.2714	89.6611	1.5277
Table_017	0.9669	38.7009	91.6541	2.5077	0.9645	37.4933	92.0956	2.3481	0.9663	38.3512	92.1693	2.2957	0.9615	35.9631	92.3982	2.4673
Table_019	0.9718	38.5869	90.8241	1.6497	0.9692	37.7830	90.7415	1.6311	0.9708	38.2598	90.7717	1.5752	0.9654	36.4868	90.1328	1.6119
Table_020	0.9782	38.9681	92.5009	0.9587	0.9753	38.3294	92.5189	1.0181	0.9771	38.8922	92.3414	0.9811	0.9745	37.5417	92.2857	0.9645
Table_023	0.9938	43.3632	93.8248	0.1180	0.9853	36.1571	93.7027	0.1208	0.9921	42.5329	93.8091	0.1192	0.9933	43.0188	93.7005	0.1229
Table_026	0.9842	35.6100	92.8967	0.1748	0.9835	35.4618	92.8540	0.1797	0.9840	35.5904	92.8703	0.1750	0.9822	35.2354	92.7446	0.1797
Mean	**0.9861**	**37.8808**	90.5413	0.9186	0.9825	36.9948	**90.6208**	0.9178	0.9850	37.6928	90.6024	**0.9018**	0.9803	36.4242	90.5987	0.9116

Table 11. Ablation studies of iterations of mutual-boosted optimization on OmniObject3D dataset. NC is multiplied by 10^2, CD is multiplied by 10^3.

Object	Model G				Model H				**Model I**				Model J			
	SSIM↑	PSNR↑	NC↑	CD↓	SSIM↑	PSNR↑	NC↑	CD↓	SSIM↑	PSNR↑	NC↑	CD↓	SSIM↑	PSNR↑	NC↑	CD↓
Dinosaur_006	0.9919	38.0976	90.0934	0.0769	0.9928	38.5022	90.0895	0.0736	0.9932	38.6247	90.4567	0.0746	0.9931	38.6431	90.4450	0.0776
Dinosaur_013	0.9903	35.7568	93.5658	0.0529	0.9914	36.0772	93.9775	0.0421	0.9914	36.1045	94.0746	0.0440	0.9915	36.1522	93.9302	0.0497
Light_009	0.9838	32.8212	97.1579	1.9856	0.9855	33.2162	97.0609	2.0314	0.9858	33.3903	97.1616	2.0088	0.9858	33.3579	97.1217	2.0307
Light_031	0.9882	38.6820	88.1789	0.1404	0.9908	39.6393	88.4260	0.1427	0.9920	40.2280	88.0956	0.1581	0.9920	40.3040	88.0958	0.1356
Light_032	0.9777	33.9552	90.1784	0.3003	0.9814	34.4455	90.2375	0.2861	0.9825	34.6889	90.2715	0.3159	0.9826	34.7544	90.3048	0.3026
Light_039	0.9867	37.0563	89.0423	0.2848	0.9886	37.6654	89.8071	0.2799	0.9895	38.0387	90.1561	0.3062	0.9892	38.0389	89.0618	0.3165
Ornaments_006	0.9918	36.0027	89.0806	0.0937	0.9926	36.2039	88.9162	0.0977	0.9928	36.3158	89.0818	0.0989	0.9928	36.2769	89.1383	0.0950
Ornaments_007	0.9880	33.7663	87.6941	0.4568	0.9886	33.8786	87.5242	0.4428	0.9886	33.9089	87.8159	0.4503	0.9886	33.9240	87.7423	0.4519
Ricecooker_004	0.9785	34.0600	93.6265	0.8532	0.9821	34.7303	93.2170	0.9331	0.9838	35.1740	93.2487	0.9029	0.9832	34.9293	93.5894	0.9690
Ricecooker_006	0.9719	33.9802	93.9333	2.0943	0.9741	34.3491	93.9516	2.1742	0.9746	34.7367	94.3153	2.2044	0.9744	34.7051	93.9445	2.1523
Ricecooker_007	0.9847	35.7660	94.2507	1.1895	0.9875	36.6956	94.4850	1.3280	0.9884	37.2851	94.5260	1.3389	0.9884	37.3517	94.3819	1.3020
Ricecooker_008	0.9857	35.4079	92.5156	0.8517	0.9872	36.1642	92.9819	0.9104	0.9878	36.4294	92.9716	0.8535	0.9877	36.3816	92.8991	0.9216
Ricecooker_009	0.9809	35.9723	93.1824	1.9736	0.9846	37.2459	93.3837	2.0341	0.9862	37.9426	93.6561	1.9475	0.9861	37.9504	93.6212	1.9253
Sofa_006	0.9832	39.1484	89.1391	1.3010	0.9851	39.7106	89.4871	1.2953	0.9860	40.0727	89.4616	1.2888	0.9861	40.0419	89.5790	1.3121
Sofa_007	0.9634	34.8655	89.7890	1.8807	0.9680	35.6193	89.9762	2.1302	0.9696	35.9044	89.9815	2.0386	0.9700	35.9934	90.2142	2.1299
Sofa_009	0.9795	34.4533	87.3781	1.8196	0.9809	34.8214	87.0543	1.7670	0.9821	35.3156	87.1883	1.9120	0.9816	35.3030	87.2689	1.7620
Sofa_010	0.9831	36.2458	90.8361	1.7007	0.9848	36.7963	90.9125	1.8497	0.9857	37.3777	90.6918	1.7355	0.9855	37.3082	90.9672	1.7997
Sofa_011	0.9927	40.5870	89.3115	0.4104	0.9936	41.2728	89.2118	0.4072	0.9940	41.6715	89.7134	0.4064	0.9939	41.6524	89.2488	0.4331
Suitcase_001	0.9700	35.1519	90.5080	0.9917	0.9757	36.0293	89.8770	1.0160	0.9784	36.5393	89.7266	0.9542	0.9776	36.3779	90.2653	1.0378
Suitcase_003	0.9832	37.4646	88.5807	0.1284	0.9855	38.2793	88.7146	0.1332	0.9863	38.5199	88.7826	0.1370	0.9863	38.6851	88.8046	0.1523
Suitcase_004	0.9774	38.2992	81.2235	0.3312	0.9835	39.4158	81.6032	0.2922	0.9836	39.5490	81.2096	0.3125	0.9848	39.8903	81.6224	0.3220
Suitcase_005	0.9875	38.9638	86.4377	0.1326	0.9901	40.0463	86.3628	0.1326	0.9906	40.3685	86.3998	0.1478	0.9908	40.4081	87.1310	0.1341
Suitcase_006	0.9879	37.5567	91.8885	0.6287	0.9893	37.9435	91.6888	0.6201	0.9904	38.5725	92.1355	0.5843	0.9899	38.4910	91.5118	0.6240
Suitcase_007	0.9862	38.4473	84.7678	0.2618	0.9889	39.3398	85.1933	0.2513	0.9903	39.9557	85.5167	0.2656	0.9900	39.8852	85.3722	0.2555
Table_009	0.9850	39.6287	89.2592	1.3782	0.9865	40.2311	88.8564	1.4217	0.9870	40.4433	89.4704	1.4223	0.9868	40.4354	89.6436	1.4370
Table_017	0.9646	37.4566	91.7903	2.3807	0.9661	38.2929	92.1498	2.2562	0.9663	38.3512	92.1693	2.2957	0.9659	38.2507	92.2816	2.2413
Table_019	0.9690	37.5148	90.3345	1.7056	0.9706	38.1829	90.4667	1.6635	0.9708	38.2598	90.7717	1.5752	0.9707	38.2408	90.7012	1.4730
Table_020	0.9738	37.8008	92.3906	0.9540	0.9758	38.2749	92.6057	0.9873	0.9771	38.8922	92.3414	0.9811	0.9769	38.8380	92.3607	1.0167
Table_023	0.9919	42.5166	93.3231	0.1167	0.9922	42.6253	93.8670	0.1240	0.9921	42.5329	93.8091	0.1192	0.9927	43.1277	93.7175	0.1195
Table_026	0.9826	35.2435	92.5842	0.1654	0.9837	35.4517	92.8908	0.1667	0.9840	35.5904	92.8703	0.1750	0.9838	35.5457	93.1755	0.1816
Mean	0.9820	36.7556	90.4014	0.8880	0.9842	37.3716	90.4992	0.9097	**0.9850**	37.6928	90.6024	**0.9018**	**0.9850**	**37.7081**	**90.6047**	0.9054

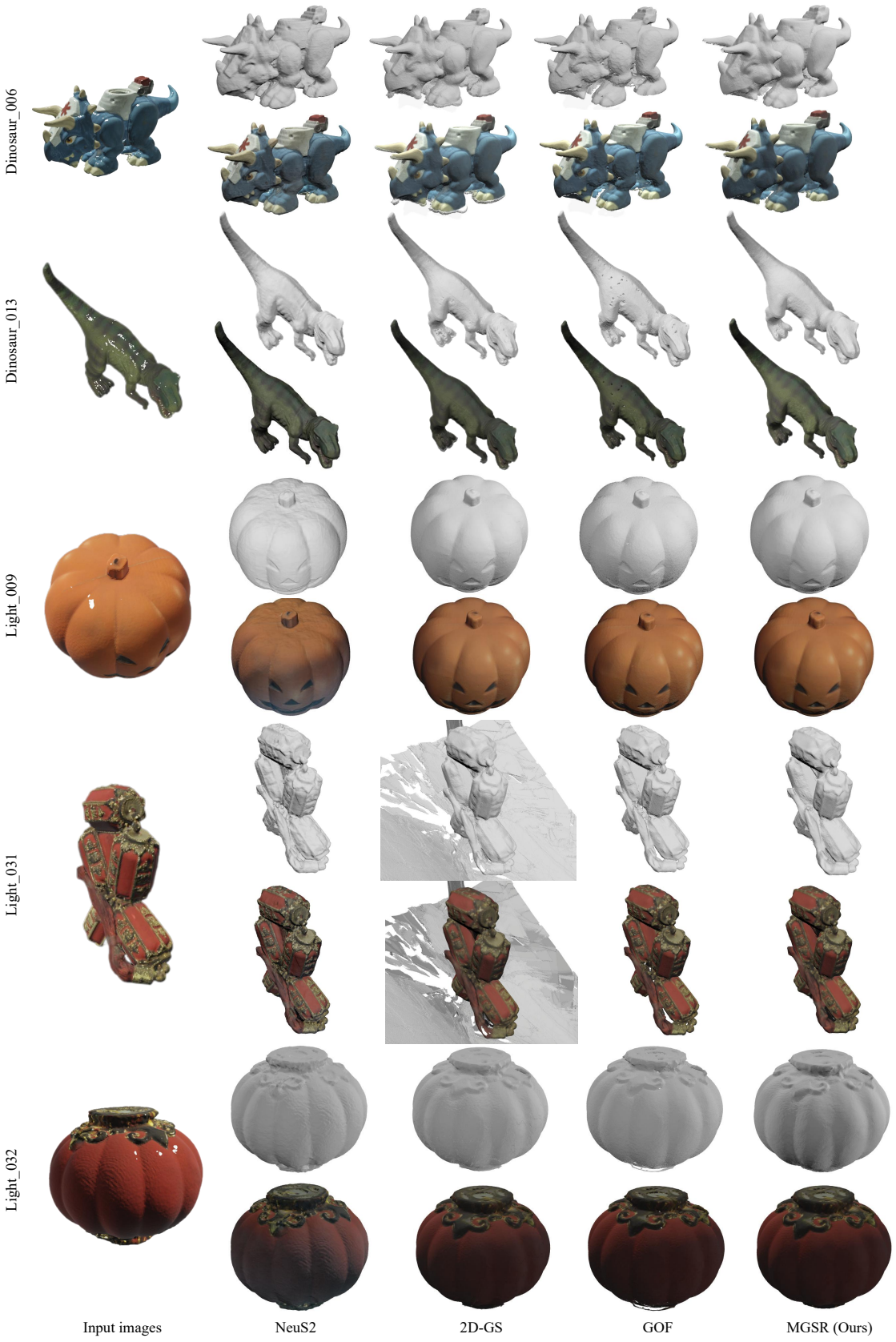


Figure 13. Instance-level SR results on Omni3DObject dataset.

Figure 14. Instance-level SR results on Omni3DObject dataset.

Figure 15. Instance-level SR results on Omni3DObject dataset.

Figure 16. Instance-level SR results on Omni3DObject dataset.

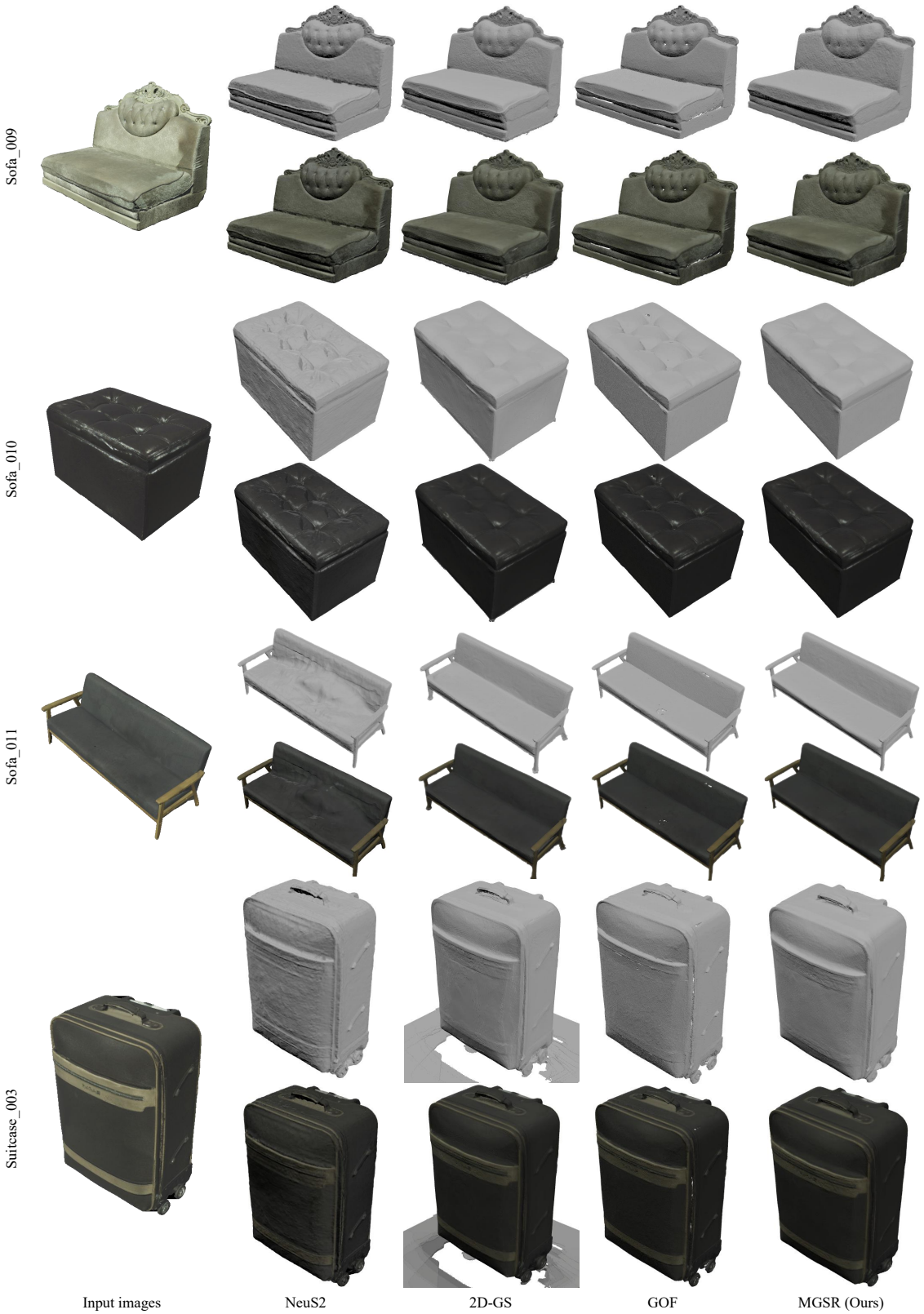


Figure 17. Instance-level SR results on Omni3DObject dataset.

Figure 18. Instance-level SR results on Omni3DObject dataset.

Figure 19. Instance-level SR results on Omni3DObject dataset.

Figure 20. Instance-level SR results on Omni3DObject dataset.

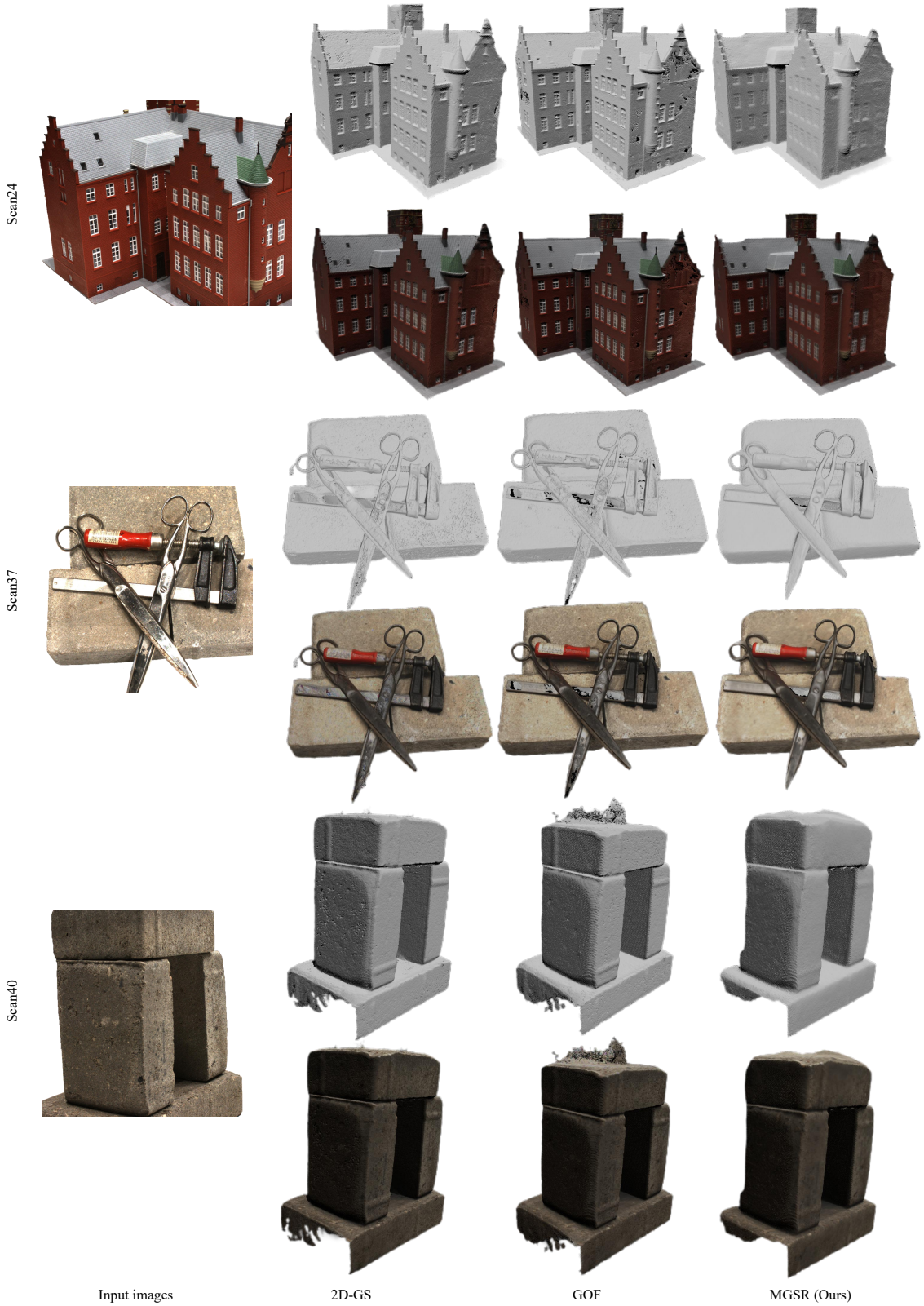


Figure 21. Instance-level SR results on DTU dataset.

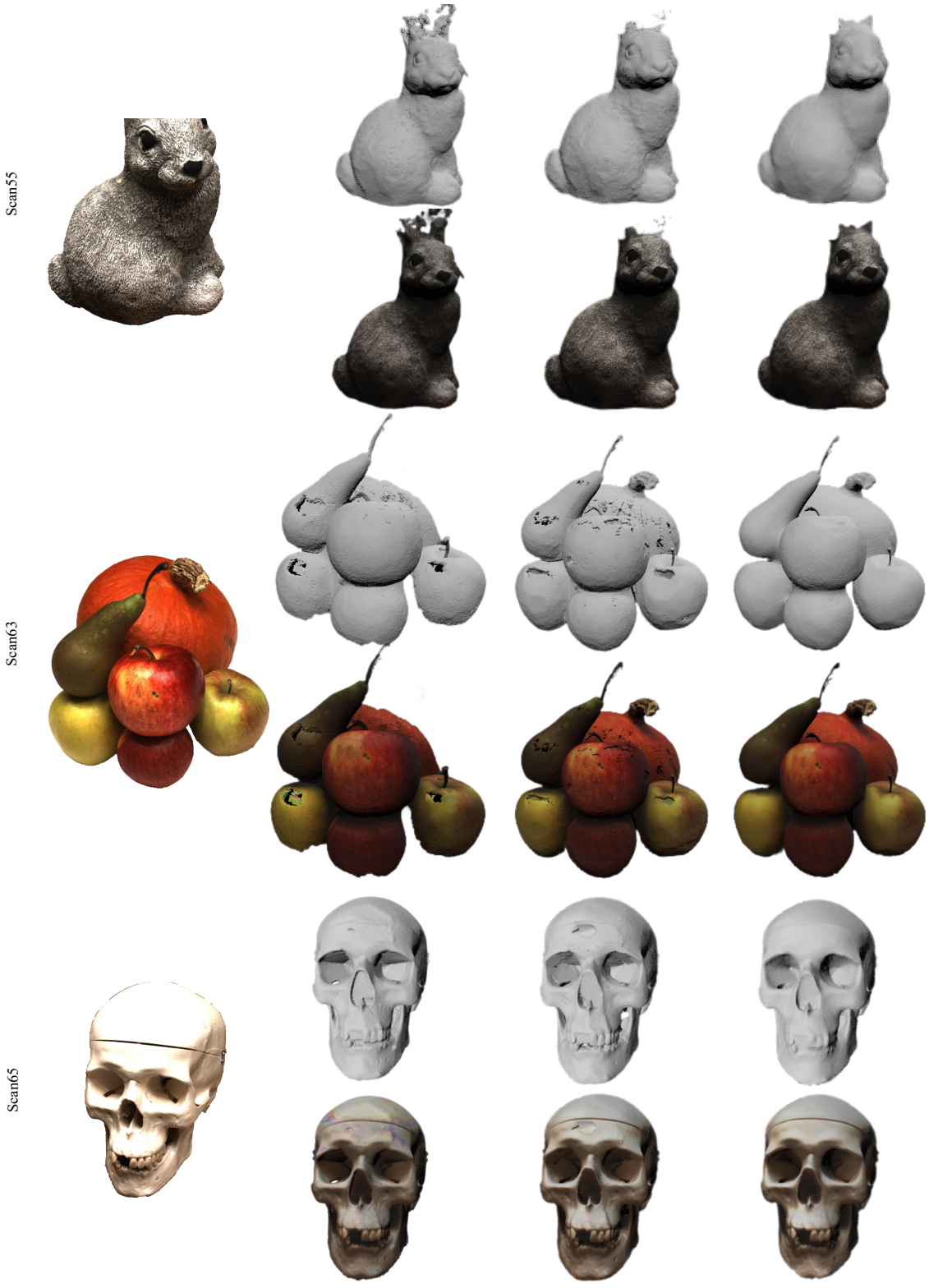


Figure 22. Instance-level SR results on DTU dataset.

Figure 23. Instance-level SR results on DTU dataset.

Figure 24. Instance-level SR results on DTU dataset.

Figure 25. Instance-level SR results on DTU dataset.

Table 12. Ablation studies of bidirectional BP and auto-stop warm-up strategy on OmniObject3D dataset. NC is multiplied by 10^2, CD is multiplied by 10^3. Method marked with * fails on certain objects, which are excluded from the average metric values presented in this table

Object	Model K				Model L*				**Model M**			
	SSIM↑	PSNR↑	NC↑	CD↓	SSIM↑	PSNR↑	NC↑	CD↓	SSIM↑	PSNR↑	NC↑	CD↓
Dinosaur_006	0.9902	37.6185	90.2938	0.0730	0.9934	38.8427	90.3657	0.0778	0.9932	38.6247	90.4567	0.0746
Dinosaur_013	0.9892	35.6438	94.0546	0.0487	0.9803	33.5458	94.0482	0.0485	0.9914	36.1045	94.0746	0.0440
Light_009	0.9842	33.0099	96.8985	2.2351	0.9862	33.2679	97.1528	1.9649	0.9858	33.3903	97.1616	2.0088
Light_031	0.9884	38.8302	87.7759	0.1493	0.9910	39.8473	88.2097	0.1584	0.9920	40.2280	88.0956	0.1581
Light_032	0.9770	34.0532	90.2556	0.3058	0.9817	34.7272	90.2377	0.3054	0.9825	34.6889	90.2715	0.3159
Light_039	0.9859	36.8948	89.9330	0.3031	0.9885	36.7214	89.6433	0.3034	0.9895	38.0387	90.1561	0.3062
Ornaments_006	0.9909	35.9048	88.9350	0.1021	0.9927	36.3036	88.8142	0.0958	0.9928	36.3158	89.0818	0.0989
Ornaments_007	0.9862	33.5829	87.6465	0.4457	0.9886	33.8908	87.6701	0.4453	0.9886	33.9089	87.8159	0.4503
Ricecooker_004	0.9784	34.3485	93.7338	0.8584	0.9836	35.0506	93.5838	0.9057	0.9838	35.1740	93.2487	0.9029
Ricecooker_006	0.9714	34.2361	94.1540	2.0880	0.9748	34.5483	94.1974	2.2726	0.9746	34.7367	94.3153	2.2044
Ricecooker_007	0.9853	36.5868	94.3110	1.2944	0.9882	37.1607	94.2153	1.2952	0.9884	37.2851	94.5260	1.3389
Ricecooker_008	0.9845	35.4008	92.7921	0.8269	0.9880	36.5642	93.3433	0.8833	0.9878	36.4294	92.9716	0.8535
Ricecooker_009	0.9812	36.6171	93.5544	2.1285	0.9862	37.8864	93.5668	2.0946	0.9862	37.9426	93.6561	1.9475
Sofa_006	0.9827	39.2141	89.4597	1.3003	0.9858	40.0283	89.3436	1.2615	0.9860	40.0727	89.4616	1.2888
Sofa_007	0.9606	35.0700	90.0267	2.1101	0.9709	36.0204	90.6215	2.0761	0.9696	35.9044	89.9815	2.0386
Sofa_009	0.9756	34.4014	87.0719	1.7128	0.9822	35.3364	87.1454	1.7317	0.9821	35.3156	87.1883	1.9120
Sofa_010	0.9836	36.5641	90.8665	1.8130	0.9850	37.1427	90.6086	1.8929	0.9857	37.3777	90.6918	1.7355
Sofa_011	0.9930	41.0359	89.4050	0.3928	0.9936	41.4638	89.4231	0.4139	0.9940	41.6715	89.7134	0.4064
Suitcase_001	0.9715	35.5951	90.6384	0.9785	0.9766	36.1709	89.7044	1.0042	0.9784	36.5393	89.7266	0.9542
Suitcase_003	0.9820	37.3058	88.7202	0.1401	0.9856	38.3055	88.9057	0.1409	0.9863	38.5199	88.7826	0.1370
Suitcase_004	0.9790	38.5087	81.0100	0.3056	0.9834	39.3719	81.1360	0.3487	0.9836	39.5490	81.2096	0.3125
Suitcase_005	0.9885	39.3619	87.1081	0.1436	0.9913	40.5460	87.7543	0.1553	0.9906	40.3685	86.3998	0.1478
Suitcase_006	0.9887	38.1695	91.8769	0.6332	0.9906	38.7087	91.8136	0.6696	0.9904	38.5725	92.1355	0.5843
Suitcase_007	0.9858	38.5319	85.0053	0.2613	0.9886	39.4112	84.1985	0.3117	0.9903	39.9557	85.5167	0.2656
Table_009	0.9847	39.6718	89.4222	1.5149	0.9845	39.4550	89.3656	1.4432	0.9870	40.4433	89.4704	1.4223
Table_017	0.9641	37.4614	92.3904	2.2920	0.9664	38.4373	92.1324	2.2608	0.9663	38.3512	92.1693	2.2957
Table_019	0.9673	36.6661	90.6596	1.5155	0.9639	36.7295	90.8214	1.6402	0.9708	38.2598	90.7717	1.5752
Table_020	0.9738	37.7489	92.3219	0.9492	0.9549	32.2389	92.1171	1.0314	0.9771	38.8922	92.3414	0.9811
Table_023	0.9842	36.0065	93.6873	0.1190	-	-	93.8726	0.1211	0.9921	42.5329	93.8091	0.1192
Table_026	0.9823	35.1996	93.1973	0.1866	0.9839	35.6746	92.9366	0.1822	0.9840	35.5904	92.8703	0.1750
Mean	0.9813	36.6413	90.5735	0.9076	0.9831	37.0137	90.5650	0.9179	**0.9850**	**37.6928**	**90.6024**	**0.9018**