3DIS: REAL-TIME HYBRID RELIGHTING VIA 3D IMPORTANCE SAMPLING FROM 3DGS SCENES

Anonymous authors
Paper under double-blind review

ABSTRACT

3D Gaussian Splatting (3DGS) supports high fidelity real time scene rendering, but its baked appearance representation makes it difficult to achieve harmonious illumination for inserted meshes. We propose a scene-level lighting estimation framework that recovers a reusable, spatially varying illumination representation from an existing 3DGS scene, enabling real-time and illumination-consistent rendering of inserted objects. Instead of predicting lighting independently for each view or object pose, our method estimates a compact virtual-point-light configuration from multi-location HDR observations of the 3DGS scene. To make this estimation both spatially stable and source-aware, we introduce a mixed 3D importance sampling strategy that allocates point lights across scene coverage, HDR radiance importance, and semantic light-source regions such as windows and lamps. The resulting point-light configuration captures both dominant direct illumination and distributed environmental light, enabling inserted meshes to be rendered consistently to both the 3DGS environment and the viewing direction (like specular reflection), even the inserted mesh is moved/manipulated. Experiments on multiple indoor scenes demonstrate that our method provides a practical solution for illumination-consistent mesh insertion and interactive relighting in reconstructed Gaussian environments.

1 INTRODUCTION

Recent advances in neural scene representations have enabled high-quality reconstruction and real-time rendering of complex environments. In particular, 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) represents a scene using explicit anisotropic Gaussian primitives and supports efficient rasterization with high visual fidelity, making it attractive for immersive applications such as virtual reality (Tu et al., 2025), augmented reality (Zhai et al., 2025), and interactive visualization (Sinha et al., 2024). However, the appearance encoded by 3DGS is primarily optimized for reproducing the captured scene from the observed viewpoints. It does not explicitly provide a reusable representation of the scene illumination, which makes the integration of separately created virtual objects (e.g. triangular mesh) difficult.

Inserting a textured 3D mesh into a reconstructed 3DGS scene requires more than geometric alignment and alpha compositing. The inserted object should be illuminated with a lighting condition that is consistent with the surrounding scene. This may include both the direct (e.g. the contribution from dominant light sources) and indirect illumination (reflection from the nearby environment). Without such illumination coupling, the inserted object may appear visually inconsistent with the reconstructed environment. Such inconsistency is especially apparent when the object is moving, or as viewed from different directions. Existing image-based and video-based illumination estimation methods can recover HDR environment maps for specific viewpoints, but their predictions are generally tied to a particular query location or camera trajectory. They therefore do not directly provide a persistent spatial lighting representation for interactive object manipulation.

We propose to address this problem by estimating a reusable set of 3D point light sources that approximates the illumination of the given 3DGS scene. This set of point lights only cast light to the inserted objects, and does not affect the 3DGS during the final rendering. Our key observation is that the Gaussian scene can be queried at multiple spatial locations to obtain local RGB-D observations of the surrounding environment. Such observations provide the supervision for reconstructing a set

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

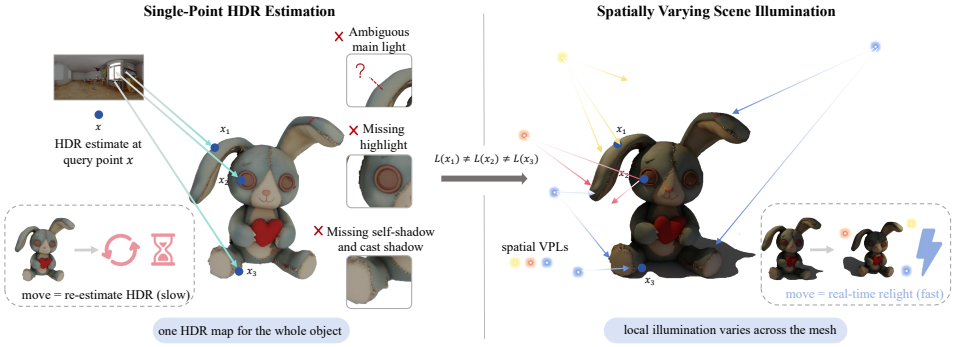

Figure 1: Comparison of single-point HDR estimation and spatially varying illumination. Left: estimating HDR at a single query point x applies one HDR map to the entire object, which can produce an ambiguous main-light direction, miss highlights, and omit self-shadows and cast shadows; moving the object requires recomputing the HDR estimate. Right: spatial virtual point lights (VPLs) provide illumination that varies across the mesh, enabling fast, real-time relighting as the object moves. $L(x_i)$ denotes the illumination at query point x_i.

of point light sources, each with different position, intensity, and color. Collectively, these point lights ressemble the complex illumination contributions, including the direct illumination from the windows (area light) and lamps (area/volume lights), and the indirect illumination reflected from the wall or nearby sofa. Our work is inspired by the early works in 2D importance sampling (Ostromoukhov et al., 2004; Wan et al., 2005), in which the illumination contribution from the 2D environment map can be approximated by a set of directional lights (2D samples on the spherical environment map). In contrast, our work is *3D importance sampling* that approximates the illumination contribution, from a given 3DGS, by a set of 3D point light sources (3D samples in a volume).

With this approach, we developed an real-time hybrid renderer that first renders the 3DGS scene, and then render the inserted object illuminated by the estimated point lights. Since the estimated lights are defined in the 3D space rather than for a single environment map observed at a fixed location, the inserted object can be relit at any 3D location and from any viewpoint, without repeatedly invoking a location-dependent, view-dependent generative illumination estimator as in existing methods (Liang et al., 2026). The representation also supports source-aware energy allocation and environment-color regularization using the original Gaussian-rendered appearance, improving the visual stability of the lighting applied to the inserted mesh. Our contributions are summarized as follows:

- Given a 3D Gaussian Splatting scene, we propose a reusable, scene-level 3D illumination approximation for harmoniously relighting inserted mesh objects. Unlike existing methods that rely on location-specific environment maps, our representation can be reused across different object locations.
- We introduce a 3D importance sampling strategy that approximates the scene illumination with a compact set of point lights, capturing both spatial coverage and important lighting contributions.
- We develop a real-time hybrid 3DGS–mesh rendering pipeline that applies the estimated lighting to inserted objects with realistic shading and self-shadowing. We also construct a benchmark for evaluating mesh insertion in 3DGS scenes and evaluate our pipeline across indoor scenes with diverse lighting conditions.

2 Related Work

2.1 Scene Lighting Estimation

Scene lighting estimation aims to recover illumination from visual observations for inverse rendering, relighting, and augmented reality. Classical and learning-based methods estimate indoor or

outdoor illumination from a single image, panorama, or video, commonly representing lighting as HDR environment maps, spherical harmonics, or parametric light sources (Gardner et al., 2017; Hold-Geoffroy et al., 2017; Somanath & Kurz, 2021; Dastjerdi et al., 2023; Li et al., 2023; Xu et al., 2022). Recent generative methods further improve high-frequency illumination prediction by leveraging diffusion or video priors. DiffusionLight estimates HDR light probes by inpainting a chrome ball into the input image (Phongthawee et al., 2024), LuxDiT predicts HDR environment maps from images or videos using a diffusion transformer (Liang et al., 2026), and LiMo estimates spatiotemporal HDR illumination with explicit 3D conditioning and HDR merging (Bolduc et al., 2026). These methods produce plausible lighting for target images, query points, or trajectories, but their predictions are typically tied to specific views or locations. Recent 3D-aware lighting methods move towards spatially varying illumination. GaSLight represents spatial HDR lighting with Gaussian splats estimated from regular images (Bolduc et al., 2025), while relightable 3DGS methods decompose Gaussian appearance into geometry, material, visibility, shadow, transport, or illumination-related factors (Liang et al., 2024; Jiang et al., 2024; Guo et al., 2024; Chen et al., 2024; Lai et al., 2025; Ye et al., 2024; Sun et al., 2025; Zheng et al., 2026). These methods are designed to relight or improve the reconstructed scene representation itself, often requiring modified Gaussian attributes, inverse rendering, or precomputed transport. In contrast, our method estimates an external scene-level point-light configuration from an existing 3DGS scene. By querying the Gaussian scene at multiple spatial probes and optimizing a compact set of virtual point light sources, our approach recovers reusable illumination for downstream interactive object relighting.

2.2 Illumination-Consistent Insertion

Illumination-consistent insertion seeks to place virtual objects into images, videos, or reconstructed 3D scenes while matching the target environment's lighting, shadows, and appearance. Image-based and neural compositing methods estimate illumination and shadow cues from the target image to render or harmonize inserted objects (Ren et al., 2026). More recent methods exploit diffusion or generative priors for lighting-aware 3D insertion and harmonization (Skorokhodov et al., 2025; Yu et al., 2025). However, image-level harmonization and per-query lighting estimation do not directly provide a persistent 3D lighting representation for interactive object movement. Object insertion in Gaussian scenes has recently received increasing attention. D3DR (Skorokhodov et al., 2025) optimizes inserted 3DGS objects using diffusion priors to correct lighting and visual inconsistencies; TranSplat (Yu et al., 2025) transfers objects between Gaussian scenes with lighting-consistent relighting; ComGS studies efficient 3D object-scene composition with surface octahedral probes (Gao et al., 2026); and LightHarmony3D predicts insertion-location HDR illumination and shadows for mesh insertion in 3DGS scenes (Huang et al., 2026). These works highlight the importance of lighting-aware composition in reconstructed Gaussian environments. Our work shares this goal but focuses on scene-level lighting estimation: instead of predicting a view-specific or location-specific illumination condition, we reconstruct a reusable spatial point-light configuration from the 3DGS scene using multi-probe HDR observations, semantic light-source priors, and mixed 3D importance sampling.

3 Method

3.1 Preliminaries

3D Gaussian Splatting. A scene reconstructed from posed multi-view images can be represented by a set of anisotropic 3D Gaussians (Kerbl et al., 2023):

$$\mathcal{G} = \{\mathcal{G}_i\}_{i=1}^{N_G},$$

where each Gaussian is parameterized by its center $\boldsymbol{\mu}_i \in \mathbb{R}^3$, covariance matrix $\boldsymbol{\Sigma}_i \in \mathbb{R}^{3\times3}$, opacity α_i, and view-dependent color $\mathbf{c}_i(\mathbf{v})$ represented by spherical harmonics:

$$\mathcal{G}_i = (\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, \alpha_i, \mathbf{c}_i(\mathbf{v}))\,.$$

The covariance matrix is parameterized as

$$\boldsymbol{\Sigma}_i = \mathbf{R}_i\mathbf{S}_i\mathbf{S}_i^\top\mathbf{R}_i^\top,$$

where $\mathbf{R}_i$ and $\mathbf{S}_i$ denote the rotation and scaling parameters, respectively.

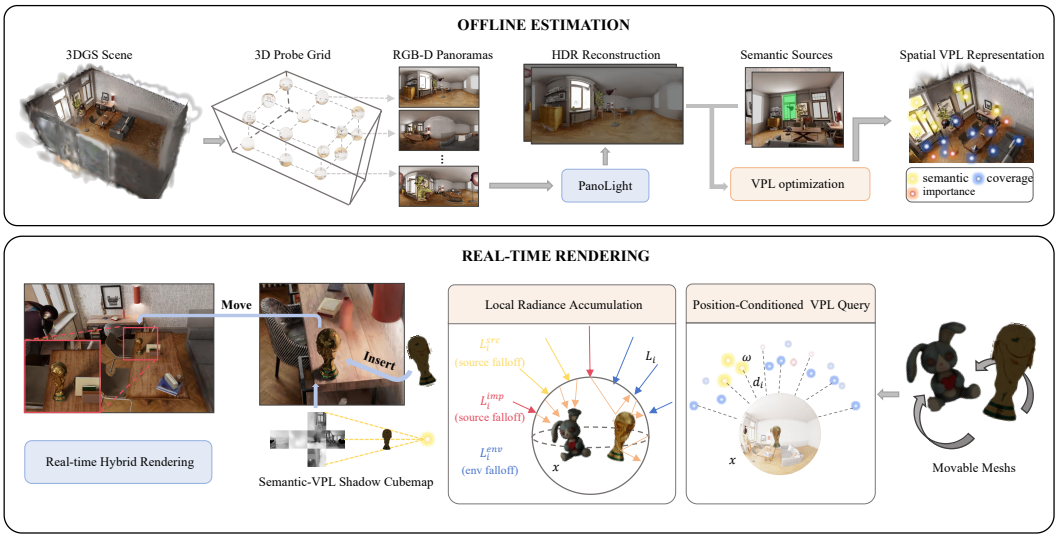

Figure 2: Overview of our scene illumination estimation and real-time rendering pipeline. **Offline estimation:** We sample panoramas from a 3D Gaussian Splatting (3DGS) scene using a 3D probe grid. PanoLight reconstructs an HDR environment map from the RGB-D panoramas. We identify semantic light sources and optimize a spatial virtual point light (VPL) representation, with lights encoding semantic importance and spatial coverage. **Real-time rendering:** For a movable mesh, we query the spatial VPLs based on its current position and accumulate local radiance from source, indirect, and environment falloff terms. We also generate a semantic-VPL shadow cubemap. These components are integrated in our real-time hybrid renderer to relight the inserted mesh as it moves through the 3DGS scene.

For a camera view, each 3D Gaussian is projected onto the image plane as a 2D Gaussian. The rendered color of a pixel or camera ray $\mathbf{r}$ is computed using front-to-back alpha compositing:

$$\mathbf{C}(\mathbf{r}) = \sum_i T_i(\mathbf{r})\alpha'_i(\mathbf{r})\mathbf{c}_i(\mathbf{v}), \qquad T_i(\mathbf{r}) = \prod_{j<i}\left(1-\alpha'_j(\mathbf{r})\right),$$

where $\alpha'_i(\mathbf{r})$ denotes the projected opacity and the Gaussians are ordered by their depth along the ray. This rasterization process produces the view-dependent appearance of the reconstructed scene.

3.2 Method Overview

Our goal is to recover a reusable spatial illumination representation from an existing 3DGS scene for real-time, illumination-consistent mesh insertion. As shown in Fig. 2, our framework consists of an offline estimation stage and an online rendering stage. The offline stage converts spatial RGB panoramas into HDR illumination observations and uses semantic light source priors to optimize a compact set of virtual point lights (VPLs). At runtime, the VPLs provide position dependent illumination, while a dynamic radiance cubemap captures local scene reflections. The relit mesh, its shadow, and the original 3DGS rendering are then composited into the final image.

3.3 Spatial Illumination Observation

We sample valid probe locations

$$\mathcal{Q} = \{\mathbf{q}_k\}_{k=1}^{K}$$

within the reconstructed 3DGS scene. Locations with insufficient Gaussian coverage are discarded. At each probe, six perspective views with a 90° field-of-view are rendered along the positive and negative coordinate axes and assembled into an equirectangular observation

$$\mathcal{O}_k = (\mathbf{I}_k, \mathbf{D}_k, \mathbf{M}_k),$$

where $\mathbf{I}_k$, $\mathbf{D}_k$, and $\mathbf{M}_k$ denote the RGB panorama, depth panorama, and validity mask, respectively. PanoLight processes the RGB panorama and validity mask to reconstruct the HDR observation

$$\mathbf{H}_k = \mathcal{R}_{\mathrm{HDR}}(\mathbf{I}_k, \mathbf{M}_k).$$

The depth panorama is not provided to PanoLight. It is retained to associate the reconstructed radiance with scene geometry and to support semantic source lifting and VPL initialization. The original 3DGS RGB panoramas are also kept as appearance references for environment related chromaticity.

3.4 Semantic Guided VPL Estimation

We compress the spatial HDR observations into a compact set of virtual point lights:

$$\mathcal{V} = \{(\mathbf{p}_i, \mathbf{e}_i, \mathbf{a}_i, c_i)\}_{i=1}^{N},$$

where $\mathbf{p}_i$ and $\mathbf{e}_i$ are the position and linear RGB intensity of the ith VPL, respectively. The anchor $\mathbf{a}_i$ and category c_i record its initialization source.

Semantic source priors. Radiance based sampling may underrepresent bright sources that occupy only a small panorama region. We therefore detect source regions in rendered multiview images, propagate their masks with DEVA Cheng et al. (2023), and lift them into the scene using depth maps and camera poses. Windows are represented as fitted planes, while local lamps are represented by point positions. These semantic sources guide VPL allocation and optimization without introducing a separate illumination representation.

VPL initialization and optimization. VPL anchors are initialized using spatial coverage sampling, HDR importance sampling, and semantic source sampling. These strategies provide broad scene support, concentrate samples in high radiance regions, and allocate additional capacity near detected sources. Their relative budgets remain configurable.

For a probe $\mathbf{q}_k$ and panorama direction $\boldsymbol{\omega}$, the radiance predicted by the VPLs is

$$\widehat{\mathbf{H}}_k(\boldsymbol{\omega}) = \sum_{i=1}^{N} \frac{\mathbf{e}_i}{\|\mathbf{p}_i - \mathbf{q}_k\|_2^2 + \epsilon^2} \mathcal{K}_\kappa \left(\boldsymbol{\omega}, \frac{\mathbf{p}_i - \mathbf{q}_k}{\|\mathbf{p}_i - \mathbf{q}_k\|_2} \right),$$

where $\mathcal{K}_\kappa$ is a normalized directional lobe. We jointly optimize the VPL positions and intensities across all probes using

$$\mathcal{L}_{\mathrm{HDR}} = \frac{1}{Z} \sum_{k,\boldsymbol{\omega}} \Delta\Omega(\boldsymbol{\omega}) C_k(\boldsymbol{\omega}) \left\| \log(1 + \gamma \widehat{\mathbf{H}}_k(\boldsymbol{\omega})) - \log(1 + \gamma \mathbf{H}_k(\boldsymbol{\omega})) \right\|_1.$$

The optimization further regularizes VPL displacement and spatial clustering. Semantic priors constrain source related VPLs to their corresponding geometry, while HDR observations and 3DGS appearance references provide illumination intensity and chromaticity supervision.

3.5 Position Conditioned Hybrid Rendering

At runtime, the optimized VPL representation is queried at the current mesh position. For a mesh surface point $\mathbf{x}$ with normal $\mathbf{n}$, the direction and distance to VPL i are

$$\boldsymbol{\omega}_i(\mathbf{x}) = \frac{\mathbf{p}_i - \mathbf{x}}{\|\mathbf{p}_i - \mathbf{x}\|_2}, \qquad d_i(\mathbf{x}) = \|\mathbf{p}_i - \mathbf{x}\|_2.$$

The local VPL illumination is computed as

$$\mathbf{L}_{\mathrm{VPL}}(\mathbf{x}) = \sum_{i=1}^{N} V_i(\mathbf{x}) \frac{\mathbf{e}_i}{d_i^2(\mathbf{x}) + \epsilon^2} \max\left(\mathbf{n}^\top \boldsymbol{\omega}_i(\mathbf{x}), 0\right),$$

where $V_i(\mathbf{x})$ denotes the runtime visibility approximation. All VPLs contribute to the mesh illumination, while a source dominant VPL is selected as the representative hard shadow caster.

Dynamic reflection rendering. Direct VPL shading does not reproduce the local scene appearance reflected by glossy or metallic surfaces. We therefore construct a radiance cubemap $\mathcal{C}_{\mathbf{x}_o}$ centered at the current mesh position $\mathbf{x}_o$. The surrounding 3DGS scene is captured along six cubemap directions and modulated using the estimated scene illumination. For viewing direction $\boldsymbol{\omega}_o$, the reflection direction is

$$\boldsymbol{\omega}_r = \mathrm{reflect}(-\boldsymbol{\omega}_o, \mathbf{n}),$$

and the reflected radiance is sampled as

$$\mathbf{L}_{\mathrm{refl}}(\mathbf{x}) = \mathcal{C}_{\mathbf{x}_o}\left(\boldsymbol{\omega}_r, \ell(\rho)\right),$$

where $\ell(\rho)$ selects the cubemap mip level according to material roughness ρ.

The final mesh color is

$$\mathbf{C}_{\mathrm{mesh}}(\mathbf{x}) = \mathbf{C}_{\mathrm{base}}(\mathbf{x}) \odot \mathbf{L}_{\mathrm{VPL}}(\mathbf{x}) + \mathbf{L}_{\mathrm{refl}}(\mathbf{x}).$$

The shaded mesh, its representative source shadow, and the original 3DGS rendering are composited according to their depth relationships. When the mesh moves, the renderer updates the VPL directions and distances and refreshes the local radiance cubemap, without repeating HDR reconstruction or VPL optimization.

4 Experiments

We conduct extensive experiments to evaluate the accuracy and visual quality of our scene illumination estimation and object relighting framework. We first describe the constructed benchmark and experimental settings, then compare our method with existing illumination estimation and object insertion methods. Finally, we analyze the contribution of individual components through ablation studies.

4.1 Experimental Setup

Datasets. We construct a benchmark of five indoor scenes with paired object-free and object-present observations for quantitative evaluation. For each scene, the object-free observations are used to reconstruct the 3DGS representation and estimate scene illumination, while the corresponding object-present images provide ground-truth appearances of the inserted object. The benchmark covers diverse object geometries, materials, light-source types, and illumination conditions. We select six calibrated views around the inserted object in each scene, resulting in 30 images for evaluation. In addition, we use five indoor 3DGS scenes without paired ground truth to qualitatively assess the generalization of our method to different scene layouts and lighting configurations.

Evaluation Metrics. Quantitative evaluation is performed on the five scenes in our paired benchmark. We compare the rendered images with their ground-truth counterparts within the inserted-object regions using Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al., 2018). PSNR and SSIM measure pixel-level and structural similarity, while LPIPS evaluates perceptual agreement. Since the background is shared across methods and does not reflect the quality of object relighting, all metrics are computed within the object region of interest. The additional scenes without paired ground truth are evaluated qualitatively.

Implementation Details. We reconstruct each object-free scene using 3D Gaussian Splatting and render spatial RGB panoramas at valid probe locations. PanoLight processes the rendered RGB panoramas together with their validity masks to recover HDR illumination observations. We represent the illumination of every scene using 100 virtual point lights, whose positions and linear RGB intensities are jointly optimized across the available probes. HDR reconstruction and VPL optimization are performed on a single NVIDIA A800 GPU, while the final interactive rendering is performed on an NVIDIA RTX 3060. All compared methods use the same inserted mesh, object transformation, camera views, material settings, output resolution, and composition pipeline for fair comparison.

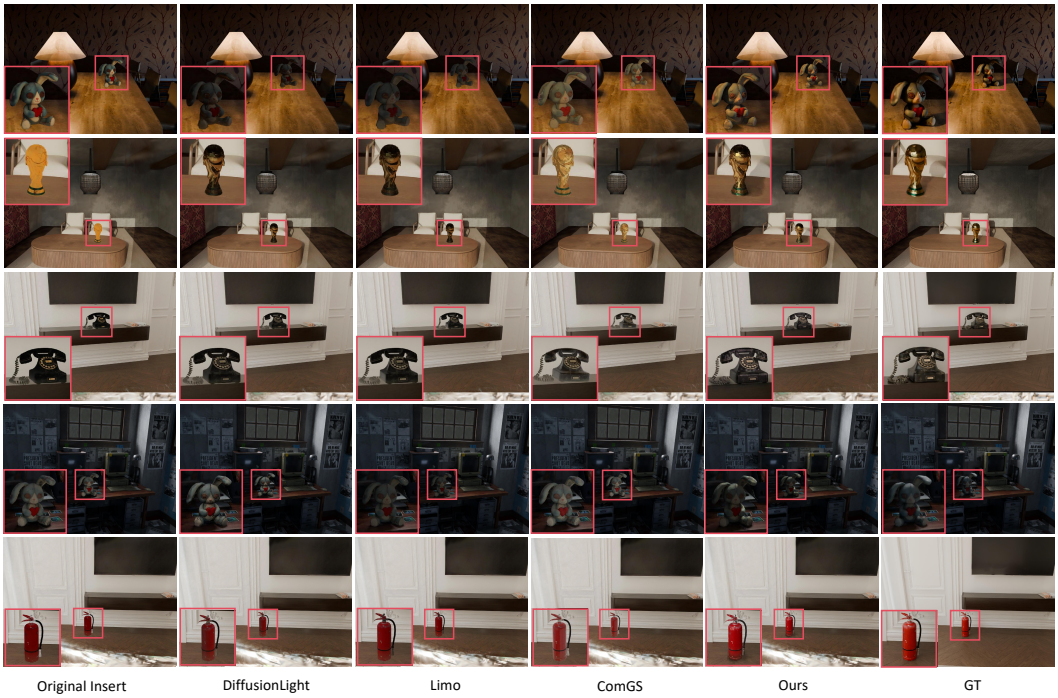

Figure 3: Qualitative comparison on the five paired benchmark scenes. The columns show Original Insert, DiffusionLight, LiMo, ComGS, Ours, and ground truth, respectively. Red boxes indicate the inserted object regions and the corresponding enlarged views. Our method more faithfully reproduces the local brightness, color, highlights, and shadow cues of the ground truth under different scene illumination conditions.

Compared Methods. We compare our method with four alternatives. **Original Insert** directly composites the mesh into the scene without scene-specific relighting, serving as a lower bound reference. **DiffusionLight** (Phongthawee et al., 2024) estimates an HDR environment map from a single scene observation. **LiMo** (Bolduc et al., 2026) recovers spatially varying HDR illumination from image sequences with geometric conditioning. **ComGS** (Gao et al., 2026) estimates scene illumination for object insertion from a 3D Gaussian representation. For a controlled comparison, all methods use the same inserted meshes, object poses, material settings, evaluation views, output resolution, and rendering protocol.

4.2 Comparison with Existing Methods

Quantitative Comparison. Table 1 reports the quantitative results on our paired benchmark. Our method achieves the best average performance across all three metrics, with a PSNR of 18.549 dB, an SSIM of 0.6372, and an LPIPS of 0.1970. Compared with the strongest competing method on average, LiMo, our method improves PSNR by 1.457 dB and SSIM by 0.0170, while reducing LPIPS by 0.0361. Our method also obtains the best PSNR in four of the five scenes and performs particularly well on Room Bunny and Bedroom Fire, where it ranks first across all metrics. Although individual baselines occasionally achieve better results on several metrics in Window Bunny, Room Cup, and Bedroom Telephone, their performances are unstable across scenes. In contrast, our method achieves the highest accuracy in most cases and delivers the most consistent overall performance, indicating that the estimated spatial illumination representation more faithfully reproduces the appearance of inserted objects under diverse lighting conditions.

Qualitative Comparison. Figure 3 compares methods on the paired benchmark. Direct insertion fails to match the original illumination, leading to incorrect brightness, color, and material appearance. DiffusionLight and LiMo improve overall illumination but often produce overly dark appearance or inconsistent highlights. ComGS recovers stronger illumination in some cases, but can

Table 1: Quantitative comparison on the inserted-object regions. Each value is averaged over six nearby training views. The Average columns report the mean over all five scenes. PSNR and SSIM are higher-is-better, while LPIPS is lower-is-better.

Scenes	**Window Bunny**			**Room Bunny**			**Room Cup**		
Methods	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Original Insert	25.338	0.8528	0.2039	15.308	0.6739	0.2606	10.844	0.4798	0.3570
DiffusionLight	26.724	0.8492	0.1965	14.527	0.6582	0.3095	13.945	0.5516	**0.2199**
LiMo	**27.330**	**0.8807**	**0.1823**	15.259	0.6641	0.3261	13.567	0.5493	0.2304
ComGS	25.970	0.8518	0.1987	15.472	0.6559	0.2957	11.962	0.4713	0.3139
Ours	26.405	0.8680	0.2061	**18.752**	**0.7121**	**0.2105**	**15.436**	**0.5680**	0.2304

Scenes	**Bedroom Telephone**			**Bedroom Fire**			**Average**		
Methods	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Original Insert	13.382	0.4441	0.2651	14.692	0.5640	0.1771	15.913	0.6029	0.2527
DiffusionLight	13.912	0.4724	0.2526	14.527	0.4945	0.2956	16.727	0.6052	0.2548
LiMo	13.938	**0.4727**	0.2456	15.367	0.5344	0.1811	17.092	0.6202	0.2331
ComGS	13.749	0.4463	0.2418	15.558	0.5330	0.1747	16.542	0.5917	0.2450
Ours	**14.102**	0.4436	**0.2085**	**18.051**	**0.5942**	**0.1294**	**18.549**	**0.6372**	**0.1970**

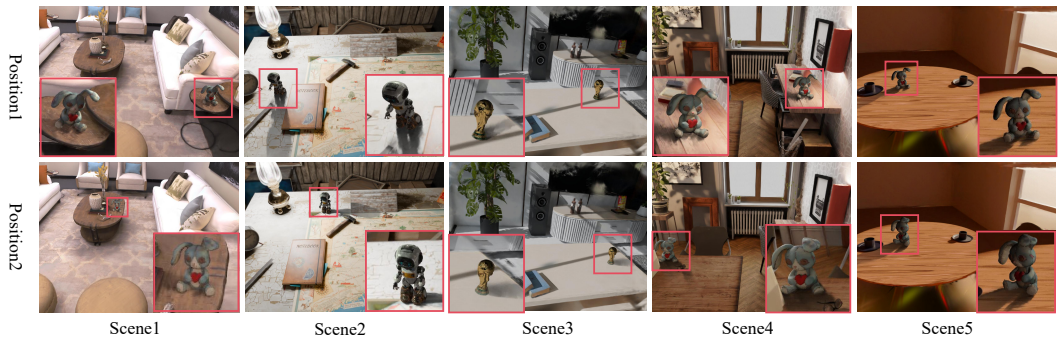

Figure 4: Qualitative evaluation of the generalization of our method across five scenes from three datasets. Each column shows a different scene, and the two rows show the inserted mesh at two positions. Red boxes and insets highlight the inserted object and its appearance in each scene. The results show that our method generalizes across diverse environments and object locations.

introduce excessive brightness or inaccurate colors. Our method better matches the ground truth across objects and lighting conditions, reproducing warm illumination on the bunny and trophy, material highlights on the telephone and fire extinguisher, and object visibility in dim scenes. Figure 4 further shows plausible relighting across five scenes from three datasets (Lanvin et al., 2026; Gao et al., 2026; Straub et al., 2019), with objects at different positions, demonstrating generalization across diverse scenes and locations.

4.3 Ablation

We evaluate mixed VPL sampling, semantic guidance, and shadow rendering by removing each component while keeping the remaining settings unchanged. Table 2 reports the average results on the five benchmark scenes, while Fig. 5 shows qualitative results at different object positions.

Removing mixed VPL sampling decreases PSNR from 18.549 to 17.650 and increases LPIPS from 0.1970 to 0.2214. This confirms that combining broad scene coverage, high radiance observations, and detected light sources provides a more representative illumination distribution. Without semantic guidance, the estimated VPLs are less strongly constrained by the physical light sources, which reduces performance across all three metrics. The effect is also visible in Fig. 5, particularly in the brightness and color response of the inserted objects.

Table 2: Quantitative ablation results averaged over the five benchmark scenes. The best results are shown in bold.

Methods	PSNR↑	SSIM↑	LPIPS↓
Original Insert	15.913	0.6029	0.2527
w/o Shadows	15.658	0.5391	0.3036
w/o Semantic Guidance	18.128	0.6301	0.2150
w/o Mixed VPLs	17.650	0.6270	0.2214
Ours	**18.549**	**0.6372**	**0.1970**

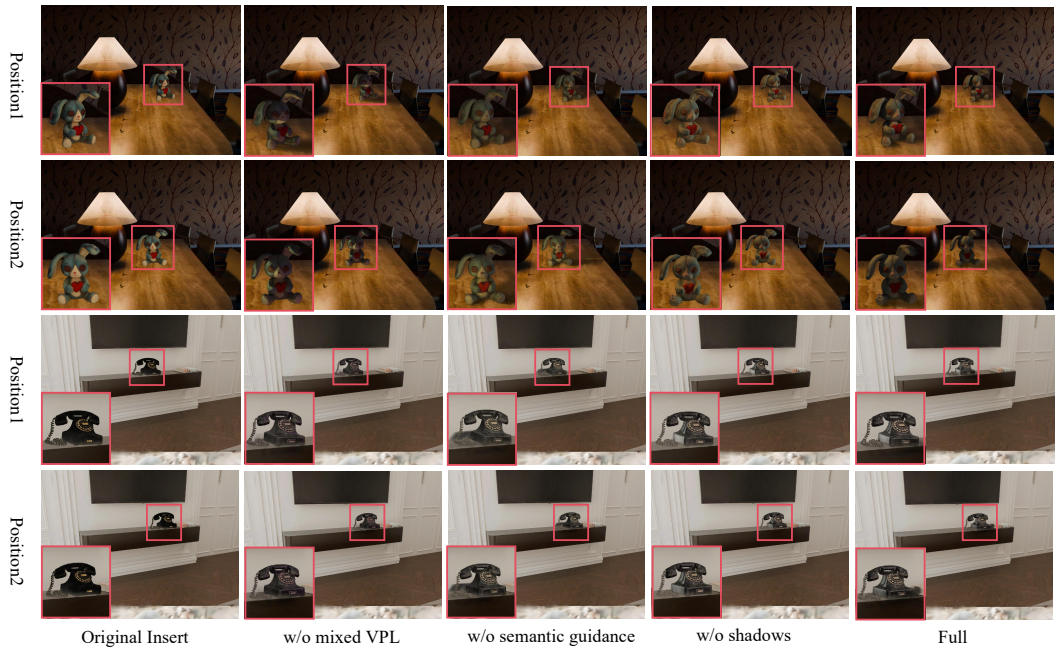

Figure 5: Qualitative ablation results at different object positions. Mixed VPL sampling and semantic guidance improve the local intensity and color response, while shadow rendering supplies geometric cues for object integration. The complete method produces the most consistent appearance across different positions.

Disabling shadow rendering causes the largest degradation, reducing PSNR and SSIM to 15.658 and 0.5391, respectively. Without shadows, the inserted objects lack important geometric cues relating them to the surrounding scene, leading to less plausible integration. In contrast, the complete method maintains consistent illumination and shadow appearance as the object position changes. It achieves the best overall performance, with a PSNR of 18.549, an SSIM of 0.6372, and an LPIPS of 0.1970.

5 Conclusion

We presented a framework for estimating a reusable spatial illumination representation from 3D Gaussian Splatting scenes for real-time object relighting. By optimizing virtual point lights and combining position conditioned illumination, reflection rendering, and shadow generation, our method produces scene-consistent appearances for inserted meshes. We also constructed a paired benchmark with object-free and object-present observations, enabling quantitative evaluation of relighting quality. Experiments and ablation studies demonstrate the effectiveness of the proposed components. One current limitation is that indirect illumination from the inserted object towards the original 3DGS scene, are not yet modeled. This may reduce the realism when strong light transport between the inserted object and the surrounding scene is present. Extending the framework to more complete scene relighting is left for future work.

REFERENCES

Christophe Bolduc, Yannick Hold-Geoffroy, and Jean-François Lalonde. Gaslight: Gaussian splats for spatially-varying lighting in hdr. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 29120–29130. IEEE, 2025.

Christophe Bolduc, Julien Philip, Li Ma, Mingming He, Paul Debevec, and Jean-François Lalonde. Lighting in motion: Spatiotemporal hdr lighting estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19696–19705, 2026.

Hongze Chen, Zehong Lin, and Jun Zhang. Gi-gs: Global illumination decomposition on gaussian splatting for inverse rendering. *arXiv preprint arXiv:2410.02619*, 2024.

Ho Kei Cheng, Seoung Wug Oh, Brian Price, Alexander Schwing, and Joon-Young Lee. Tracking anything with decoupled video segmentation. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1316–1326. IEEE, 2023.

Mohammad Reza Karimi Dastjerdi, Jonathan Eisenmann, Yannick Hold-Geoffroy, and Jean-François Lalonde. Everlight: Indoor-outdoor editable hdr lighting estimation. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7386–7395. IEEE, 2023.

Jian Gao, Mengqi Yuan, Yifei Zeng, Chang Zeng, Zhihao Li, Dave Chen, Weichao Qiu, Xiao-Xiao Long, Hao Zhu, Xun Cao, et al. Comgs: Efficient 3d object-scene composition via surface octahedral probes. In *International Conference on Learning Representations*, volume 2026, pp. 20941–20965, 2026.

Marc-André Gardner, Kalyan Sunkavalli, Ersin Yumer, Xiaohui Shen, Emiliano Gambaretto, Christian Gagné, and Jean-François Lalonde. Learning to predict indoor illumination from a single image. *arXiv preprint arXiv:1704.00090*, 2017.

Yijia Guo, Yuanxi Bai, Liwen Hu, Ziyi Guo, Mianzhi Liu, Yu Cai, Tiejun Huang, and Lei Ma. Prtgs: Precomputed radiance transfer of gaussian splats for real-time high-quality relighting. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 5112–5120, 2024.

Yannick Hold-Geoffroy, Kalyan Sunkavalli, Sunil Hadap, Emiliano Gambaretto, and Jean-François Lalonde. Deep outdoor illumination estimation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2373–2382. IEEE, 2017.

Tianyu Huang, Zhenyang Ren, Zhenchen Wan, Jiyang Zheng, Wenjie Wang, Runnan Chen, Mingming Gong, and Tongliang Liu. Lightharmony3d: Harmonizing illumination and shadows for object insertion in 3d gaussian splatting. *arXiv preprint arXiv:2603.29209*, 2026.

Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5322–5332. IEEE, 2024.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, et al. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

Shuichang Lai, Letian Huang, Jie Guo, Kai Cheng, Bowen Pan, Xiaoxiao Long, Jiangjing Lyu, Chengfei Lv, and Yanwen Guo. Glossygs: Inverse rendering of glossy objects with 3d gaussian splatting. *IEEE Transactions on Visualization and Computer Graphics*, 31(10):7478–7491, 2025.

Alexandre Lanvin, Jeffrey Hu, Simon Lucas, Adrien Bousseau, and George Drettakis. Intrinsic decomposition and editing of 3d gaussian splats. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 9(1):1–18, 2026.

Zhengqin Li, Li Yu, Mikhail Okunev, Manmohan Chandraker, and Zhao Dong. Spatiotemporally consistent hdr indoor lighting estimation. *ACM Transactions on Graphics*, 42(3):1–15, 2023.

Ruofan Liang, Kai He, Zan Gojcic, Igor Gilitschenski, Sanja Fidler, Nandita Vijaykumar, and Zian Wang. Luxdit: Lighting estimation with video diffusion transformer. *Advances in Neural Information Processing Systems*, 38:5015–5046, 2026.

Zhihao Liang, Qi Zhang, Ying Feng, Ying Shan, and Kui Jia. Gs-ir: 3d gaussian splatting for inverse rendering. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21644–21653. IEEE, 2024.

Victor Ostromoukhov, Charles Donohue, and Pierre-Marc Jodoin. Fast hierarchical importance sampling with blue noise properties. *ACM Trans. Graph.*, 23(3):488–495, August 2004. URL https://doi.org/10.1145/1015706.1015750.

Pakkapon Phongthawee, Worameth Chinchuthakun, Nontaphat Sinsunthithet, Varun Jampani, Amit Raj, Pramook Khungurn, and Supasorn Suwajanakorn. Diffusionlight: Light probes for free by painting a chrome ball. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 98–108. IEEE, 2024.

Kerui Ren, Jiayang Bai, Linning Xu, Lihan Jiang, Jiangmiao Pang, Mulin Yu, and Bo Dai. Mvcolight: Efficient object compositing with consistent lighting and shadow generation. *Advances in Neural Information Processing Systems*, 38:170882–170902, 2026.

Saptarshi Neil Sinha, Julius Kühn, Holger Graf, and Michael Weinmann. Spectralsplatsviewer: An interactive web-based tool for visualizing cross-spectral gaussian splats. In *Proceedings of the 29th International ACM Conference on 3D Web Technology*, pp. 1–10, 2024.

Vsevolod Skorokhodov, Nikita Durasov, and Pascal Fua. D3dr: Lighting-aware object insertion in gaussian splatting. *arXiv e-prints*, pp. arXiv–2503, 2025.

Gowri Somanath and Daniel Kurz. Hdr environment map estimation for real-time augmented reality. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11298–11306, 2021.

Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

Hanxiao Sun, Yupeng Gao, Jin Xie, Jian Yang, and Beibei Wang. Svg-ir: Spatially-varying gaussian splatting for inverse rendering. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16143–16152. IEEE, 2025.

Xuechang Tu, Lukas Radl, Michael Steiner, Markus Steinberger, Bernhard Kerbl, and Fernando De la Torre. Vrsplat: Fast and robust gaussian splatting for virtual reality. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 8(1):1–22, 2025.

Liang Wan, Tien-Tsin Wong, and Chi-Sing Leung. Spherical q2-tree for sampling dynamic environment sequences. In *Proceedings of the Sixteenth Eurographics Conference on Rendering Techniques*, EGSR '05, pp. 21–30, 2005.

Jun-Peng Xu, Chenyu Zuo, Fang-Lue Zhang, and Miao Wang. Rendering-aware hdr environment map prediction from a single image. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 2857–2865, 2022.

Keyang Ye, Qiming Hou, and Kun Zhou. 3d gaussian splatting with deferred reflection. In *ACM SIGGRAPH 2024 conference papers*, pp. 1–10, 2024.

Tony Yu, Yanlin Jin, Ashok Veeraraghavan, Akshat Dave, and Guha Balakrishnan. Transplat: Lighting-consistent cross-scene object transfer with 3d gaussian splatting. *arXiv e-prints*, pp. arXiv–2503, 2025.

Hongjia Zhai, Xiyu Zhang, Boming Zhao, Hai Li, Yijia He, Zhaopeng Cui, Hujun Bao, and Guofeng Zhang. Splatloc: 3d gaussian splatting-based visual localization for augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 31(5):3591–3601, 2025.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

Iris Zheng, Guojun Tang, Alexander Doronin, Paul Teal, and Fang-Lue Zhang. Ssd-gs: Scattering and shadow decomposition for relightable 3d gaussian splatting. *arXiv preprint arXiv:2604.13333*, 2026.

A Appendix

A.1 VPL Fitting Objective

Let $\mathbf{q}_p$ denote a probe position, $\boldsymbol{\omega}_u$ a panorama direction, and $\mathbf{H}_{pu}$ its RGB HDR observation. Each VPL has position $\mathbf{x}_i$ and nonnegative RGB intensity $\mathbf{I}_i = \mathrm{softplus}(\mathbf{z}_i)$. The fitting renderer predicts

$$\widehat{\mathbf{H}}_{pu} = \sum_i \frac{\mathbf{I}_i}{\|\mathbf{x}_i - \mathbf{q}_p\|_2^2 + \epsilon^2} K_\kappa(\boldsymbol{\omega}_u^\top \mathbf{d}_{pi}), \tag{1}$$

$$\mathbf{d}_{pi} = \frac{\mathbf{x}_i - \mathbf{q}_p}{\|\mathbf{x}_i - \mathbf{q}_p\|_2}, \qquad K_\kappa(t) = \frac{\kappa e^{\kappa(t-1)}}{2\pi(1 - e^{-2\kappa})}. \tag{2}$$

Here $\epsilon = 0.35$ and $\kappa = 1/\sigma^2$, with σ in radians. This angular kernel is used for HDR fitting; it is distinct from the distance-falloff settings of the interactive renderer.

Define $f(\mathbf{v}) = \log(1 + 3\mathbf{v})$ componentwise and $c_{pu} = \eta + (1 - \eta)m_{pu}$, where m_{pu} is the known-region confidence. Let s_u denote pixel solid angle and r_{pu} the semantic source-region weight. We use

$$\mathcal{L}_{\mathrm{photo}} = \frac{\sum_{pu} c_{pu} s_u \|f(\widehat{\mathbf{H}}_{pu}) - f(\mathbf{H}_{pu})\|_1}{3 \sum_{pu} c_{pu} s_u}, \tag{3}$$

$$\mathcal{L}_{\mathrm{region}} = \frac{\sum_{pu} c_{pu} s_u r_{pu} \|f(\widehat{\mathbf{H}}_{pu}) - f(\mathbf{H}_{pu})\|_1}{3 \sum_{pu} c_{pu} s_u r_{pu}}, \tag{4}$$

$$\mathcal{L}_{\mathrm{under}} = \frac{\sum_{pu} c_{pu} s_u r_{pu} \|[f(\mathbf{H}_{pu}) - f(\widehat{\mathbf{H}}_{pu})]_+\|_1}{3 \sum_{pu} c_{pu} s_u r_{pu}}. \tag{5}$$

Denominators are clamped away from zero in the implementation. The source-region terms emphasize emitter observations and penalize underestimation of their radiance.

Let $y_i = (0.2126, 0.7152, 0.0722)^\top \mathbf{I}_i$ and $\boldsymbol{\chi}_i = \mathbf{I}_i / (\sum_c I_{ic})$. For source memberships a_{ik}, define $a_i = \max_k a_{ik}$. The semantic energy and color losses are

$$\mathcal{L}_{\mathrm{energy}} = \left(\frac{\sum_i a_i y_i}{\sum_i y_i} - \rho \right)^2, \tag{6}$$

$$\mathcal{L}_{\mathrm{color}} = \frac{\sum_i a_i \,\mathrm{sg}(y_i) \|\boldsymbol{\chi}_i - \overline{\boldsymbol{\chi}}_i\|_2^2}{3 \sum_i a_i \,\mathrm{sg}(y_i)}. \tag{7}$$

Here sg stops gradients and $\overline{\boldsymbol{\chi}}_i$ is the normalized membership-weighted blend of estimated source chromaticities. Point-source membership is Gaussian in spatial distance; window membership combines distance to the plane with its bounded extent.

Environment weights are $b_i = [1 - a_i/0.3]_0^1 \,\mathrm{sg}(y_i)$. With anchor chromaticity prior $\boldsymbol{\chi}_i^0$ and threshold t,

$$\mathcal{L}_{\mathrm{env}} = \frac{\sum_i b_i \|\boldsymbol{\chi}_i - \boldsymbol{\chi}_i^0\|_2^2}{3 \sum_i b_i}, \tag{8}$$

$$\mathcal{L}_{\mathrm{sat}} = \frac{\sum_i b_i [\max_c \chi_{ic} - \min_c \chi_{ic} - t]_+^2}{\sum_i b_i}. \tag{9}$$

The anchor prior is blended toward neutral chromaticity before fitting. The saturation quantity above is a chromaticity range, not HSV saturation.

For anchors $\mathbf{x}_i^0$ and global displacement parameter D,

$$\mathcal{L}_{\mathrm{anchor}} = \frac{\sum_i \|\mathbf{x}_i - \mathbf{x}_i^0\|_2^2}{3ND^2}, \tag{10}$$

$$\mathcal{L}_{\mathrm{rep}} = \frac{1}{N^2} \sum_{i \neq j} \exp[-(\|\mathbf{x}_i - \mathbf{x}_j\|_2 / 0.45)^2], \tag{11}$$

$$\mathcal{L}_{\mathrm{sparse}} = \frac{1}{3N} \sum_{ic} I_{ic}. \tag{12}$$

Parameter	Room 1	Window Bunny	Room 3	Bedroom
VPL count	150	100	150	150
Coverage / importance / semantic	105/30/15	70/20/10	105/30/15	30/30/90
Angular width (degrees)	18	12	12	12
Inpaint weight η	0.25	0.30	0.30	0.30
Surface offset	0.02	0.08	0.08	0.08
Global displacement D	0.75	0.45	0.45	0.45
Anchor weight λ_a	0.03	0.02	0.02	0.02
Source energy target ρ	0.68	0.68	0.68	0.68
Energy weight λ_e	1.2	1.2	1.2	1.2
Color weight λ_c	1.4	1.4	1.4	1.8
Attribution weight λ_t	0	0.8	0	0.8
Environment color λ_b	1.6	1.6	0	1.6
Saturation weight λ_s	1.2	1.2	0	1.2
Saturation threshold t	0.18	0.22	0.22	0.22
Neutral prior blend	0.45	0.45	0.25	0.45

Table 3: Recorded fitting settings. Bedroom denotes the shared scene field used by the Telephone and Fire configurations.

Offsets are parameterized as $D_i\mathbf{o}_i/(1+\|\mathbf{o}_i\|_2)$. Semantic window samples are projected onto and clamped to their window rectangles. Coverage and semantic displacement limits are 0.18 and 0.10, respectively; importance samples use D. Distances are in reconstruction units.

For each semantic group G_k containing at least two samples, we penalize energy collapse using

$$\mathcal{L}_{\text{spread}} = \frac{1}{|\mathcal{G}|}\sum_{G_k\in\mathcal{G}}\frac{1}{|G_k|}\sum_{i\in G_k}\left(\frac{y_i}{\sum_{j\in G_k} y_j}-\frac{1}{|G_k|}\right)^2. \tag{13}$$

For the enabled point-source attribution term $\mathcal{L}_{\text{attr}}$, we render each semantic point-source group separately and fit its HDR excess above a local angular background using solid-angle-weighted log-space L1. The background is the per-channel median in the source-mask annulus $[0.015, 0.12]$ over sufficiently known pixels; if fewer than 16 pixels are available, the support expands to mask values at most 0.12. The target is $[\mathbf{H}-\mathbf{B}]_+$. Weights are squared source masks times an active-region indicator (mask at least 0.12, known confidence at least 0.15, excess luminance above 10^{-5}), multiplied by excess luminance normalized by its active-region 75th percentile and clipped to $[0,2]$. Losses are averaged over valid point-source groups.

The total objective is

$$\begin{aligned}\mathcal{L} = \mathcal{L}_{\text{photo}} &+ \lambda_r\mathcal{L}_{\text{region}} + \lambda_u\mathcal{L}_{\text{under}} + \lambda_a\mathcal{L}_{\text{anchor}} + 0.004\mathcal{L}_{\text{rep}} + 0.0004\mathcal{L}_{\text{sparse}} \\ &+ \lambda_e\mathcal{L}_{\text{energy}} + \lambda_c\mathcal{L}_{\text{color}} + 0.03\mathcal{L}_{\text{spread}} + \lambda_t\mathcal{L}_{\text{attr}} + \lambda_b\mathcal{L}_{\text{env}} + \lambda_s\mathcal{L}_{\text{sat}} + \mathcal{L}_{\text{window}}.\end{aligned} \tag{14}$$

These terms have the same forms as the semantic energy and color losses.

All four fitted scene fields use 1,200 Adam iterations. Bedroom Telephone and Bedroom Fire share the same fitted field.

A.2 Benchmark Construction

We construct a paired synthetic benchmark for evaluating 3D object insertion and relighting. The benchmark comprises five scene–object configurations assembled from freely available Blender assets: Blender Room 1, Window Bunny, Room 3, Bedroom Telephone, and Bedroom Fire. These configurations include objects with different material properties and indoor scenes with different lighting conditions.

For each configuration, we render two versions of the scene, with and without the target object, using identical camera poses, camera intrinsics, lighting, and rendering settings. Each configuration contains 150–200 paired viewpoints at a resolution of 1920×1440. The two versions share the same camera projection and background geometry, while allowing physically induced appearance changes, such as cast shadows, reflections, and indirect illumination, caused by the inserted object.



Figure 6: Example scenes from our paired benchmark. From left to right and top to bottom: bedroom telecom, room cup, room bunny, and bedroom fire. Within each scene, paired images show the scenes without (left) and with (right) the inserted object.

All reference images are rendered using Blender Cycles with 500 samples per pixel, a maximum of eight light bounces, and denoising enabled. The object-free images are used to reconstruct the Gaussian scene and estimate its illumination. The corresponding object-present images serve as reference renderings for evaluating the inserted object's appearance. We insert the corresponding mesh into the reconstructed scene and align its placement, orientation, and scale with those of the reference object.

For quantitative evaluation, we select six nearby training viewpoints for each configuration and render all compared methods using the same cameras and object placement. We compute PSNR, SSIM, and LPIPS within the inserted-object regions using a consistent region-of-interest protocol across methods. Per-configuration scores are averaged over the six views, and the overall scores are computed as the unweighted mean over the five configurations. This protocol evaluates insertion appearance at observed viewpoints rather than novel-view generalization.

Figure 6 presents four representative configurations. Each pair shows the same viewpoint without and with the target object, illustrating the paired observations used for scene reconstruction and reference evaluation, respectively.

A.3 Algorithm Overview

Algorithm 1 summarizes our pipeline, which consists of offline lighting estimation and online object relighting. In the offline stage, we localize semantic light sources, estimate HDR illumination at spatial probes, and optimize a compact VPL field under semantic and photometric constraints. In the online stage, the VPL field provides position-dependent illumination for a local environment cubemap, enabling material-aware shading as the inserted object moves through the scene. A nearby

Algorithm 1 Semantic-Guided Lighting Estimation and Object Insertion

Require: Reconstructed Gaussian scene $\mathcal{G}$, calibrated images $\mathcal{I}$, insertion mesh $\mathcal{M}$, VPL budgets (N_c, N_i, N_s), optimization steps T
Ensure: Optimized VPL field $\mathcal{V}$ and relit insertion images

Offline lighting estimation
Detect candidate light sources using Qwen-VL.
Track source masks across views using DEVA.
Recover semantic source geometry using calibrated views: point positions for lamps and bounded planes for windows.
Partition the scene into a $3 \times 3 \times 3$ grid and select valid probe positions.
Render GS panoramas at the probes and estimate spatial HDR maps using PanoLight.
Initialize N_c coverage, N_i importance, and N_s semantic VPLs.
Associate semantic VPLs with the recovered light sources.
for $t = 1, \ldots, T$ **do**
    Predict probe radiance from the current VPL field.
    Evaluate HDR reconstruction and source-region losses.
    Add semantic energy/color constraints and spatial/color regularization.
    Update VPL positions and nonnegative RGB intensities.
    Enforce semantic window-plane constraints.
end for
Apply source/environment energy calibration and export $\mathcal{V}$.
Online object insertion
for each frame **do**
    Update the mesh pose and its local lighting probe.
    Inject VPL radiance into the local environment cubemap.
    Evaluate diffuse and specular mesh shading using the material parameters.
    Select the nearest semantic-source VPL as the representative shadow caster.
    Compute shadows and composite the shaded mesh with the Gaussian scene.
end for
return $\mathcal{V}$ and the rendered insertion images.

semantic-source VPL is selected as the representative shadow caster to support interactive shadow rendering.

A.4 AI USE STATEMENT

AI-based tools were used during this work to assist with language refinement and code development. Specifically, they supported sentence restructuring, grammar and wording improvements, and enhancements to the clarity and readability of the manuscript, as well as code implementation and debugging. The authors are responsible for reviewing and verifying all AI-assisted content and code, and take full responsibility for the methodology, experimental results, and conclusions presented in this work.