
MV-CoLight: Efficient Object Compositing with Consistent Lighting and Shadow Generation

- *Supplementary Material*

Kerui Ren^{1,2} Jiayang Bai³ Linning Xu⁴ Lihan Jiang^{2,5}
Jiangmiao Pang² Mulin Yu^{2*} Bo Dai^{6*}

¹Shanghai Jiao Tong University, ²Shanghai Artificial Intelligence Laboratory,

³Nanjing University, ⁴The Chinese University of Hong Kong,

⁵University of Science and Technology of China, ⁶The University of Hong Kong

A Supplementary Material

In the supplementary material, we first present a brief overview of the core concepts underlying Gaussian splatting [6] and the Hilbert curve [4] in Sec. A.1. Subsequently, Sec. A.2 elaborates in detail on the dataset construction process and preprocessing procedures. Furthermore, the details of our object compositing models are elaborated in Sec. A.3, along with the reason why we select this specific model architecture. In Sec. A.4 we analyze the trade-off of the time required for scene reconstruction and the feasibility of using Anysplat [5]. In Sec. A.5 and Sec. A.6, we present a user study and additional qualitative results demonstrating objectively and subjectively outcomes in both single-view and multi-view object compositing across multiple public datasets and our proposed dataset. We further showcase multi-view visualization for illuminative object insertion. Sec. A.7 illustrates the effectiveness of different training stages of the model and analyzes the training dynamics of the 2D compositing model. Moreover, for unposed inputs, we present multi-view object insertion results based on camera poses and depth maps provided by VGGT [8], as shown in Sec. A.8.

A.1 Preliminaries

Gaussian Splatting 3D Gaussian splatting [6] models 3D scenes using anisotropic Gaussian primitives, employing a projection-based rasterization process to generate photorealistic renderings. Each primitive is mathematically represented by a multivariate Gaussian distribution parameterized as:

$$G(\mathbf{x}) = \exp\left(-\frac{1}{2}(\mathbf{x} - \mu)^\top \Sigma^{-1}(\mathbf{x} - \mu)\right) \quad (1)$$

where $\mu \in \mathbb{R}^3$ specifies the spatial centroid and $\Sigma \in \mathbb{R}^{3 \times 3}$ denotes the covariance matrix. For geometric interpretability, the covariance matrix is factorized into rotation and scaling components through the decomposition $\Sigma = R S S^\top R^\top$, where $R \in \text{SO}(3)$ represents the rotation matrix and $S = \text{diag}(s_x, s_y, s_z)$ encodes axis-aligned scaling factors.

During differentiable rendering, each Gaussian primitive is augmented with additional attributes, an opacity coefficient $\sigma \in [0, 1]$ controlling light transmittance and a spherical harmonics $F \in \mathbb{R}^C$ enable view-dependent color estimation $c \in \mathbb{R}^3$ through directional decoding. The rendering process employs a tile-based rasterization pipeline that first performs efficient depth sorting of Gaussians in camera-facing order, followed by perspective projection to transform 3D Gaussian distributions

*Corresponding author.

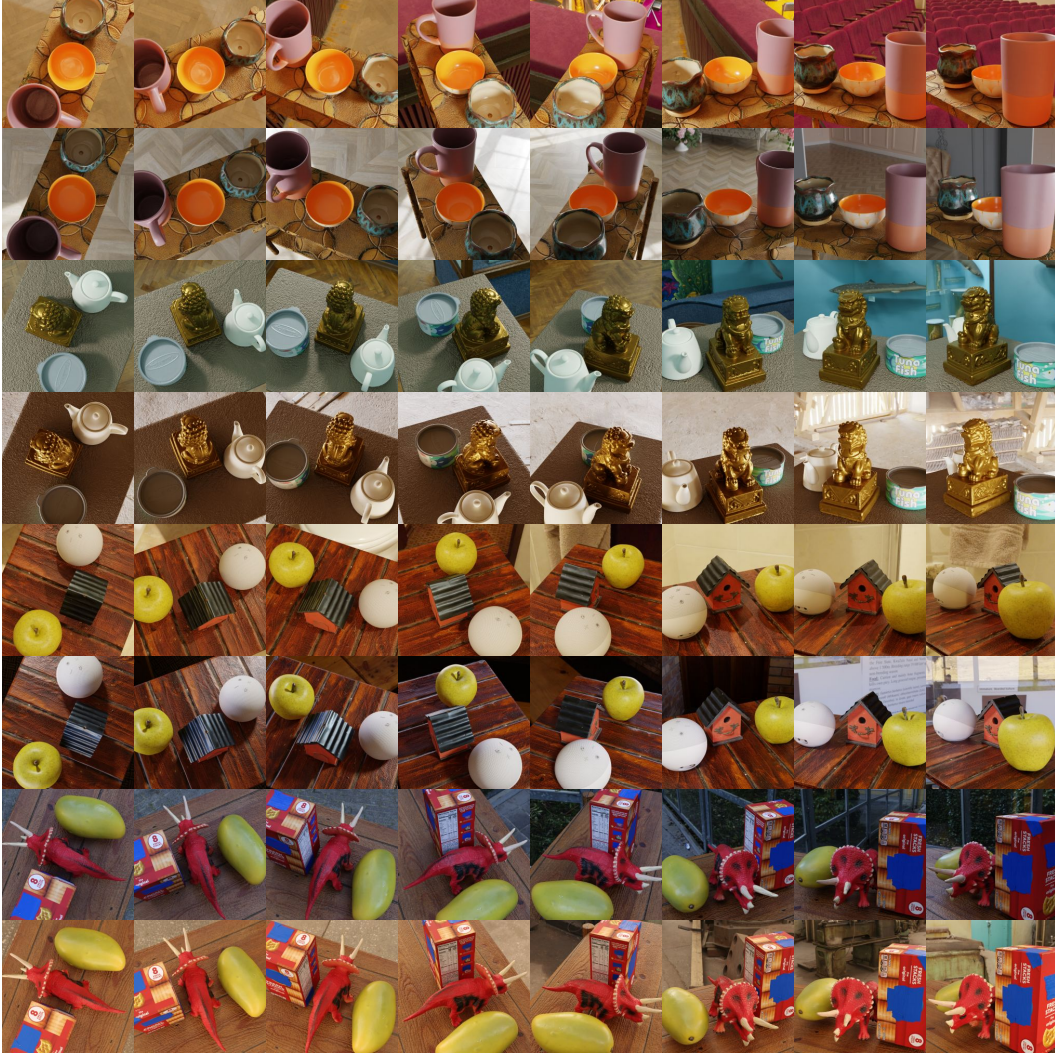


Figure 1: Visualization of the DTC-MultiLight dataset. We showcase rendered results of diverse scenes created using objects from the DTC dataset within the Blender engine, highlighting multi-view perspectives and varying lighting conditions.

into 2D image-plane counterparts $G'(\mathbf{x}')$, and finally executes per-pixel α -compositing through the rendering equation:

$$C(\mathbf{x}') = \sum_{i \in \mathcal{N}} T_i c_i \alpha_i, \quad \alpha_i = \sigma_i G'_i(\mathbf{x}') \quad (2)$$

where x' is the queried pixel, \mathcal{N} represents depth-sorted sequence of 2D Gaussians associated with x' and T denotes the cumulative transmittance term as $\prod_{j=1}^{i-1} (1 - \alpha_j)$.

Hilbert Curve the Hilbert curve[4] is a space-filling curve that maps multidimensional data to a one-dimensional sequence while preserving locality. The Hilbert curve recursively partitions the space, ensuring that points close in the multidimensional domain remain near each other in the one-dimensional ordering. This locality preservation is critical for maintaining the spatial coherence of Gaussian primitives during their projection onto a 2D image space.

A.2 Dataset Curation

We present a large-scale synthetic dataset, named **DTC-MultiLight**, specifically designed for consistent object compositing in 3D scenes, comprising about 480,000 procedurally generated scenes with systematically varied scene components and illumination conditions, as shown in Fig. 1. This comprehensive dataset, created using Blender’s rendering engine, serves as a robust benchmark for both training and evaluating object compositing models across diverse object placements and illumination settings.

When constructing object repositories for scene composition, we select the Digital Twin Catalog (DTC) dataset [2] over the widely adopted Objaverse [1] due to its superior scanning accuracy, which enables more physically accurate simulations of lighting interactions and shadow casting. From this dataset, we curate four distinct tables serving as placement surfaces, complemented by 1,752 diverse household objects. However, we observe that the original table surface materials exhibit insufficient albedo values for clear shadow visualization. To address this visual limitation, we implement 69 distinct material properties from Poly Haven (primarily wood and concrete textures). This material diversification strategy serves to enhance dataset variability while improving the model’s generalizability to real-world surface reflectance conditions. However, we observe that the original table surface materials exhibit insufficient albedo values for clear shadow visualization. To address this visual limitation, we implement 69 different physically-based material presets from Poly Haven (encompassing wood and concrete textures). This material diversification strategy serves to enhance dataset variability while improving the model’s generalizability to real-world surface reflectance conditions. To establish photorealistic illumination conditions and enhance contextual background elements, we integrate 207 indoor high-dynamic-range (HDR) environment maps sourced from Poly Haven². Recognizing the inherent limitations of environment maps in generating sharp shadow boundaries, we strategically augment scenes with supplementary light sources to achieve enhanced lighting variation and directional shadow effects.

During the Blender rendering process, we randomly select three distinct objects, placing them on the table in a random arrangement. Under the illumination of a random environment map and additional light sources, we render 16 images from specific perspectives centered on the objects, striving to cover the scene comprehensively, as shown in Fig. 1. Furthermore, we intentionally assign emissive material properties to the inserted objects in a subset of our dataset, enabling these luminous entities to physically influence the ambient illumination conditions of background scenes, augmenting the diversity and challenging of our dataset. Ultimately, we develop a comprehensive multi-view object compositing dataset featuring varied illumination scenarios, heterogeneous object categories, and diverse background materials. This versatile dataset demonstrates broad applicability across multiple computer vision domains, such as multi-view object compositing, scene relighting and scene generation.

To simulate object insertion, we first randomly select two sets of multi-view images captured under different lighting conditions for the same scene. We then use mask maps to separate each image into foreground and background, where the foreground contains the same object across both sets. Finally, we merge the foreground and background from the same viewpoint but under different lighting conditions to simulate object insertion. The mask maps are generated by rendering the scenes in Blender, where the foreground object and background are assigned distinct colors to facilitate their separation.

A.3 Detailed Model Architecture

We have provided a detailed model architecture figure, as shown in Fig. 2. For the 2D object compositing model, we built upon the original Swin-Transformer architecture by adding CNN-based feature extraction and output modules to align the input/output dimensionalities. Additionally, we incorporated residual modules in the feature space to stabilize the model’s learning of scene harmonization. Subsequently, we utilized multiple Swin Transformer blocks to extract color attributes, illumination properties, and spatial shadow relationships from shallow features. For the 3D object compositing model, we projected the output features of the 2D model into a Gaussian color-aligned 2D space, fused inconsistent Gaussian color representations during the encoding phase, and employed

²<https://polyhaven.com/>

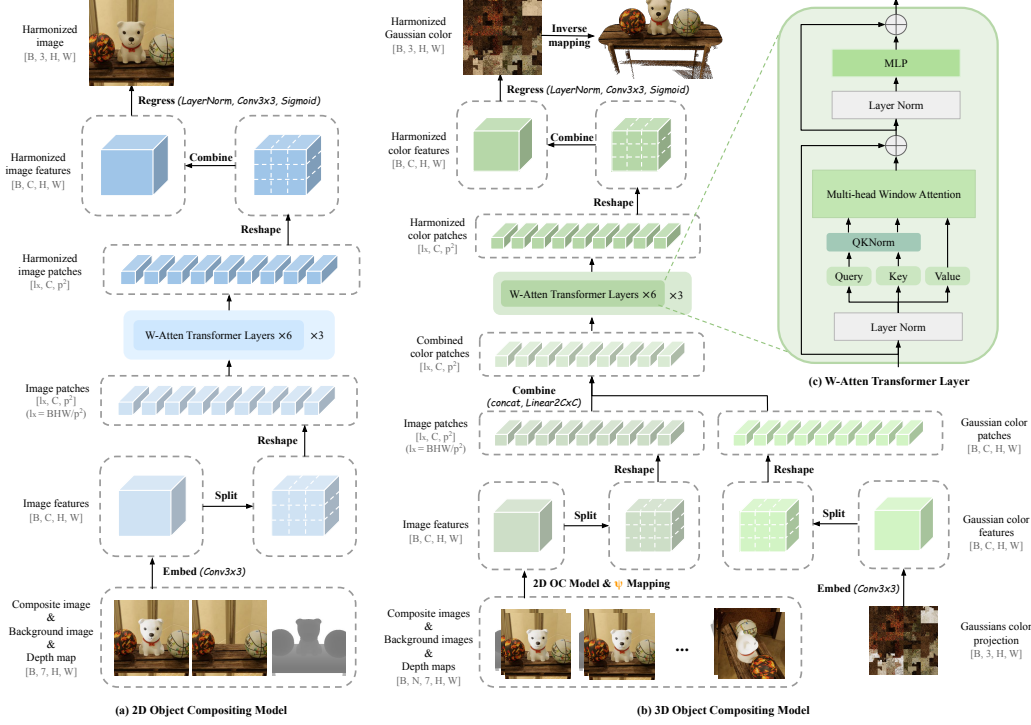


Figure 2: Detailed model architecture of our compositing models. We introduce two compositing models: (a) The 2D object compositing model encodes input data into shallow feature space, which is then partitioned into $p \times p$ size patches. These features are decoded through a Swin Transformer-based decoder to produce harmonized output. (b) The 3D object compositing model encodes the output features from the 2D model and inharmonious Gaussian colors. Through a similar process, it generates harmonized Gaussian colors and reprojects them into the Gaussian model. Both models employ pure window-attention layers as shown in (c).

Swin Transformer blocks with identical structures to address 3D scene harmonization. Finally, the Gaussian color outputs from the model were back-projected into the Gaussian space.

By choosing the more efficient Swin Transformer over the original ViT, we accelerated both training and inference speeds while increasing the upper limits for input quantity and resolution. The local attention mechanism further directs focused attention to detailed highlights and shadow generation. As shown in Figure 8, our method even captures fine-grained texture shadows on inserted objects.

A.4 Time Consumption of Gaussian Initialization

Given the constructed 3D scene representation, our framework can rapidly harmonize any number of viewpoints or even newly inserted objects (e.g., in 0.07 seconds per frame). Thus, the reconstruction cost is analogous to loading or preparing a 3D asset, after which real-time interaction becomes feasible. To balance reconstruction time against quality, we conduct an ablation study on training iterations. We are also exploring how feed-forward 3D reconstruction can accelerate this process. Recent geometry foundation models generate Gaussian representations directly from sparse inputs in a single forward pass. To validate this, we experiment with AnySplat [5], which constructs Gaussian models from input frames without per-scene optimization, significantly reducing initial setup time.

The results in Tab. 1 reveal a clear trade-off between reconstruction paradigm, speed, and scene complexity. AnySplat [5] achieves the fastest reconstruction time at under one second and delivers quality nearly on par with the highest-quality, per-scene optimized 3DGS in complex, forward-facing scenes. However, its performance significantly degrades in simple, surround-view scenes, likely due to the inherent challenge of jointly estimating camera poses. Meanwhile, by reducing training iterations of the vanilla 3DGS to 1k, it produces comparable results in just a few seconds, offering a

Table 1: Quantitative analysis and time consumption of Gaussian initialization with diverse methods.

Method	Simple Synthetic Scene			Complex Synthetic Scene			Real Captured Scene			Time(s)
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	
3DGS(#10k iter)	30.29	0.960	0.030	30.13	0.952	0.027	26.39	0.927	0.040	64.8
3DGS(#1k iter)	28.43	0.926	0.076	28.54	0.895	0.092	25.24	0.899	0.081	6.5
3DGS(#250 iter)	26.50	0.892	0.115	27.16	0.869	0.126	22.03	0.853	0.148	1.6
AnySplat	22.28	0.705	0.211	30.03	0.935	0.050	24.46	0.864	0.095	0.8

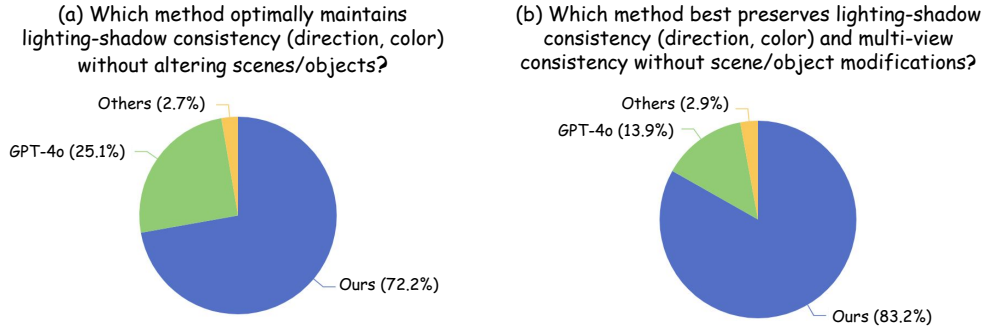


Figure 3: User Study Results. We respectively compare our method with baseline methods for both single-view object compositing and multi-view object compositing to quantify the realism in lighting and shadow generation. Results demonstrate that our method outperforms baseline methods.

practical compromise between the instant inference of AnySplat and the lengthy optimization of a full 10k-iteration run.

A.5 User Study

The user study was conducted via an online questionnaire and we collected 374 valid responses from human participants. The study presented participants with side-by-side comparisons and asked them to choose which result demonstrated more realistic and consistent lighting and shadows. The two questions corresponded to the single-view and multi-view compositing tasks, respectively. Our method surpasses all baseline methods under objective criteria, as demonstrated in Fig. 3.

A.6 Additional Visual Results

We provide additional visualizations comprising single-view object compositing (Fig. 4, 5), multi-view object compositing (Fig. 6), and multi-view light insertion (Fig. 7). Comprehensive qualitative comparisons substantiate the superiority and robustness of our approach.

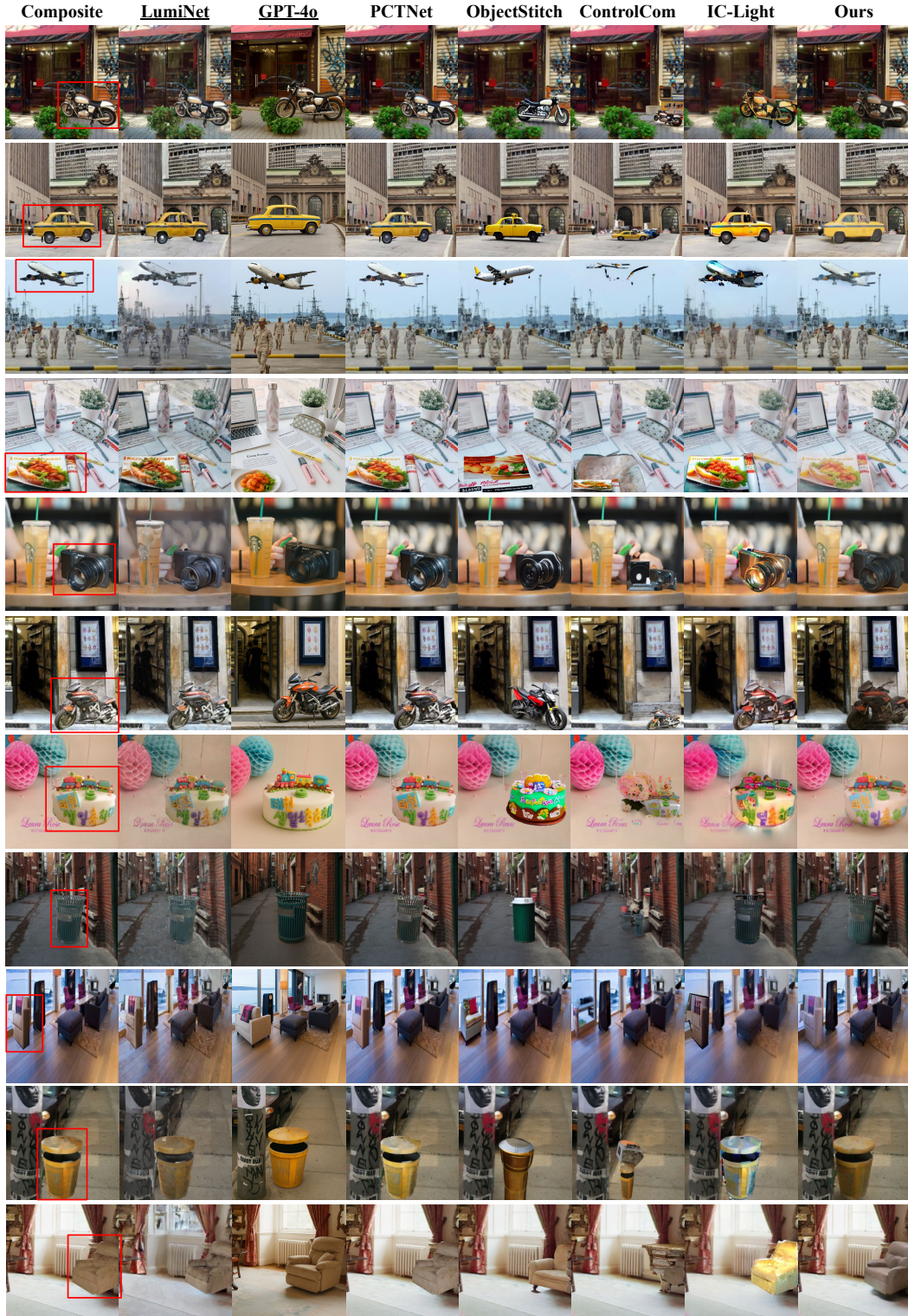


Figure 4: Single-view qualitative comparison on the Foscom dataset [9]. Our method not only performs color harmonization but also generates realistic highlights and shadows, providing more visually convincing results compared to baseline methods such as PCTNet [3], which focus solely on color harmonization. The methods do not require background image input, while others include.



Figure 5: Single-view qualitative comparison on the Object with Lighting dataset [7] and our proposed dataset. Our approach achieves implicit lighting disentanglement for inserted objects, synthesizing spatially consistent illumination and shadows that adaptively align with background lighting conditions. Unlike baseline methods, our framework produces photorealistic lighting effects surpassing existing approaches in both object-centric simple scenes and indoor complex environments, while strictly preserving the original scene geometry, scale, and object positioning. The methods do not require background image input, while others include.

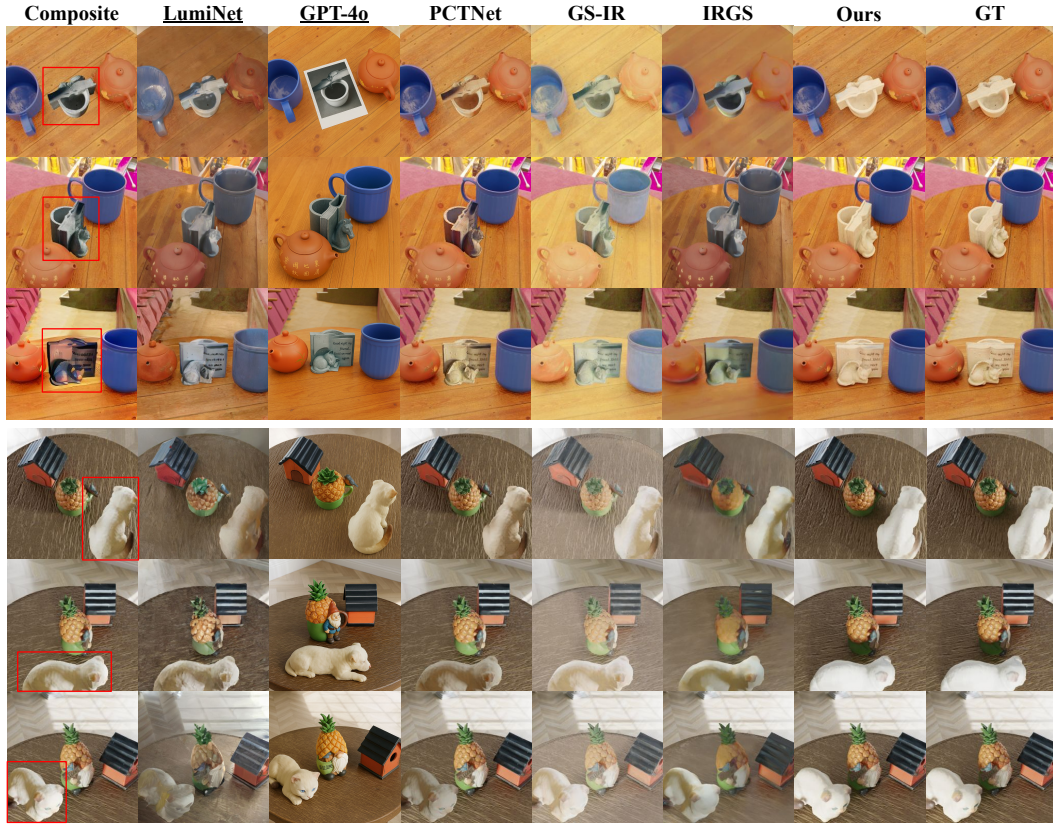


Figure 6: Multi-view qualitative comparison on our rendered scenes. In the first case, we remove existing shadows on the inserted object. In the second case, while eliminating shadows, we generate new shadows on the left side of the kitten and corresponding desktop based on the top-right light source in the scene. In addition, our method achieves multi-view consistency, whereas those methods that focus on image-level tasks exhibit color and shape discrepancies. Gaussian-based inverse rendering, constrained by environment map inputs, produces overly bright or dark visual artifacts. The methods do not require background image input, while others include.



Figure 7: Multi-view visualization after light source insertion. Our method meticulously simulates the emission effects of inserted light sources, their illumination on surrounding objects, and shadows.

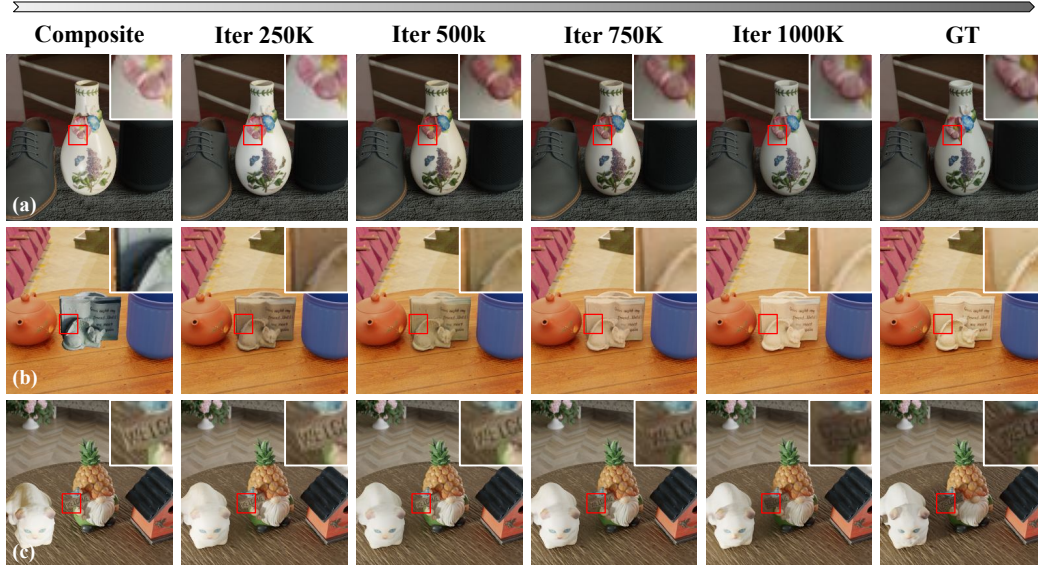


Figure 8: Qualitative comparison under sequential training iterations. We demonstrate the performance of the 2D object compositing model on test set scenarios on 250K, 500K, 750K, and 1000K training iterations. At 250K iterations, the model’s output diverges from the input primarily in the attenuation or removal of highlights and shadows on inserted objects. For example, the highlights on the vase surface in (a) and the shadow beneath the cat in (c) are significantly diminished. At 500K iterations, the source lighting of inserted objects is largely eliminated, while faint shadows emerge on the side opposing the scene lighting. At 750K iterations, inserted objects integrate coherently into the background scene, with generated shadows naturally cast onto the table or surrounding objects. At 1000K iterations, the model refines surface highlights and shadows with enhanced realism, emphasizing shadow details in localized regions. Notably, the carved patterns on the vase in (a) exhibit nuanced shadow variations, and the raised signboard in (c) casts a partial shadow occluded by the cat.

A.7 Learning Trend of 2D Object Compositing Model

In this section, we provide a systematic analysis of the training behavior of the 2D object compositing model, which is trained for a total of 1000K iterations. As shown in Fig. A.7, the model initially learns to remove highlights and shadows from the inserted objects. In the subsequent stages, the inserted objects become progressively harmonized with the scene, and new, scene-consistent shadows and highlights are generated. In the final phase, these lighting effects are further refined to enhance realism and local detail.

A.8 Object Compositing with VGGT Priors

VGGT has recently garnered significant attention due to its robust geometric capabilities in efficiently establishing relative relationships across unposed images. Leveraging this powerful prior, we explore the performance boundaries of our model under constrained input conditions, as shown in Fig. 9. When initializing Gaussians using VGGT-estimated poses and depth maps with fixed Gaussian positions and colors, the model produces blurred harmonized results. This stems from the insufficient accuracy of VGGT-estimated camera poses. To address this, we introduce positional optimization for Gaussian primitives, which enhances geometric coherence and yields sharper composited outputs. Furthermore, increasing the density of Gaussian primitives improves detail preservation. However, optimizing Gaussian colors degrades performance by inducing overfitting to training views, manifesting as noisy artifacts in the Gaussian representation and consequently deteriorating harmonization quality. Our experiments demonstrate that under VGGT-derived pose and depth constraints, denser Gaussian distributions and positional optimization positively impact final results, while color optimization adversely affects output clarity.

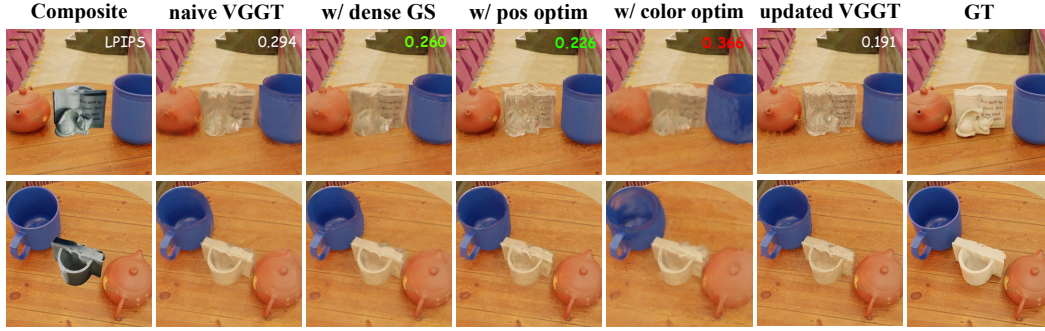


Figure 9: Object insertion results based on VGGT [8] priors. Given multiple unposed images as input, we estimate camera poses and depth maps via VGGT to compute point cloud positions for initializing Gaussians. We conduct several comparative experiments to analyze the impact of adjustments in the Gaussian training process on the final compositing results.

References

- [1] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *Advances in Neural Information Processing Systems*, 36:35799–35813, 2023.
- [2] Zhao Dong, Ka Chen, Zhaoyang Lv, Hong-Xing Yu, Yunzhi Zhang, Cheng Zhang, Yufeng Zhu, Stephen Tian, Zhengqin Li, Geordie Moffatt, et al. Digital twin catalog: A large-scale photorealistic 3d object digital twin dataset. *arXiv preprint arXiv:2504.08541*, 2025.
- [3] Julian Jorge Andrade Guerreiro, Mitsuru Nakazawa, and Björn Stenger. Pct-net: Full resolution image harmonization using pixel-wise color transformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5917–5926, 2023.
- [4] David Hilbert. *Dritter Band: Analysis· Grundlagen der Mathematik· Physik Verschiedenes: Nebst Einer Lebensgeschichte*. Springer-Verlag, 2013.
- [5] Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, et al. Anysplat: Feed-forward 3d gaussian splatting from unconstrained views. *arXiv preprint arXiv:2505.23716*, 2025.
- [6] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023.
- [7] Benjamin Ummerhofer, Sanskar Agrawal, Rene Sepulveda, Yixing Lao, Kai Zhang, Tianhang Cheng, Stephan Richter, Shenlong Wang, and German Ros. Objects with lighting: A real-world dataset for evaluating reconstruction and rendering for object relighting. In *2024 International Conference on 3D Vision (3DV)*, pages 137–147. IEEE, 2024.
- [8] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.
- [9] Bo Zhang, Yuxuan Duan, Jun Lan, Yan Hong, Huijia Zhu, Weiqiang Wang, and Li Niu. Control-com: Controllable image composition using diffusion model. *arXiv preprint arXiv:2308.10040*, 2023.