SGS-Intrinsic: Semantic-Invariant Gaussian Splatting for Sparse-View Indoor Inverse Rendering

Supplementary Material

This supplementary material provides a detailed description of our method's implementation and more visual comparison results.

1. Representation

In the vanilla 3DGS, each 3D Gaussian utilizes learnable parameters $\boldsymbol{T} = \{\boldsymbol{p}, \boldsymbol{s}, \boldsymbol{q}\}$ and $\boldsymbol{C} = \{\alpha, \boldsymbol{f}_c\}$ to describe its geometric properties and volumetric appearance, respectively. Here, $\boldsymbol{p}$ denotes the position vector, $\boldsymbol{s}$ denotes the scaling vector, $\boldsymbol{q}$ denotes the unit quaternion for rotation, α denotes the opacity, and $\boldsymbol{f}c$ denotes the spherical harmonics (SH) coefficients for view-dependent color. In SGS-Intrinsic, we extend $\boldsymbol{C}$ to $\{\alpha, \boldsymbol{f}_c, \boldsymbol{f}_s, \boldsymbol{n}, \boldsymbol{a}, \boldsymbol{r}\}$ to describe the material and semantic properties of the 3D Gaussian, where $\boldsymbol{a}$ and $\boldsymbol{r}$ denote the albedo and roughness values, $\boldsymbol{n}$ denotes the Normal, and $\boldsymbol{f}_s$ denotes the semantic feature.

2. Implementation Details and Constraints of Hybrid Illumination Models

We employ physically-based rendering (PBR) to model the view-dependent appearance. Specifically, we utilize the CookTorrance model to formulate the bidirectional reflectance distribution function (BRDF) fr:

$$
\begin{aligned}
f_r(\omega_i, \omega_o) &= \underbrace{(1-M)\frac{A}{\pi}}_{\text{diffuse component}} + \underbrace{\frac{DFG}{4(n\cdot\omega_i)(n\cdot\omega_o)}}_{\text{specular component}}, \\
h &= \text{normalize}(\omega_o + \omega_i), \\
F_0 &= (1-M)\cdot 0.04 + M\cdot A, \\
D(n,h) &= \frac{R^4}{\pi\left(n\cdot h\left(R^4-1\right)+1\right)^2}, \\
F(\omega_i, n) &= F_0 + (1-F_0)\left(1-n\cdot\omega_i\right)^5, \\
G(\omega_o, \omega_i, h) &= G_1(\omega_o, h)\cdot G_1(\omega_i, h), \\
G_1(n,h) &= \frac{1}{1+n\cdot h\sqrt{R^4+n\cdot h-R^4\cdot n\cdot h}},
\end{aligned}
\tag{1}
$$

where A, R, and M denote the albedo, roughness, and metallicity, respectively. The normal distribution function (NDF) D, fresnel function F, and geometry function G are derived from physical materials.

We employ a mixture model of a set of spherical Gaussians (SGMs) to represent the localized highlight illumination as $L_o^{\text{SGMs}}(\boldsymbol{x}, \boldsymbol{\omega}_o)$:

$$
L_o^{\text{SGMs}}(\boldsymbol{x}, \boldsymbol{\omega}_o) = \sum_i^{N_{\text{light}}} \frac{f_{ri}^{\text{SGM}}(\boldsymbol{n}\cdot\boldsymbol{\omega}_i) * \mathrm{V}_i}{d_i^2} \sum_j^{N_{\text{sg}}} W_{i,j} SG(j),
\tag{2}
$$

where d_i denotes the distance from the i-th spherical Gaussian mixture to the surface point $\boldsymbol{x}$. f_{ri}^{SGM} denotes the BRDF function, and SG denotes the spherical Gaussian Function, respectively. Each spherical Gaussian mixture contains N_{sg} spherical Gaussians.

Following [3], we represent environment lighting as a learnable cubemap with size $6\times512\times512\times3$. Using split-sum approximation, the environment radiance can be decomposed as:

$$
\begin{aligned}
L_o^{\text{env}}(\boldsymbol{x}, \boldsymbol{\omega}_o) &= L_{o\text{-diff}}^{\text{env}} + L_{o\text{-spec}}^{\text{env}}, \\
L_{o\text{-diff}}^{\text{env}} &\approx K_{\text{diff}}^{\text{env}} I_{\text{diff}}^{\text{env}}, \quad K_{\text{diff}}^{\text{env}} = (1-M)\frac{A}{\pi}, \\
L_{o\text{-spec}}^{\text{env}} &\approx \underbrace{\int_\Omega \frac{DFG}{4(\boldsymbol{n}\cdot\boldsymbol{\omega}_o)}\,\mathrm{d}\boldsymbol{l}}_{\text{Environment BRDF } (K_{\text{spec}}^{\text{env}})} \cdot \underbrace{\int_\Omega DL_i(\boldsymbol{l})(\boldsymbol{l}\cdot\boldsymbol{n})\,\mathrm{d}\boldsymbol{l}}_{\text{Pre-Fil. Env. Map } (I_{\text{spec}}^{\text{env}})}.
\end{aligned}
\tag{3}
$$

We further introduce two regularization terms to en-

Algorithm 1 Pseudo view sampling algorithm

Require: Training views V_{train}, test views V_{test}; interpolation counts $N_{\text{train}}, N_{\text{test}}$; distance weights λ_t, λ_r

function POSEDIST($(q_1, t_1), (q_2, t_2)$)
  $D \leftarrow \lambda_t \|t_1 - t_2\|_2 + \lambda_r \arccos\left(2(q_1^\top q_2)^2 - 1\right)$
  return D
end function

for each input view (q, t) in V_{train} **do**
  Find nearest training view $(q_{\text{tr}}^{\text{nn}}, t_{\text{tr}}^{\text{nn}})$ in V_{train} using POSEDIST
  Generate N_{train} spline-interpolated pseudo-views
  $\mathcal{S}_{\text{train}} \leftarrow \text{CubicSpline}\big((q,t), (q_{\text{tr}}^{\text{nn}}, t_{\text{tr}}^{\text{nn}})\big)$
  Find nearest test view $(q_{\text{te}}^{\text{nn}}, t_{\text{te}}^{\text{nn}})$ in V_{test} using POSEDIST
  Generate N_{test} spline-interpolated pseudo-views
  $\mathcal{S}_{\text{test}} \leftarrow \text{CubicSpline}\big((q,t), (q_{\text{te}}^{\text{nn}}, t_{\text{te}}^{\text{nn}})\big)$
  Randomly sample $v_{\text{tr}} \sim \mathcal{S}_{\text{train}},\ v_{\text{te}} \sim \mathcal{S}_{\text{test}}$
  Construct view-sequence $\mathcal{V} \leftarrow \{v_{\text{tr}}, (q,t), v_{\text{te}}\}$
  Feed sequence $\mathcal{V}$ into the SAM2 model
end for

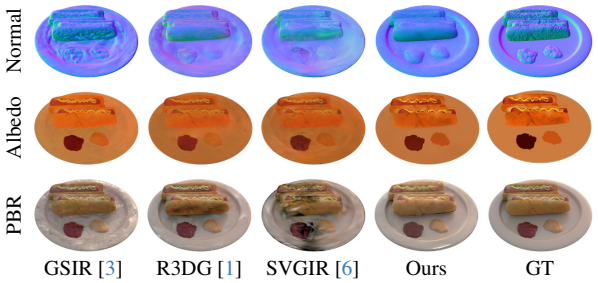


Figure 1. **Qualitative evaluation on the TensoIR dataset.**

courage compact and stable light representations:

$$L_{\text{pos}} = \sum_{i}^{N_{\text{light}}} \frac{1}{d_{i,\text{near}}}, \tag{4}$$

$$L_{\text{val}} = \sum_{i}^{N_{\text{light}}} \sum_{j}^{N_{\text{sg}}} w_{i,j}. \tag{5}$$

3. Pseudo Views Sampling Algorithm

Please refer to the pseudocode in Algorithm 1 for further details.

4. Deshaowing Module design

The deshadowing module incorporates a hash encoder to perform multi-resolution encoding of the 3D Gaussian field, along with a lightweight MLP decoder that predicts the occlusion relationship of a given spatial point relative to a point light source. The specific details of this module are presented in Table 1.

Table 1. Architecture of Deshadowing Module

Module	Parameter	Value
HashEnc	Number of levels	16
	Max. entries per level	2^{19}
	feature dimensions	2
	Coarsest resolution	32
	Finest resolution	4096
MLP	MLP layers	$36 \times 32 \times 32 \times 1$
	Initialization	Kaiming-uniform
	Final activation	Sigmoid

5. Training Details

We implement our method and conduct all experiments on a single NVIDIA RTX 4090 GPU.

For the first-stage geometric reconstruction, we set the total number of training iterations to 7000. Virtual view sampling and the semantic consistency constraint between virtual and real views are activated after 2000 iterations. The learnable semantic feature dimension attached to each Gaussian is set to 32. We employ CLIP ViT-B/16 [4] for semantic feature extraction and SAM2 Hiera-L [5] for region mask generation.

For the second stage, corresponding to the inverse rendering phase, the number of training iterations is set to 3000. At the beginning of this stage, we initialize the point light sources represented by Spherical Gaussian Mixtures (SGMs) uniformly within the scene's axis-aligned bounding box (AABB) of range $[-1.5, 1.5]$. To balance representational expressiveness and computational cost, we place $[4, 4, 4]$ point lights in total, resulting in 64 point light sources. Each point light source contains $n_{SG} = 12$ spherical Gaussians. The illumination invariance constraint is introduced after 8000 iterations, and both the self-view invariance and multi-view invariance constraints are applied every three iterations.

Table 2. **Quantitative comparison on the TensoIR dataset.**

Method	Albedo			NVS For PBR		
	PSNR↑	SSIM↑	LIPPS↓	PSNR↑	SSIM↑	LIPPS↓
GSIR [3]	23.67	0.90	0.11	23.98	0.82	0.14
R3DG [1]	22.46	0.89	0.12	25.11	0.85	0.12
SVGIR [6]	24.98	0.90	0.13	21.05	0.68	0.19
GeoSplat [7]	24.31	0.89	0.13	25.45	**0.87**	0.12
IRGS [2]	22.57	0.86	0.14	25.32	0.85	0.13
Ours	**25.65**	**0.93**	**0.10**	**26.02**	**0.87**	**0.11**

Table 3. Training time breakdown

	VGGT	LSeg	StableNorm	RGBX	Training
Time	∼20s	∼2min	∼10s	∼30s	∼40min

6. Evaluation on the TensoIR dataset

We also evaluate our approach on the TensoIR dataset. The quantitative and qualitative results are presented in Table 2 and Figure 1, respectively. As shown, our method achieves leading performance on object-level data, further validating its robustness and generalization capability.

7. Qualitative Results of Geometry and Semantic Reconstruction

Figure 2 demonstrates the respective impacts of the normal prior and the semantic prior on the reconstruction results. The normal priors provide smoother geometric surfaces, significantly improving the quality of geometry reconstruction. The semantic priors, together with multi-view semantic consistency supervision, effectively alleviate the geometric reconstruction issues of 3D Gaussians in textureless regions.

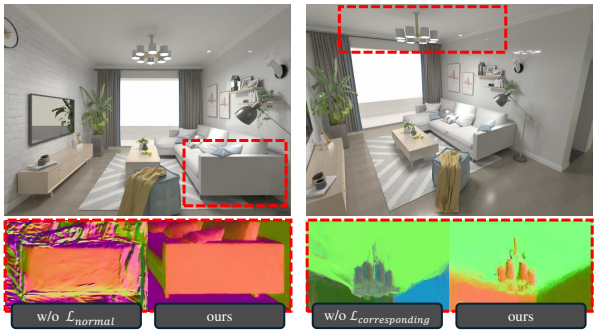


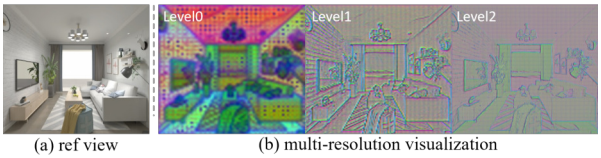
Figure 2. **Effect of normal and semantic priors.**

8. Breakdown of the total time

The total duration of the preprocessing workflow is summarized in Table 3, amounting to a cumulative processing time of about 3 minutes.

9. Visualizing multi-resolution features

As shown in Figure 3, the multi-resolution features are highly correlated with the scene geometry.



(a) ref view (b) multi-resolution visualization

Figure 3. **Multi-resolution feature visualization and comparison with GT view**

10. More Visual Comparison

Figure 4,5,6,7,8,9 provide more visual comparison on material estimation and Novel View PBR Rendering result. Figure 10 presents the detailed decomposition results of our hybrid illumination model, while Figure 11 provides additional application results.

References

[1] Jian Gao, Chun Gu, Youtian Lin, Zhihao Li, Hao Zhu, Xun Cao, Li Zhang, and Yao Yao. Relightable 3d gaussians: Realistic point cloud relighting with brdf decomposition and ray tracing. In *ECCV*, 2024. 2

[2] Chun Gu, Xiaofei Wei, Zixuan Zeng, Yuxuan Yao, and Li Zhang. Irgs: Inter-reflective gaussian splatting with 2d gaussian ray tracing. In *CVPR*, 2025. 2

[3] Zhihao Liang, Qi Zhang, Ying Feng, Ying Shan, and Kui Jia. Gs-ir: 3d gaussian splatting for inverse rendering. In *CVPR*, 2024. 1, 2

[4] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 2

[5] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, and Laura Gustafson. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 2

[6] Hanxiao Sun, Yupeng Gao, Jin Xie, Jian Yang, and Beibei Wang. Svg-ir: Spatially-varying gaussian splatting for inverse rendering. In *CVPR*, pages 16143–16152, 2025. 2

[7] Kai Ye, Chong Gao, Guanbin Li, Wenzheng Chen, and Baoquan Chen. Geosplatting: Towards geometry guided gaussian splatting for physically-based inverse rendering. In *ICCV*, 2025. 2

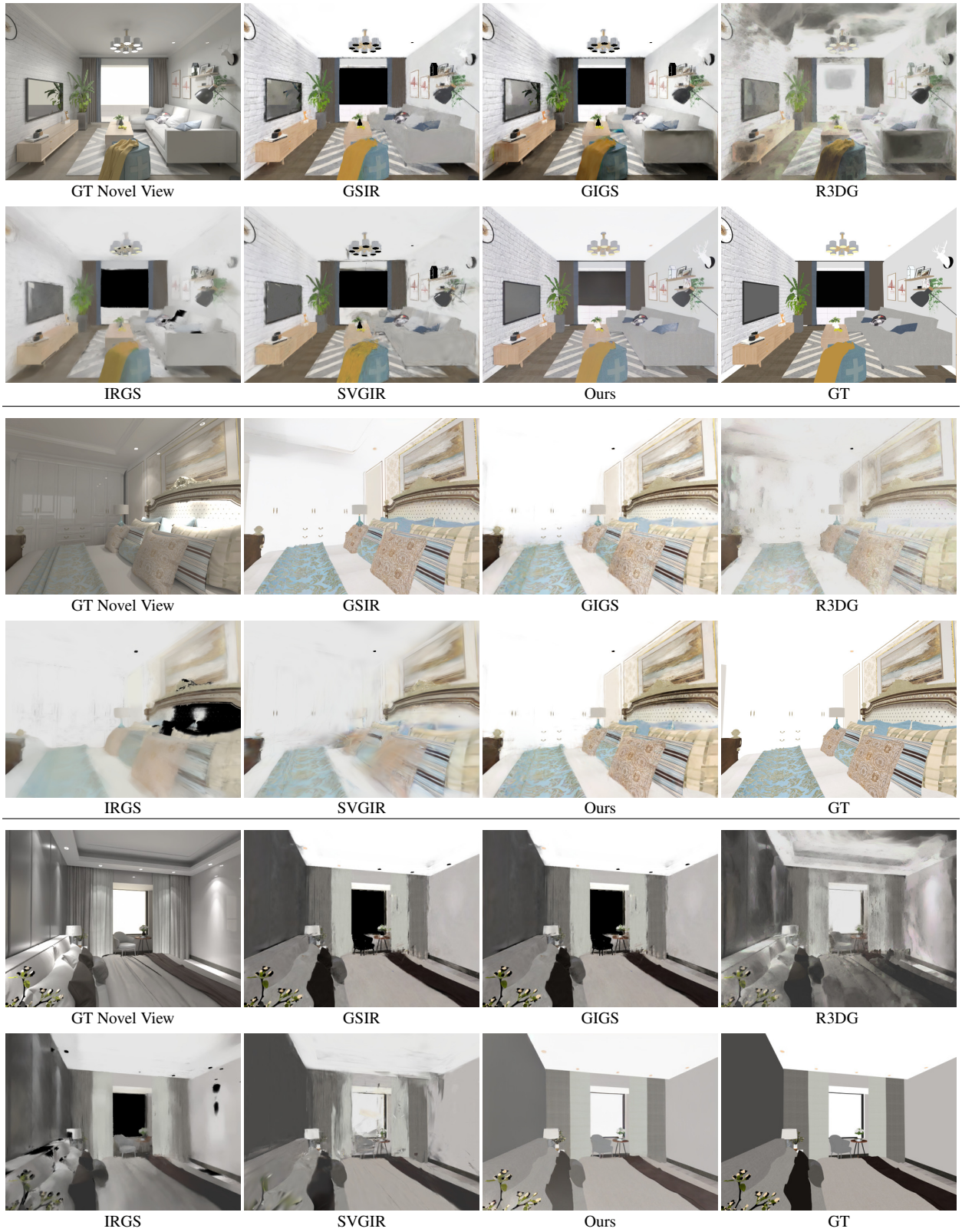


Figure 4. **More qualitative comparison of albedo estimation on the synthetic InteriorVerse dataset.**

Figure 5. **More qualitative comparison of albedo estimation on the synthetic InteriorVerse dataset.**

Figure 6. **More qualitative comparison of PBR Rendered Novel View on the synthetic InteriorVerse dataset.**

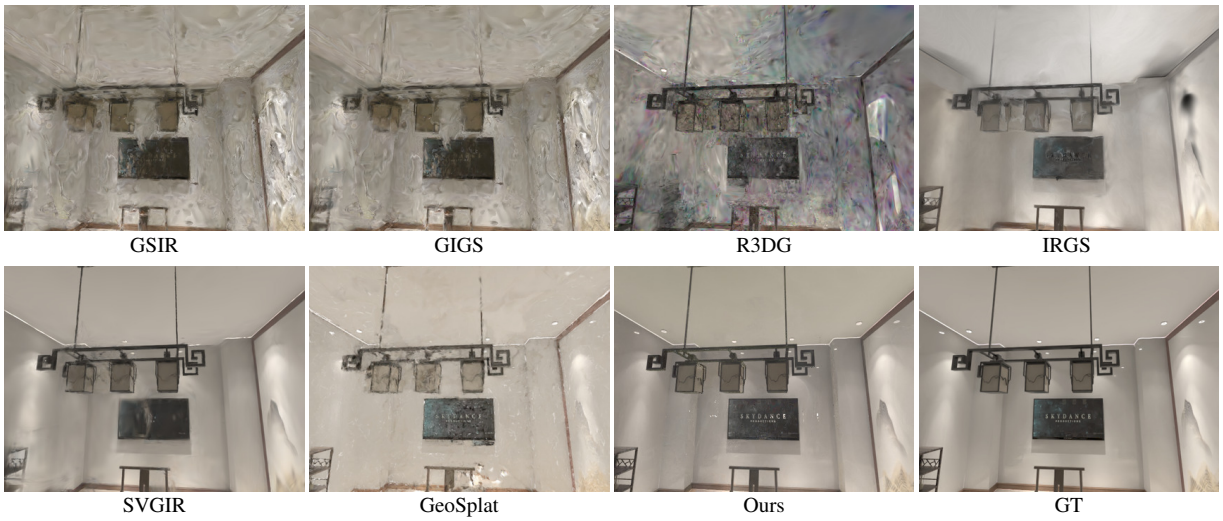


Figure 7. **More qualitative comparison of PBR Rendered Novel View on the synthetic InteriorVerse dataset.**

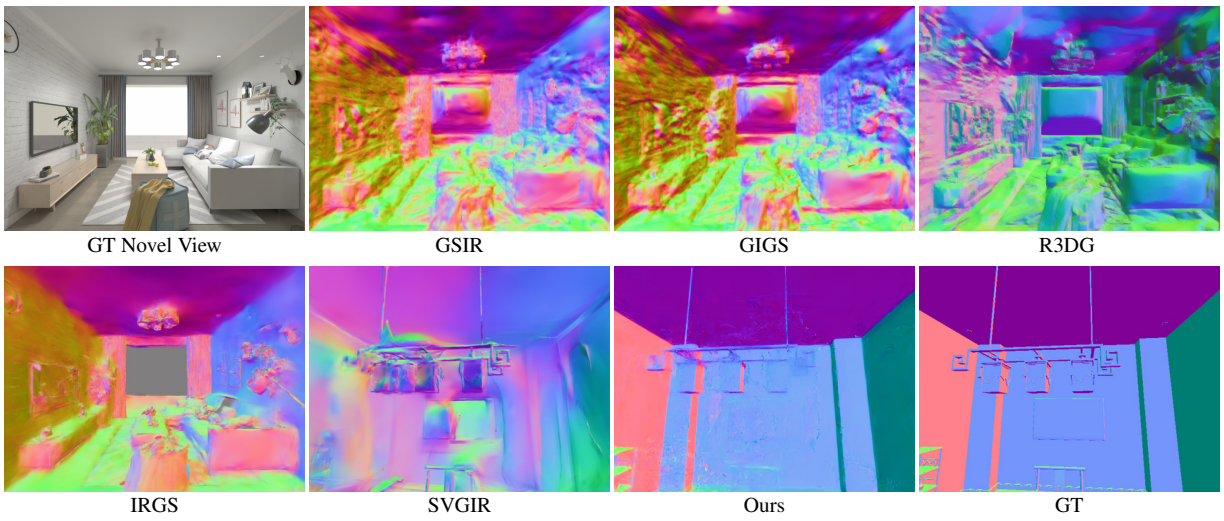


Figure 8. **More qualitative comparison of albedo estimation on the synthetic InteriorVerse dataset.**

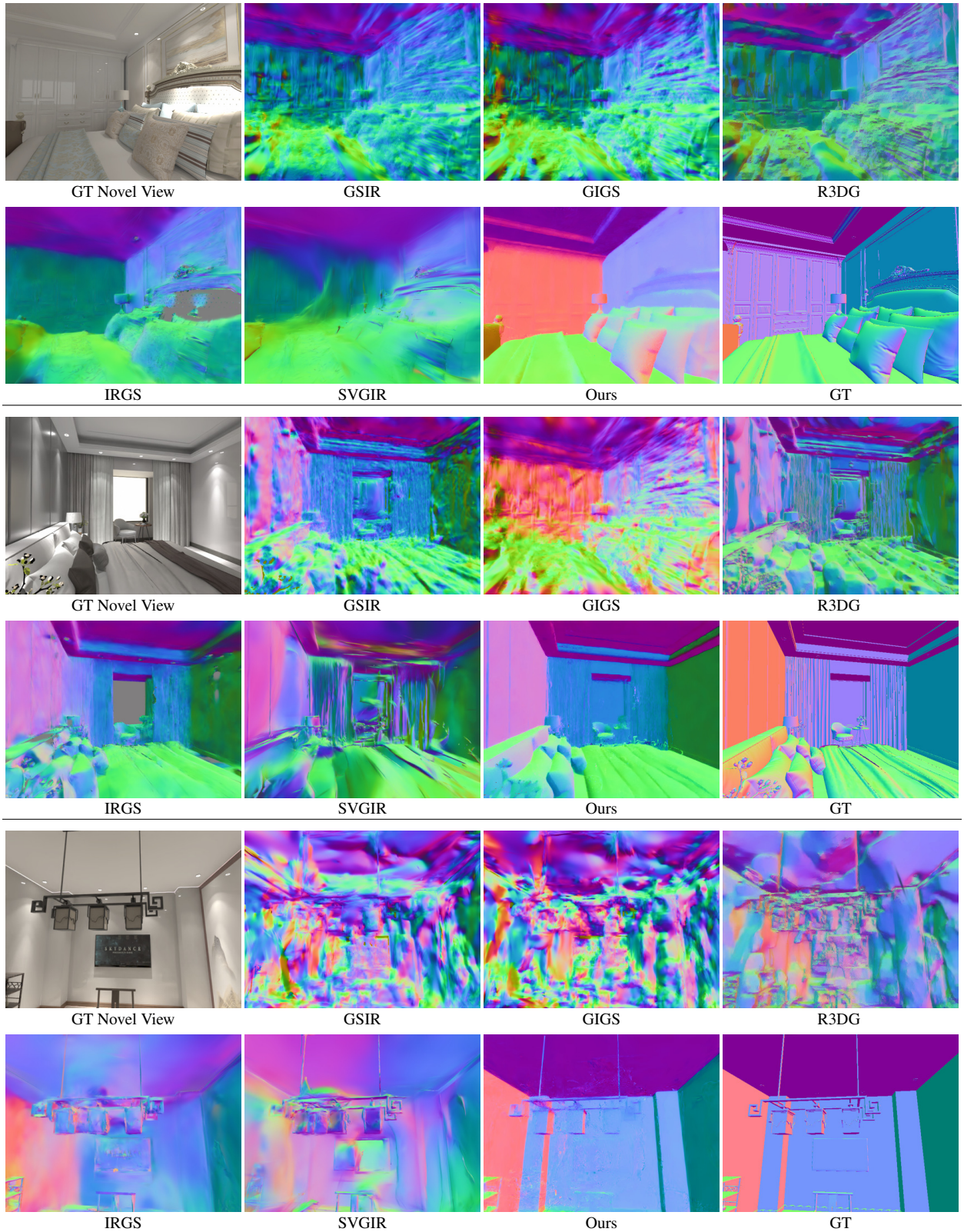


Figure 9. **More qualitative comparison of albedo estimation on the synthetic InteriorVerse dataset.**

Figure 10. **More illumination decomposition results on the synthetic InteriorVerse dataset.**

Figure 11. **More results of object insertion, material and light editing**