IntrinsicNeRF: Learning Intrinsic Neural Radiance Fields for Editable Novel View Synthesis

Weicai Ye[1*] Shuo Chen[1*] Chong Bao[1] Hujun Bao[1] Marc Pollefeys[2,3] Zhaopeng Cui[1]
Guofeng Zhang[1†]
[1]State Key Lab of CAD&CG, Zhejiang University [2] ETH Zurich [3] Microsoft

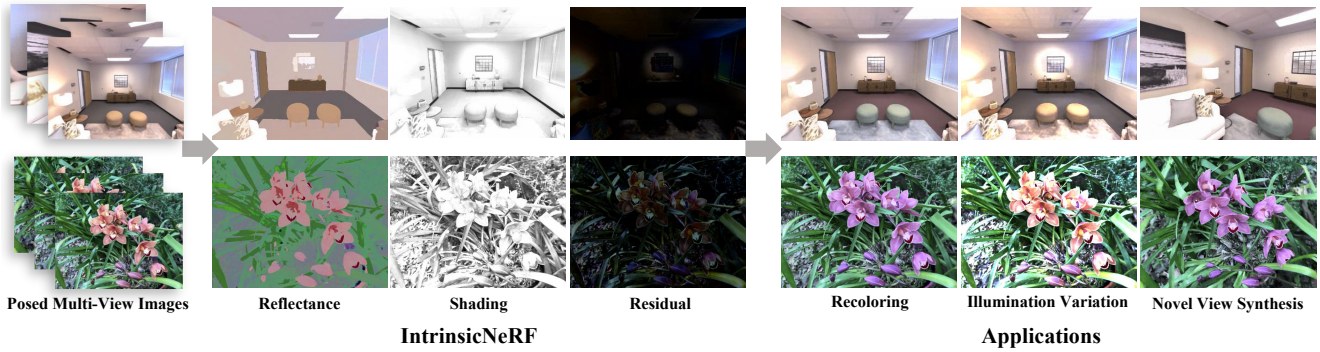

Figure 1: **Intrinsic Neural Radiance Fields (IntrinsicNeRF).** Given multi-view posed images of static scenes, IntrinsicNeRF can factorize the scene into the temporally consistent components: reflectance, shading, and residual layers. The decomposition can support online applications such as scene recoloring, illumination variation, and novel view synthesis.

Abstract

Existing inverse rendering combined with neural rendering methods [72, 75] can only perform editable novel view synthesis on object-specific scenes, while we present intrinsic neural radiance fields, dubbed IntrinsicNeRF, which introduce intrinsic decomposition into the NeRF-based [43] neural rendering method and can extend its application to room-scale scenes. Since intrinsic decomposition is a fundamentally under-constrained inverse problem, we propose a novel distance-aware point sampling and adaptive reflectance iterative clustering optimization method, which enables IntrinsicNeRF with traditional intrinsic decomposition constraints to be trained in an unsupervised manner, resulting in temporally consistent intrinsic decomposition results. To cope with the problem that different adjacent instances of similar reflectance in a scene are incorrectly clustered together, we further propose a hierarchical clustering method with coarse-to-fine optimization to obtain a fast hierarchical indexing representation. It supports compelling real-time augmented applications such as recoloring and illumination variation. Extensive experiments and editing samples on both object-specific/room-scale scenes and synthetic/real-word data demonstrate that we can obtain consistent intrinsic decomposition results and high-fidelity novel view synthesis even for challenging sequences. Project page: https://zju3dv.github.io/intrinsic_nerf/.

1. Introduction

Recently neural rendering techniques have gained increasing attention and demonstrated tremendous performance in novel view synthesis, ranging from small objects [34, 38, 43, 61] to large outdoor scenes [38, 58], but they struggle to perform further intuitive editing like realistic scene recoloring, relighting, etc, for the scenes are usually represented as neural fields implicitly and required to be decomposed into the editable properties explicitly.

Several works have proposed to fulfill this goal by introducing inverse rendering into neural rendering [72, 75], where the scene is decomposed into geometry, reflectance, and illumination. However, since inverse rendering is fundamentally ambiguous and highly ill-posed, these works introduce many prior assumptions preventing the modeling of mutual occlusion, inter-reflection, and indirect light propagation of different objects in the scene. An accurate 3D surface recovery is also required as a prerequisite. All these factors limit their application to object-specific scenarios.

To empower such editable capabilities to the scene-level neural rendering, we present intrinsic neural radiance fields, which introduce intrinsic decomposition into neural render-

∗ indicates equal contribution. † indicates the corresponding author.

ing, based on the fact that intrinsic decomposition can be considered as a simplification of inverse rendering designed to provide interpretable intermediate representations (i.e., reflectance and shading) that are relatively easy to solve for both in small objects and large scenes. A potential naive solution may use the trained NeRF model to generate multi-view images and then performs multi-view intrinsic decomposition, where these two tasks are separated. In contrast, extending from NeRF [43], IntrinsicNeRF (see Sec. 3.1 and Fig. 2) takes the sampled spatial coordinate point $\mathbf{x} = (x, y, z)$ and the direction $\mathbf{d} = (\theta, \phi)$ as input and regresses them into density σ, the view-independent reflectance r and shading s (Lambertian reflectance assumption) and additional view-dependent residual term re [39, 59] (Eq. 2), which naturally guarantees the multi-view consistency of decomposition after training, thanks to neural rendering.

However, it is nontrivial to design such a framework due to huge gaps in optimization between traditional intrinsic decomposition and NeRF-based methods. Traditional intrinsic decomposition methods optimize the energy equation by establishing constraints related to the image pixels, while NeRF-based methods optimize the view-dependent densities and colors of several sampled 3D points through volume rendering, which makes it hard to exploit the commonly used prior knowledge in intrinsic decomposition (see Sec. 3.2) such as chromaticity prior, reflectance sparsity, etc. To address this problem, we propose a distance-aware sampling method (see Fig. 3) that allows the sampled points not only to be random but also to establish local and global relationships between points. In this way, IntrinsicNeRF satisfies both the novel view synthesis and the better recovery of the intrinsic properties of the scene.

Moreover, to deal with the inconsistencies of similar reflectance regions [41], we present an adaptive reflectance iterative clustering method (see Sec. 3.3) with mean shift [12] to adaptively cluster color points with similar reflectance based on the scene itself, rather than K-Means used in [41], which limits the number of specific classes. A continuously updated clustering operation with the voxel grid filter is constructed to map similar reflectance colors to the same target reflectance color and then obtain the clustered category for each color point (see Fig. 4).

To settle the problem of different adjacent instances of similar reflectance in a scene being clustered together, we propose a semantic-aware reflectance sparsity constraint during training. Inspired by Semantic-NeRF [77], we add an additional semantic branch to IntrinsicNeRF, along with reflectance clustering, which yields a hierarchical reflectance iterative clustering and indexing method (see Fig. 5), optimizing the network from coarse to fine. Extensive experiments on Blender Object and Replica Scene demonstrate our method can obtain consistent intrinsic decomposition results and high-fidelity novel view synthesis even for challenging sequences. Based on rendering results of IntrinsicNeRF, we develop video editing software to facilitate users to perform scene recoloring, illumination variation, and editable view synthesis on real-world and synthetic data in real-time on the CPU (see Fig. 1).

2. Related Work

Intrinsic Image Decomposition. Intrinsic decomposition [1] is a typical image layer separation problem aimed at decomposing images into reflectance, shading, etc., and has been studied for decades. To deal with this ill-posed problem, additional priors [19, 28, 54] with optimization framework have been used. Recently, deep learning methods [2, 15, 33, 37, 71, 78] have emerged to perform intrinsic decomposition, and with large datasets [31, 32, 51], they have shown further improvement. Unsupervised intrinsic image decomposition works [20, 35] have also achieved impressive results. IntrinsicNeRF considers not only the intrinsic decomposition prior but also the consistency of different perspectives in neural rendering, performing unsupervised optimization of the network.

Intrinsic Video Decomposition. Intrinsic video decomposition extends intrinsic decomposition from the image domain to the video domain and can be roughly divided into two types. One is to perform the intrinsic image decomposition first and use the motion information to establish the correlation between frames for post-processing [7, 27, 65]. The other is to directly unify the image's local and global relations using some prior, by optimizing the energy equation [6, 41]. There are also works [14, 21, 26, 68] on intrinsic decomposition from multi-view images. These methods have some consistency in intrinsic video decomposition but are unable to perform novel view synthesis. While IntrinsicNeRF introduces traditional intrinsic decomposition prior to the neural radiance fields to achieve end-to-end optimization, which not only performs better intrinsic video decomposition than previous methods but also allows for realistic editable novel view synthesis.

Inverse Rendering. Inverse rendering [17] is another way to restore the basic properties of scene elements, which can be broadly classified into three categories: traditional approaches [4, 47, 22], differentiable renders [29, 46, 76, 36] and neural rendering methods. Plenty of works combining neural rendering with inverse rendering [5, 8, 52, 63, 74, 72, 75] have shown realistic view synthesis and consistent estimation of the underlying properties of the objects. In contrast, we introduce intrinsic decomposition into neural rendering, which extends editable novel view synthesis applications not only for objects but room-scale scenes.

3. Method

Given multi-view posed images under unknown illumination of a static scene, our goal is to achieve a reliable

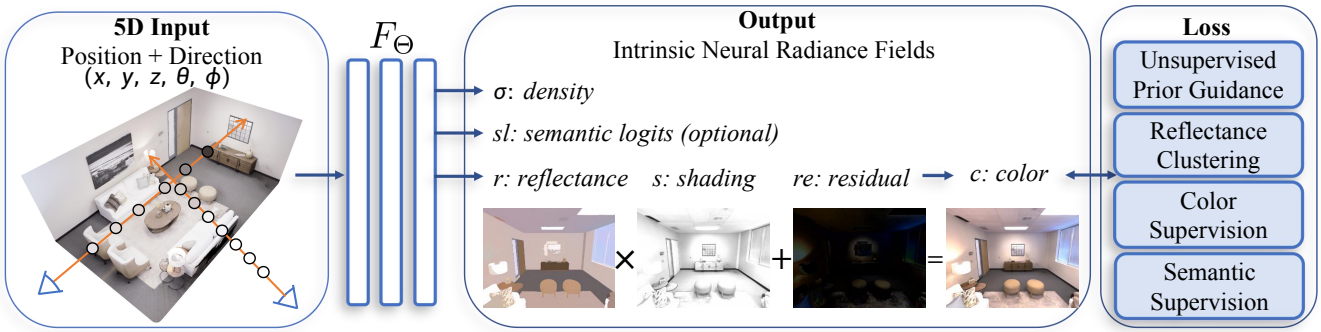

Figure 2: **IntrinsicNeRF Framework.** IntrinsicNeRF takes the sampled spatial coordinate point and direction as input, and outputs the density, reflectance, shading, and residual term. The semantic branch is optional. Unsupervised Prior and Reflectance Clustering are exploited as the loss function constraints to train the IntrinsicNeRF in an unsupervised manner.

understanding of the basic properties of the scene, such as reflectance, shading, etc, and to enable real-time editable novel view synthesis. Fig. 2 outlines the general framework of our approach.

3.1. Intrinsic Neural Radiance Fields

Preliminaries: Intrinsic Decomposition. Lambertian and grayscale shading assumptions [15] are commonly used and introduced to simplify this inverse problem, achieving good approximations of most scenarios. Based on Lambertian assumption, Intrinsic decomposition [15] presents an input image I as the pixel-wise product of the illumination-invariant reflectance $R(I)$, and the illumination-varying shading $S(I)$:

$$C(I) = R(I) \odot S(I) \text{ [15]} \tag{1}$$

where $\odot$ is channel-wise multiplication. However, the Lambertian assumption is difficult to be satisfied in realistic scenes, and the intrinsic residual model [59, 39] introduces view-independent reflectance and shading with an additional view-dependent residual term $Re(I)$ to model scenes that do not satisfy the Lambertian assumption, such as specular reflections, metallic materials:

$$C(I) = R(I) \odot S(I) + Re(I) \text{ [59]} \tag{2}$$

Our representation. IntrinsicNeRF takes the sampled coordinate point $\mathbf{x} = (x, y, z)$ and direction $\mathbf{d} = (\theta, \phi)$ as input, and outputs the view-independent reflectance r and shading s, the view-dependent intrinsic residual term re and the volume density σ through an MLP network F_Θ:

$$(r, s, re, \sigma) = F_\Theta(\mathbf{x}, \mathbf{d}) \tag{3}$$

The predicted color c of each spatial point can be obtained by Eq. 2 and the target color $C(\mathbf{r})$ of camera ray $\mathbf{r}$ is:

$$\hat{C}(\mathbf{r}) = \sum_{k=1}^{K} \hat{T}_k \, \alpha_k \, c_k \text{ and } \hat{T}_k = \exp\left(- \sum_{k'=1}^{k-1} \sigma_k \delta_k \right) \tag{4}$$

where $\alpha_k = 1 - \exp(-\sigma_k \delta_k)$, and δ_k is the distance between two adjacent sample points. We follow NeRF's coarse-to-fine training policy and train IntrinsicNeRF from scratch with the photometric loss L_{pho} in NeRF [43].

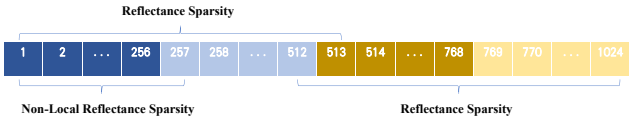

Figure 3: **Distance-Aware Point Sampling**. We first randomly sample 512 points, and then randomly sample the remaining 512 points in the eight neighborhoods of each sampled point to construct the unsupervised constraint term for the intrinsic decomposition.

Distance-Aware Point Sampling. NeRF [43] randomly samples batches of camera rays from the image pixel set (roughly 1024 points) in each optimization, where these points are random, and no relationship is established between them. It is not applicable in IntrinsicNeRF, for the introduction of ill-posed intrinsic decomposition into NeRF makes the whole optimization process stochastic, as shown in Fig. 7 (Baseline). To this end, we make a sophisticated design of the sampling policy (see Fig. 3) which helps to construct intrinsic prior constraints (see Sec. 3.2), and the network can be trained unsupervised.

3.2. Unsupervised Prior Guidance

Following intrinsic decomposition works [41], we adopt the grayscale shading assumption to simplify this inverse problem, so that the shading layer is single-channel and the reflectance chromaticity of the image I is approximated to $c(\mathbf{x}) = I(\mathbf{x})/|I(\mathbf{x})|$. We define the chromaticity similarity weight $\omega_{cs}(\mathbf{x}, \mathbf{y})$ [41] that is associated with many priors:

$$\omega_{cs(\mathbf{x},\mathbf{y})} = \exp(-\alpha_{cs} \|c(\mathbf{x}) - c(\mathbf{y})\|_2^2) \text{ [41]} \tag{5}$$

where $\mathbf{x}$ and $\mathbf{y}$ are the image pixel coordinates. Coefficient α_{cs} = 60 produces the best decomposition results.

Chromaticity Prior. Due to the residual term, the chromaticity of the unknown reflectance R and the input image

I are not the same. We want them to be as close as possible:

$$L_{chrom(\mathbf{x})} = \|c_r(\mathbf{x}) - c(\mathbf{x})\|_2^2 \quad (6)$$

where c and c_r are the chromaticity of the input sample points and the sampled points' reflectance, respectively.

Reflectance Sparsity. Two pixels that are similar in spatial location and chromaticity, have converging reflectance r, which leads to reflectance sparsity. We minimize the reflectance gradients magnitude independently:

$$L_{reflect(\mathbf{x})} = \sum_{\mathbf{y}\in\mathcal{N}(\mathbf{x})} \omega_{cs(\mathbf{x},\mathbf{y})} \|r(\mathbf{x}) - r(\mathbf{y})\|_2^2 \text{ [41]} \quad (7)$$

where $N(\mathbf{x})$ is the neighbourhood of pixel $\mathbf{x}$. Specifically, in IntrinsicNeRF, the sampled points in the first half will be adjacent to the second half, shown in Fig. 3.

Non-Local Reflectance Sparsity. In natural and man-made scenes, two distant spatial points may also have the same reflectance, such as a wall and floor that occupy a larger image area, which requires non-local reflectance sparsity. In the sampling of IntrinsicNeRF, the first half of the points are randomly sampled, so the distance between any two points can be very far. We simply bisect the first half of the points and construct a non-local reflectance sparsity constraint on the points in the first 1/4 segment and the corresponding points in the next 1/4 segment:

$$L_{non-local(\mathbf{x})} = \sum_{\mathbf{y}\in\mathcal{F}(\mathbf{x})} \omega_{cs(\mathbf{x},\mathbf{y})} \|r(\mathbf{x}) - r(\mathbf{y})\|_2^2 \text{ [41]} \quad (8)$$

where $\mathcal{F}(\mathbf{x})$ is the farhood of pixel $\mathbf{x}$. Note that the weight of this constraint is smaller than the reflectance sparsity's.

Shading Smoothness. Natural objects usually have smooth surfaces and the shading variance is expected to be smooth [41]. Moreover, neighboring pixels with different chromaticities, represent a reflectance edge, so we strongly enforce the shading smoothness:

$$L_{shade(\mathbf{x})} = \sum_{\mathbf{y}\in\mathcal{N}(\mathbf{x})} \|c(\mathbf{x}) - c(\mathbf{y})\|_2^2 \|s(\mathbf{x}) - s(\mathbf{y})\|_2^2 \quad (9)$$

Intrinsic Residual Constraints. Since diffuse light generally dominates the scene, we want the image content to be recovered by reflectance and shading as much as possible.This prevents extreme cases when R and S both converge to zero, and $Re = I$, which would destroy the efficacy of the previous constraints and fall into catastrophic results. We set this constraint as follows:

$$L_{residual(\mathbf{x})} = \|re(\mathbf{x})\|_2^2 \quad (10)$$

The weight is set higher early, so $R(I) \odot S(I)$ is close to the target image I and then dropped lately. As the output of R and S is stable, R_e can represent the view-dependent components, such as specular reflection.

Intensity Prior. The previous constraints on reflectance and shading only consider the relative relationship between two pixels. The absolute magnitude of R and S is required to prevent them from falling into certain extremes during optimization. The intensity of the unknown reflectance image R and the input image I should be close:

$$L_{intensity(\mathbf{x})} = \|i_r(\mathbf{x}) - i(\mathbf{x})\|_2^2 \quad (11)$$

where i and i_r are the average intensities of the batch sample points $\mathbf{x}$ of the input and reflectance r. The weight of this constraint is set higher early and then reduced.

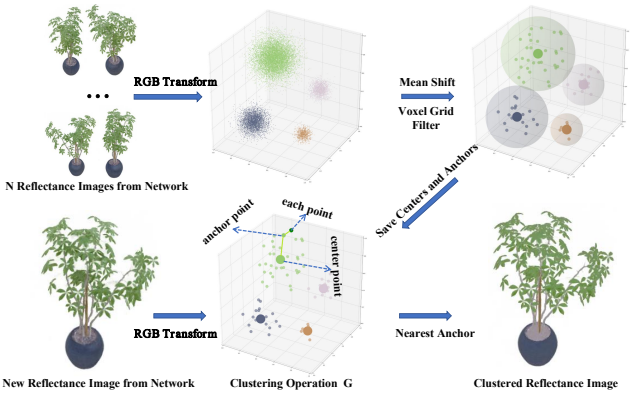

Figure 4: **Adaptive Reflectance Iterative Clustering Method**. The color of the reflectance pixels is first converted and then clustered with mean shift algorithms. The voxel grid filter is performed to accelerate the processing of the cluster operation G.

3.3. Adaptive Reflectance Iterative Clustering

Although reflectance sparsity makes sense to some extent, there still remain inconsistencies of similar reflectance regions (see Fig. 7 +w/prior), therefore we propose an adaptive reflectance iterative clustering method by constructing a continuously updated clustering operation G, which maps similar reflectance colors $r(\mathbf{x})$ to the same target reflectance color $r_{cluster}(\mathbf{x})$ by adding a clustering constraint during the optimization of the network:

$$L_{cluster(\mathbf{x})} = \|r_{cluster}(\mathbf{x}) - r(\mathbf{x})\|_2^2 \quad (12)$$

Next, we elucidate the detail of the clustering method.

RGB Transform. During the training of the network, we infer the reflectance r, shading s, and residual term re of multi-view posed images after every 10K iterations. Refer to IIW [3], we take out all pixels of all r components and convert their colors to better cluster reflectances (pixel intensity, red chromaticity, green chromaticity [41]):

$$\boldsymbol{f}([\mathbf{r},\mathbf{g},\mathbf{b}]) = [\beta\frac{\mathbf{r}+\mathbf{g}+\mathbf{b}}{\mathbf{3}}, \frac{\mathbf{r}}{\mathbf{r}+\mathbf{g}+\mathbf{b}}, \frac{\mathbf{g}}{\mathbf{r}+\mathbf{g}+\mathbf{b}}] \quad (13)$$

where β is set as 0.5 in our experiment. The RGB transformation helps reduce the effect of intensity differences on the clustering, making the clustering more focused on the similarity of chromaticity between two pixels. The transformed RGB space is considered as $\boldsymbol{f}$ space.

Mean Shift. Unlike existing methods [41] using K-Means clustering to specify K clustering categories, we instead cluster all the pixel points P every 10K iterations with a Mean Shift clustering algorithm to adaptively determine the number of reflectance classes in the scene, for we do not know the reflectance class number.

Clustering Operation *G*. After Mean Shift clustering, we get a set of clustered centers, and a classification label for each pixel point in P. During each training iteration, it is unrealistic to cluster the reflectance of each rendered pixel because the clustering is time-consuming. So we define a fast approximation clustering operation G: for an RGB value of reflectance, it considers the category of the nearest point in P as its cluster category and set the value of the category center as the target clustered reflectance $r_{cluster}(\mathbf{x})$. When calculating the clustering loss, we only use Clustering Operation G, shown in Fig. 4.

Voxel Grid Filter. Since there are plenty of points P in the $\boldsymbol{f}$ space and most of them are clustered in very small regions due to reflectance sparsity, rather than finding the nearest neighbors in all points, we perform voxel grid filter (voxel size is 0.01) on the points P in the $\boldsymbol{f}$ space, and the filtered points are regarded as anchor points. The clustering operation G therefore only needs to search the closest anchor point, and the anchor points are only been updated every 10K iterations by Mean-Shift.

Optimization. During the network optimization, the weight of the clustering loss $L_{cluster(\mathbf{x})}$ and the bandwidth parameter in the mean shift algorithm are gradually increased with the number of iterations (the larger the bandwidth is, the smaller the number of mean-shift clustering categories is). That is because, in the early stage of network optimization, the inferred reflectance r is not reliable and needs lower weight. While in the later stage, a higher weight can lead the output of the network to converge toward the effect of clustering, making the reflectance r before and after clustering indistinguishable.

3.4. Hierarchical Clustering and Indexing

The adaptive reflectance iterative clustering method can handle object-level scenes well, shown in Fig. 7 (Ours). However, when the reflectance in the scene is complex and similar, plenty of different instances with similar reflectance in room-scale scenes may be incorrectly clustered, shown in Fig. 8 (w/prior+cluster). We propose a semantic-aware reflectance sparsity constraint, where only pixels with the same semantic label will be computed, thus significantly improving the quality of reflectance. Inspired by [77], we extend IntrinsicNeRF to jointly encode appearance, geometry, and semantics by adding a segmentation renderer to the original IntrinsicNeRF. Specially, we use a view-invariant MLP function $sl = F_\Theta(\mathbf{x})$ to map a spatial coordinate $\mathbf{x}$ to semantic label and use the semantic loss L_{sem} in [77].

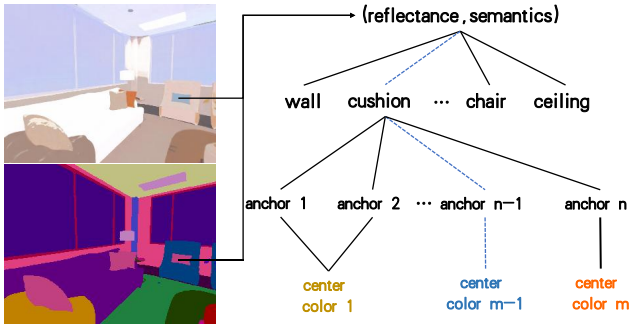

Figure 5: **Hierarchical Clustering and Indexing Method.**

Depending on the semantic labels of each pixel, the pixel set P can be divided into multiple subsets $\{\mathbf{P_1}, \mathbf{P_2}, ..., \mathbf{P_N}\}$, where N is the number of semantic categories. Then we can construct N clustering operations $\{\mathbf{G_1}, \mathbf{G_2}, ..., \mathbf{G_N}\}$ as Sec. 3.3. The hierarchical clustering operation takes the reflectance RGB value of each pixel and the corresponding semantic label as input, and outputs the result of the clustering operation for the pixel under the semantic label. Such a hierarchical clustering method allows the clustered information of each pixel to be stored in a tree structure, shown in Fig. 5, which can be indexed quickly.

3.5. Implementation Details

We implement IntrinsicNeRF on the top of SemanticNeRF [77] with additional three FC layers for intrinsic components which have 128 neurons. The network is optimized with photometric loss, semantic loss, unsupervised prior constraints, and clustering loss jointly. The final loss is:

$$\begin{aligned} L_{final} = &\lambda_{pho}L_{pho} + \lambda_{sem}L_{sem} + \lambda_{chrom}L_{chrom} \\ &+ \lambda_{reflect}L_{reflect} + \lambda_{non-local}L_{non-local} \\ &+ \lambda_{shade}L_{shade} + \lambda_{cluster}L_{cluster} \\ &+ \lambda_{residual}L_{residual} + \lambda_{intensity}L_{intensity} \end{aligned} \quad (14)$$

Here, $\lambda_{pho} = 1$, $\lambda_{sem} = 0.04$, $\lambda_{chrom} = 1$, $\lambda_{reflect} = 0.01$, $\lambda_{non-local} = 0.005$ and $\lambda_{shade} = 1$. While $\lambda_{cluster} = 10^{-2(1-iter/200K)}$, it exponentially increases from 0.01 to 1 every 10K iterations. We set $\lambda_{residual} = 1$ in the early 100K iterations and dropped to 0.02 in the later iterations. The $\lambda_{intensity}$ is set to 0.1 in the first 50K iterations and then set to 0.01. The batch size of the rays is 1024. The Adam [23] optimizer is used with a learning rate of 5e-4 for 200K iterations. The training and test time are shown in the supplemental material (Tab. B1).

Method	Reflectance (Invrender dataset)					View Synthsis (Invrender dataset)			Reflectance (our dataset)					View Synthsis (our dataset)		
	PSNR ↑	SSIM ↑	LPIPS ↓	MSE ↓	LMSE ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	MSE ↓	LMSE ↓	PSNR ↑	SSIM ↑	LPIPS ↓
IIW[3]	22.0284	0.9307	0.0847	0.0099	**0.0120**	-	-	-	20.5299	0.9079	0.1131	0.0102	0.0727	-	-	-
CGIntrinsic[31]	20.1583	0.9209	0.0996	0.0129	0.0141	-	-	-	18.3542	0.8999	0.1229	0.0156	0.0659	-	-	-
USI3D [35]	20.7571	0.9267	0.0887	0.0079	0.0149	-	-	-	19.1489	0.9115	0.1070	0.0135	0.0524	-	-	-
NeRFactor[74]	19.9167	0.9156	0.1354	0.0059	0.0210	23.0133	0.9277	0.0822	21.4440	0.9170	0.1055	**0.0063**	**0.0444**	20.6880	0.8733	0.1185
PhySG[72]	23.3748	0.9231	0.1092	0.0034	0.0396	25.4225	0.9388	0.0804	-	-	-	-	-	-	-	-
Invrender [75]	**26.3078**	**0.9380**	**0.0572**	0.0022	0.0226	29.3870	0.9522	0.0505	-	-	-	-	-	-	-	-
Baseline	16.3209	0.8637	0.1301	0.0254	0.1955	**34.0036**	**0.9670**	**0.0252**	14.8572	0.8397	0.1738	0.0451	0.1849	**28.2604**	**0.9383**	**0.0339**
Baseline + w/ prior.	21.7370	0.9278	0.1086	0.0055	0.0186	33.4909	0.9638	0.0304	20.9646	0.9140	0.1216	0.0095	0.0538	28.0633	0.9370	0.0369
Ours	24.2642	0.9371	0.0880	**0.0021**	0.0173	33.4967	0.9630	0.0306	**22.5677**	**0.9267**	**0.0975**	0.0066	0.0474	27.9494	0.9357	0.0372

Table 1: **Quantitative Results of Blender Object.** For reflectance estimation, IntrinsicNeRF achieved the best results on our dataset and ranked 2nd on the invrender dataset. For novel view synthesis, IntrinsicNeRF achieved the best performance on both datasets, while Invrender [75] and PhySG [72] require good geometric prerequisites, which makes them fail on our dataset. Moreover, intrinsic decomposition methods can not perform novel view synthesis. Bold indicates the best and underline indicates the second best. - means failure.

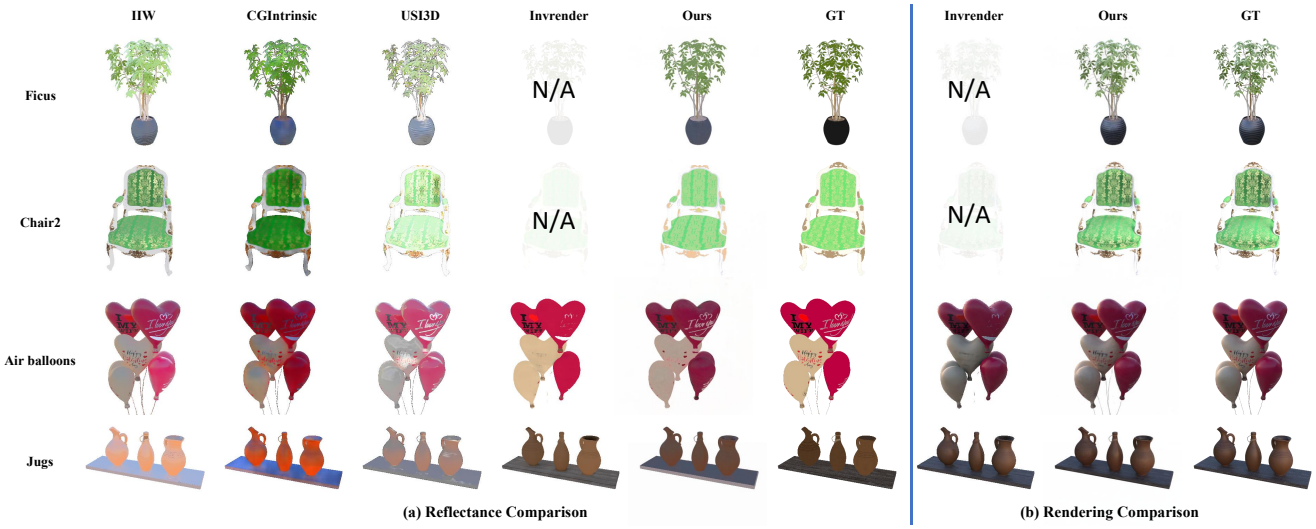

Figure 6: **Qualitative Comparison Sample Results of Reflectance and Rendering on Bleneder Object.** The top 2 rows represent our samples and the bottom 2 rows are the Invrender samples. Our method can perform reflectance estimation and novel view synthesis on both datasets well, while Invrender [75] fails to do that on our dataset. N/A means failure.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
NeRF [43]	31.0838	0.9525	0.0302
Ours	30.7230	0.9494	0.0339

Table 2: **Quantitative Results for Novel View Synthesis on Blender Object.** We achieve comparable results compared with NeRF [43], while giving the power of modeling the basic properties of scenes.

4. Experiments

We first make qualitative and quantitative comparisons of IntrinsicNeRF with traditional optimization-based [3] and learning-based [31, 35] intrinsic decomposition methods, and neural rendering methods [74, 72, 75] combined with inverse rendering on synthetic object dataset in Sec. 4.2. Then we only compare qualitative results on synthetic scenes (e.g. Replica [55]) in Sec. 4.3, due to the lack of ground-truth labels. Finally, we perform ablations to analyze the design of our framework and demonstrate its applicability both on synthetic and real-world data.

4.1. Dataset

Synthetic Data. We collect 8 Blender object (4 from Invrender [75], and 4 from NeRF [43]) and 8 Replica Scene datasets. The Invrender dataset contains hotdog, jugs, chair, and air balloons, and each dataset is rendered by Blender Cycles with their masks, albedo, and roughness maps. The NeRF dataset contains 4 objects (lego, drums, ficus, and chair2) that maintain complex geometry and realistic non-Lambertian materials. Note that some environment lighting maps in NeRF's open-source blender model were missing, we search for some environment maps that look as realistic as possible, and re-render the new image to match NeRF's settings. We regard this dataset as our dataset. The image resolution is set as 400×400.

Generated by Semantic-NeRF [77], each Replica Scene [55] of rooms and offices consists of RGB images, depth maps, and semantic labels at resolution 320x240 from randomly generated 6-DOF trajectories. It contains challenging illumination effects, such as specular reflections.

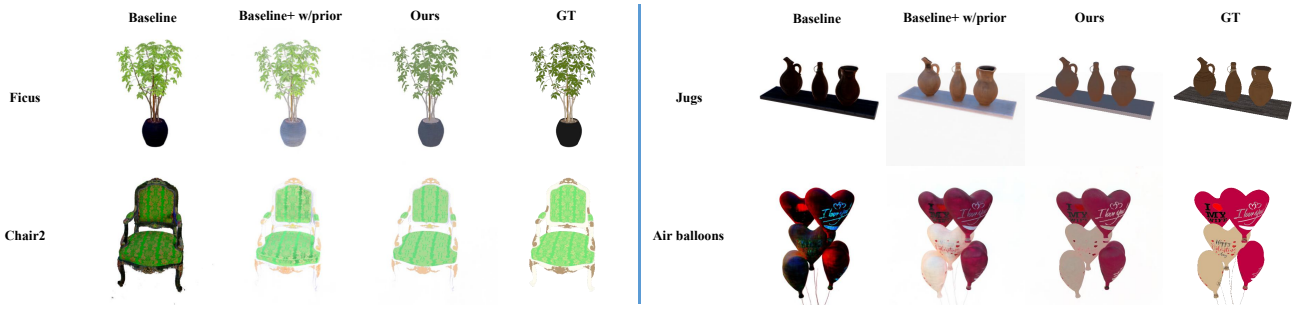

Figure 7: **Ablation study of Reflectance Estimation Sample on Blender Object.** Left: our dataset, right: Invrender dataset. The reflectance estimation of the baseline method is stochastic and unstable, while the intrinsic prior makes the optimization of the network traceable. Our final model achieves more plausible albedo results.

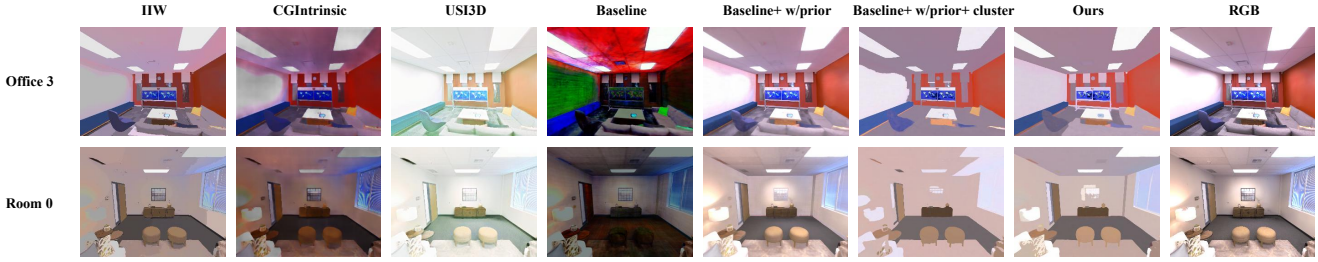

Figure 8: **Qualitative Reflectance Comparison Sample with Previous Methods on Replica Scene.** Experiments demonstrate the progressive facilitation effect of our different variants. Compared with previous methods, our final method achieves more plausible and consistent albedo estimation results, retaining the boundaries of objects.

Real-world Data. We selected 4 real data of natural scenes (orchids, flowers, horns, and ferns at 504x378 resolution) from LLFF [42] to demonstrate the generalization ability of our method in real-world lighting and reflection and its applicability, such as recoloring, illumination variation.

4.2. Comparison Results on Blender Object

We exploit Peak Signal-to-Noise Ratio (PSNR), Mean Squared Error (MSE), Structural Similarity Index Measure (SSIM), Local Mean Squared Error (LMSE), and Learned Perceptual Image Patch Similarity (LPIPS) [73] as reflectance evaluation metrics. We do not evaluate the shading quantitatively because different methods model lighting differently, and we cannot get the ground-truth shading of our model in Blender. In contrast, reflectance is a common output and has ground-truth values, so we focus on the evaluation of reflectance. The view synthesis evaluation metrics are PSNR, SSIM, and LPIPS.

We compare IntrinsicNeRF with the following methods: IIW [3] is a classic intrinsic decomposition method that does not require training. CGIntrinsic [31] is a learning method with good generalization trained on large-scale datasets, and USI3D [35] is another state-of-the-art unsupervised learning method, and we use their pre-trained models. We do not choose IRISformer [78] and intrinsic video decomposition methods [41, 40], because their codes are not available even if we send the email for the code request or the dataset is not suitable. NeRFactor [74], PhySG [72], and InvRender [75] are the neural rendering method, we have retrained them in the same setting for a fair comparison. Tab. 1 shows our method achieves the best results on our dataset and ranked 2nd on the Invrender dataset for reflectance estimation. Compared with single-view intrinsic decomposition methods, our method yields more consistent and plausible decomposition results, even in challenging object scenes, such as Chair2, and Ficus. As for view synthesis, IntrinsicNeRF achieves the best performance on both datasets, while Invrender [75] and PhySG [72] require good geometric prerequisites using IDR method [64], which makes them fail on our dataset, as shown in Fig. 6. Moreover, traditional intrinsic decomposition methods can not perform novel view synthesis. Tab. 2 shows our method achieves comparable novel view synthesis results, compared with NeRF [43], while giving the power of modeling the intrinsic components of scenes.

4.3. Comparison Results on Replica Scene

We only compare qualitative results with intrinsic decomposition methods [35, 3, 31] on Replica Scene in reflectance estimation, because we cannot obtain the ground-truth of reflectance. Fig. 8 shows that we can obtain more plausible results than other intrinsic decomposition methods, and maintain consistent reflectance estimation for multi-view images, as shown in the supplementary video. Moreover, our method obtains comparable results with Semantic-NeRF [77] in novel view synthesis and seman-

Method	PSNR ↑	SSIM ↑	LPIPS ↓	mIoU ↑
[77]	30.9770	0.8955	0.1066	0.9725
Ours	30.7044	0.8908	0.1140	0.9702

Table 3: **Quantitative Results for Novel View Synthesis and Semantic Segmentation on Replica Scene.** We achieve comparable results compared with Semantic-NeRF [77] while giving the power of modeling the basic properties of scenes.

tic segmentation (the metric is mIOU), shown in Tab. 3, and we give Semantic-NeRF the ability to model the intrinsic properties of the scene (Fig. 1). While PhySG [72] and Invrender [75] fail to do that in room-scale scenes.

4.4. Ablation Studies

We perform ablation studies to analyze three components of our method that primarily affect the intrinsic decomposition quality. The baseline method is the NeRF [43] variant with intrinsic neural radiance fields, using the proposed distance-aware point sampling policy. Tab. 1 shows that the introduction of the intrinsic prior and iterative clustering leads to more accurate reflectance estimation, with a slight decrease in the accuracy of the novel view synthesis. Fig. 7 show that the reflectance estimated by the baseline method is more stochastic and unstable. While adding the intrinsic prior, the network output is plausible. The adaptive reflectance iterative clustering method can make the reflectance regions of the same material cluster together. The complete quantitative comparison results of IntrinsicNeRF variants in different object scenarios are shown in the supplemental material (Tab. C2 and Tab. C3). However, reflectance clustering may lose some distinguishable boundaries in room-scale scenes such as Replica, for complex and similar reflectance may be clustered incorrectly. Whereas the hierarchical clustering method can retain the boundaries and still yields more plausible results, as shown in Fig. 8. More qualitative comparison results of IntrinsicNeRF variants in different scenarios (both object and scene) are shown in the supplemental material (Fig. C5 and Fig. C7).

4.5. Applications

We demonstrate the applicability of IntrinsicNeRF with its decomposed components and the rendering results, such as real-time recoloring, illumination variation, and editable novel view synthesis on both synthetic and real-world data.
Scene Recoloring. In IntrinsicNeRF, the predicted reflectance is saved as [Semantic category, reflectance category] in the hierarchical iterative clustering and indexing method. We can simply modifying the color of a certain reflectance category, the reflectance values of all pixels belonging to the selected category can be modified at the same time, and then the edited images can be reconstructed using the modified reflectance with the original shading and residual through Eq. 2. Fig. 9 shows some recoloring examples

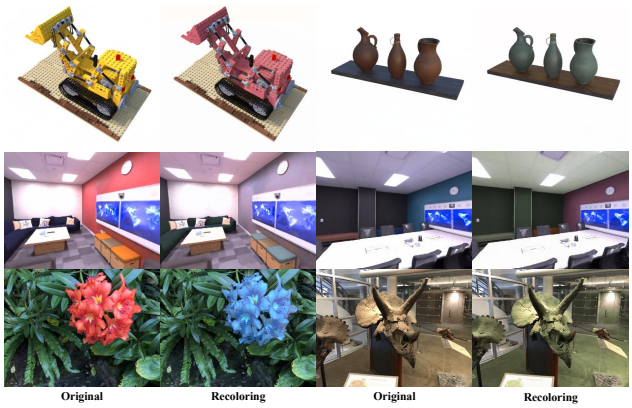

Figure 9: **Recoloring Examples on Synthetic and Real-World Data.**

Figure 10: **Illumination Variation on Replica Scene.**

on synthetic/real-world data. See more results in the supplemental material (Fig. C8).
Illumination Variation. The decomposed residual term beyond the Lambertian assumption, can represent the properties such as specular illumination, we can adjust the overall brightness directly by a multiplicative factor. Fig. 10 shows the effect of different light intensities after enhancing or diminishing the light. For more edited samples and view synthesis results, see Fig. C9 and C10 in the supp. material.

5. Conclusion

We introduce intrinsic decomposition into neural rendering and propose intrinsic neural radiance fields that can decompose the scene into reflectance, shading, and residual layers. Several techniques are proposed to make the decomposition learning feasible and support real-time augmented applications such as recoloring, illumination variation, and editable novel view synthesis. We believe our method is the step toward the intrinsic decomposition (beyond Lambertian assumption) of more general scenes with neural rendering and will inspire follow-up work. **Limitations.** The main limitation is that when the scenario does not conform to unsupervised intrinsic prior, it will struggle to obtain the correct decomposition results. Clustering errors may occur when the reflectance in a scene is complex and similar, especially in the real-world lacking semantic constraints. This can be solved by unsupervised semantic segmentation [18]. Estimating the reflectance requires a trade-off between preserving the texture and modeling the shadows correctly.

References

[1] Harry Barrow, J Tenenbaum, A Hanson, and E Riseman. Recovering Intrinsic Scene Characteristics. *Comput. vis. syst*, 2(3-26):2, 1978. 2

[2] Anil S Baslamisli, Thomas T Groenestege, Partha Das, Hoang-An Le, Sezer Karaoglu, and Theo Gevers. Joint Learning of Intrinsic Images and Semantic Segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 286–302, 2018. 2

[3] Sean Bell, Kavita Bala, and Noah Snavely. Intrinsic Images in the Wild. *ACM Transactions on Graphics (TOG)*, 33(4):1–12, 2014. 4, 6, 7, 2, 3

[4] Volker Blanz and Thomas Vetter. A Morphable Model for The Synthesis of 3D Faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, 1999. 2

[5] Boming Zhao and Bangbang Yang, Zhenyang Li, Zuoyue Li, Guofeng Zhang, Jiashu Zhao, Dawei Yin, Zhaopeng Cui, and Hujun Bao. Factorized and Controllable Neural Re-Rendering of Outdoor Scene for Photo Extrapolation. In *Proceedings of the 30th ACM International Conference on Multimedia*, 2022. 2

[6] Nicolas Bonneel, Kalyan Sunkavalli, James Tompkin, Deqing Sun, Sylvain Paris, and Hanspeter Pfister. Interactive Intrinsic Video Editing. *ACM Transactions on Graphics (TOG)*, 33(6):1–10, 2014. 2

[7] Nicolas Bonneel, James Tompkin, Kalyan Sunkavalli, Deqing Sun, Sylvain Paris, and Hanspeter Pfister. Blind Video Temporal Consistency. *ACM Transactions on Graphics (TOG)*, 34(6):1–9, 2015. 2

[8] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T Barron, Ce Liu, and Hendrik Lensch. NeRD: Neural Reflectance Decomposition from Image Collections. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12684–12694, 2021. 2

[9] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. TensoRF: Tensorial Radiance Fields. *arXiv preprint arXiv:2203.09517*, 2022. 5

[10] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. MVSNeRF: Fast Generalizable Radiance Field Reconstruction from Multi-View Stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14124–14133, 2021. 5

[11] Xingyu Chen, Qi Zhang, Xiaoyu Li, Yue Chen, Ying Feng, Xuan Wang, and Jue Wang. Hallucinated Neural Radiance Fields In The Wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12943–12952, 2022. 5

[12] Yizong Cheng. Mean Shift, Mode Seeking, and Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995. 2

[13] Chong Bao and Bangbang Yang, Zeng Junyi, Bao Hujun, Zhang Yinda, Cui Zhaopeng, and Zhang Guofeng. NeuMesh: Learning Disentangled Neural Mesh-based Implicit Field for Geometry and Texture Editing. In *European Conference on Computer Vision (ECCV)*, 2022. 5

[14] Sylvain Duchêne, Clement Riant, Gaurav Chaurasia, Jorge Lopez-Moreno, Pierre-Yves Laffont, Stefan Popov, Adrien Bousseau, and George Drettakis. Multi-view Intrinsic Images of Outdoors Scenes with An Application to Relighting. *ACM Transactions on Graphics*, page 16, 2015. 2

[15] Qingnan Fan, Jiaolong Yang, Gang Hua, Baoquan Chen, and David Wipf. Revisiting Deep Intrinsic Image Decompositions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8944–8952, 2018. 2, 3

[16] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. FastNeRF: High-fidelity Neural Rendering at 200FPS. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14346–14355, 2021. 5

[17] Elena Garces, Carlos Rodriguez-Pardo, Dan Casas, and Jorge Lopez-Moreno. A Survey on Intrinsic Images: Delving Deep into Lambert and Beyond. *International Journal of Computer Vision*, 130(3):836–868, 2022. 2

[18] Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T. Freeman. Unsupervised semantic segmentation by distilling feature correspondences. In *International Conference on Learning Representations*, 2022. 8

[19] Berthold KP Horn. Determining Lightness from An Image. *Computer graphics and image processing*, 3(4):277–299, 1974. 2

[20] Michael Janner, Jiajun Wu, Tejas D Kulkarni, Ilker Yildirim, and Josh Tenenbaum. Self-supervised Intrinsic Image Decomposition. *Advances in Neural Information Processing Systems*, 30, 2017. 2

[21] Alen Joy and Charalambos Poullis. Multi-view Gradient Consistency for SVBRDF Estimation of Complex Scenes under Natural Illumination. *arXiv preprint arXiv:2202.13017*, 2022. 2

[22] Yoshihiro Kanamori and Yuki Endo. Relighting Humans: Occlusion-aware Inverse Rendering for Full-body Human Images. *arXiv preprint arXiv:1908.02714*, 2019. 2

[23] Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[24] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. Panoptic Neural Fields: A Semantic Object-Aware Neural Scene Representation. In *CVPR*, 2022. 5

[25] Youngjoong Kwon, Dahun Kim, Duygu Ceylan, and Henry Fuchs. Neural Human Performer: Learning Generalizable Radiance Fields for Human Performance Rendering. *Advances in Neural Information Processing Systems*, 34, 2021. 5

[26] Pierre-Yves Laffont, Adrien Bousseau, Sylvain Paris, Frederic Durand, and George Drettakis. Coherent Intrinsic Images from Photo Collections. *ACM Trans. Graph.*, 2012. 2

[27] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning Blind Video Temporal Consistency. In *Proceedings of the European conference on computer vision (ECCV)*, pages 170–185, 2018. 2

[28] Edwin H Land and John J McCann. Lightness and Retinex Theory. *Josa*, 61(1):1–11, 1971. 2

[29] Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. Differentiable Monte Carlo Ray Tracing through Edge Sampling. *ACM Transactions on Graphics (TOG)*, 37(6):1–11, 2018. 2

[30] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural Scene Flow Fields for Space-time View Synthesis of Dynamic Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6498–6508, 2021. 5

[31] Zhengqi Li and Noah Snavely. CGIntrinsics: Better Intrinsic Image Decomposition through Physically-based Rendering. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 371–387, 2018. 2, 6, 7, 3

[32] Zhengqi Li and Noah Snavely. Learning Intrinsic Image Decomposition from Watching The World. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9039–9048, 2018. 2

[33] Andrew Liu, Shiry Ginosar, Tinghui Zhou, Alexei A Efros, and Noah Snavely. Learning to Factorize and Relight A City. In *European Conference on Computer Vision*, pages 544–561. Springer, 2020. 2

[34] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural Sparse Voxel Fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020. 1

[35] Yunfei Liu, Yu Li, Shaodi You, and Feng Lu. Unsupervised Learning for Intrinsic Image Decomposition from A Single Image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3248–3257, 2020. 2, 6, 7, 3

[36] Guillaume Loubet, Nicolas Holzschuch, and Wenzel Jakob. Reparameterizing Discontinuous Integrands for Differentiable Rendering. *ACM Transactions on Graphics (TOG)*, 38(6):1–14, 2019. 2

[37] Jundan Luo, Zhaoyang Huang, Yijin Li, Xiaowei Zhou, Guofeng Zhang, and Hujun Bao. NIID-Net: Adapting Surface Normal Knowledge for Intrinsic Image Decomposition in Indoor Scenes. *IEEE Transactions on Visualization and Computer Graphics*, 26(12):3434–3445, 2020. 2

[38] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf In the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7210–7219, 2021. 1

[39] Bruce A Maxwell, Richard M Friedhoff, and Casey A Smith. A Bi-illuminant Dichromatic Reflection Model for Understanding Images. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 2, 3

[40] Abhimitra Meka, Mohammad Shafiei, Michael Zollhöfer, Christian Richardt, and Christian Theobalt. Real-time Global Illumination Decomposition of Videos. *ACM Transactions on Graphics (TOG)*, 40(3):1–16, 2021. 7

[41] Abhimitra Meka, Michael Zollhöfer, Christian Richardt, and Christian Theobalt. Live Intrinsic Video. *ACM Transactions on Graphics (TOG)*, 35(4):1–14, 2016. 2, 3, 4, 5, 7

[42] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 2019. 7

[43] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020. 1, 2, 3, 6, 7, 8, 5

[44] Yuhang Ming, Weicai Ye, and Andrew Calway. iDF-SLAM: End-to-End RGB-D SLAM with Neural Implicit Mapping and Deep Feature Tracking. *arXiv preprint arXiv:2209.07919*, 2022. 5

[45] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant Neural Graphics Primitives with A Multiresolution Hash Encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022. 5

[46] Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. Mitsuba 2: A Retargetable Forward and Inverse Renderer. *ACM Transactions on Graphics (TOG)*, 38(6):1–17, 2019. 2

[47] Byong Mok Oh, Max Chen, Julie Dorsey, and Frédo Durand. Image-based Modeling and Photo Editing. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 433–442, 2001. 2

[48] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable Neural Radiance Fields. *ICCV*, 2021. 5

[49] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An Imperative Style, High-performance Deep Learning Library. *Advances in neural information processing systems*, 32, 2019. 1

[50] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural Radiance Fields for Dynamic Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 5

[51] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A Photorealistic Synthetic Dataset for Holistic Indoor Scene Understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10912–10922, 2021. 2

[52] Viktor Rudnev, Mohamed Elgharib, William Smith, Lingjie Liu, Vladislav Golyanik, and Christian Theobalt. NeRF for Outdoor Scene Relighting. In *European Conference on Computer Vision (ECCV)*, 2022. 2

[53] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. GRAF: Generative Radiance Fields for 3D-aware Image Synthesis. *Advances in Neural Information Processing Systems*, 33:20154–20166, 2020. 5

[54] Li Shen, Ping Tan, and Stephen Lin. Intrinsic Image Decomposition with Non-local Texture Cues. In *2008 IEEE Con-*

ference on Computer Vision and Pattern Recognition, pages 1–7. IEEE, 2008. 2

[55] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 6

[56] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. iMAP: Implicit Mapping and Positioning in Real-time. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6229–6238, 2021. 5

[57] Jiaming Sun, Xi Chen, Qianqian Wang, Zhengqi Li, Hadar Averbuch-Elor, Xiaowei Zhou, and Noah Snavely. Neural 3D Reconstruction in the Wild. In *SIGGRAPH Conference Proceedings*, 2022. 5

[58] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-NeRF: Scalable Large Scene Neural View Synthesis. *arXiv preprint arXiv:2202.05263*, 2022. 1

[59] Shoji Tominaga. Dichromatic Reflection Models for A Variety of Materials. *Color Research & Application*, 19(4):277–285, 1994. 2, 3

[60] Alex Trevithick and Bo Yang. GRF: Learning A General Radiance Field for 3d Representation and Rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15182–15192, 2021. 5

[61] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction. *NeurIPS*, 2021. 1

[62] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. IBRNet: Learning Multi-view Image-based Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2021. 5

[63] Bangbang Yang, Yinda Zhang, Yijin Li, Zhaopeng Cui, Sean Fanello, Hujun Bao, and Guofeng Zhang. Neural Rendering in a Room: Amodal 3D Understanding and Free-Viewpoint Rendering for the Closed Scene Composed of Pre-Captured Objects. *ACM Trans. Graph.*, 41(4):101:1–101:10, July 2022. 2

[64] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview Neural Surface Reconstruction by Disentangling Geometry and Appearance. *Advances in Neural Information Processing Systems*, 33, 2020. 7

[65] Genzhi Ye, Elena Garces, Yebin Liu, Qionghai Dai, and Diego Gutierrez. Intrinsic Video and Applications. *ACM Transactions on Graphics (ToG)*, 33(4):1–11, 2014. 2

[66] Weicai Ye, Xinyue Lan, Shuo Chen, Yuhang Ming, Xinyuan Yu, Chong Bao, Hujun Bao, Zhaopeng Cui, and Guofeng Zhang. PVO: Panoptic Visual Odometry. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 5

[67] Weicai Ye, Xinyue Lan, Shuo Chen, Yuhang Ming, Xinyuan Yu, Jinyu Li, Hujun Bao, Zhaopeng Cui, and Guofeng Zhang. DeFlowSLAM: Self-Supervised Scene Motion Decomposition for Dynamic Dense SLAM. 2022. 5

[68] Renjiao Yi, Ping Tan, and Stephen Lin. Leveraging Multi-view Image Sets for Unsupervised Intrinsic Image Decomposition and Highlight Separation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. 2

[69] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. PlenOctrees for Real-time Rendering of Neural Radiance Fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5752–5761, 2021. 5

[70] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. PixelNeRF: Neural Radiance Fields from One or Few Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021. 5

[71] Ye Yu and William AP Smith. Inverserendernet: Learning Single Image Inverse Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3155–3164, 2019. 2

[72] Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. PhySG: Inverse Rendering with Spherical Gaussians for Physics-based Material Editing and Relighting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5453–5462, 2021. 1, 2, 6, 7, 8, 3

[73] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. 2018. 7

[74] Xiuming Zhang, Pratul P Srinivasan, Boyang Deng, Paul Debevec, William T Freeman, and Jonathan T Barron. Nerfactor: Neural Factorization of Shape and Reflectance under An Unknown Illumination. *ACM Transactions on Graphics (TOG)*, 40(6):1–18, 2021. 2, 6, 7, 1, 3

[75] Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. Modeling Indirect Illumination for Inverse Rendering. *arXiv preprint arXiv:2204.06837*, 2022. 1, 2, 6, 7, 8, 3

[76] Shuang Zhao, Wenzel Jakob, and Tzu-Mao Li. Physics-based Differentiable Rendering: from Theory to Implementation. In *ACM SIGGRAPH 2020 Courses*, pages 1–30. ACM SIGGRAPH 2020 Courses, 2020. 2

[77] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison. In-place Scene Labelling and Understanding with Implicit Scene Representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15838–15847, 2021. 2, 5, 6, 7, 8, 1, 4

[78] Rui Zhu, Zhengqin Li, Janarbek Matai, Fatih Porikli, and Manmohan Chandraker. IRISformer: Dense Vision Transformers for Single-Image Inverse Rendering in Indoor Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2822–2831, 2022. 2, 7

[79] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. NICE-SLAM: Neural Implicit Scalable Encoding for SLAM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12786–12796, 2022. 5

IntrinsicNeRF: Learning Intrinsic Neural Radiance Fields for Editable Novel View Synthesis

Supplementary Material

In this supplementary document, we provide the framework of the semantic branch in IntrinsicNeRF (Sec. A), additional implementation details (Sec. B) and more experimental results (Sec. C) such as qualitative and quantitative results on Blender Object (Sec. C.1) and Replica Scene (Sec. C.2), and ablation study (Sec. C.3). We also present the applicability of our method on both synthetic and real-world data (Sec. C.4) and imagine the potential work with IntrinsicNeRF (Sec. D).

A. Semantic Branch in IntrinsicNeRF

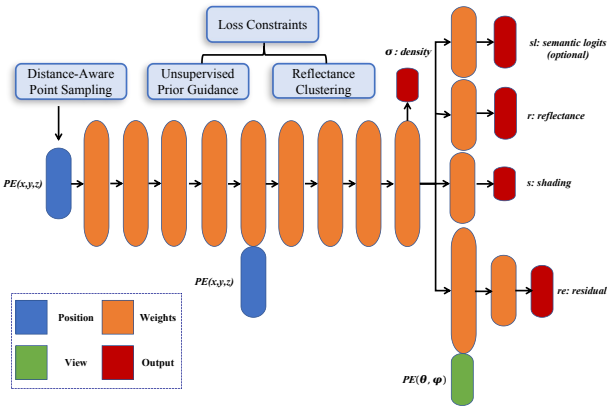

Figure A1: **IntrinsicNeRF Network.** IntrinsicNeRF takes 3D position $\mathbf{x}$=(x, y, z) as input, and outputs view-independent volume density σ, semantic logits sl, reflectance r, and shading s. While the residual term re additionally depends on the viewing direction $\mathbf{r}$=(θ, ϕ). Distance-aware point sampling, unsupervised prior, and reflectance clustering methods are used to train the network.

Inspired by [77], we extend IntrinsicNeRF to jointly encode appearance, geometry, and semantics by appending a segmentation renderer to the original IntrinsicNeRF, shown in Fig. A1. Following Semantic-NeRF [77], semantic segmentation is formalized as a view-independent function that recognized each pixel $\mathbf{x}$ as a semantic label distribution with softmax semantic logits $sl(\mathbf{x})$:

$$sl = F_\Theta(\mathbf{x}) \text{ [77]} \tag{A1}$$

where F_Θ is the MLP function. The predicted semantic logits $\hat{SL}(\mathbf{r})$ of each pixels can be written as:

$$\hat{SL}(\mathbf{r}) = \sum_{k=1}^{K} \hat{T}_k \, \alpha_k \, sl_k \text{ and } \hat{T}(t_k) = \exp\left(- \sum_{k'=1}^{k-1} \sigma_k \delta_k \right) \tag{A2}$$

where $\alpha_k = 1 - \exp(-\sigma_k \delta_k)$, and δ_k is the distance between two adjacent sampled points along the view direction $\mathbf{r}$. We present semantic logits as multi-class probabilities with the cross-entropy loss:

$$L_{sem} = - \sum_{\mathbf{r} \in \mathcal{R}} [p \log \hat{p}_c + p \log \hat{p}_f] \text{ [77]} \tag{A3}$$

where p is the multi-class semantic probabilities of the ground truth semantic map, while $\hat{p}_c$ and $\hat{p}_f$ are the probabilities of coarse and fine predictions, respectively.

B. Implementation Details

To make IntrinsicNeRF work, we jointly optimize the photometric loss, the semantic loss, the chromaticity loss, the reflectance sparsity constraint, and non-local reflectance consistency constraint, the shading smoothness, the residual constraint, and the reflectance clustering loss. The final loss function has been defined in the main manuscript. IntrinsicNeRF is implemented in PyTorch [49] and trained on an NVIDIA RTX3090-24G graphics card. The model is built on the top of NeRF [43] or SemanticNeRF [77] with the last three FC layers for intrinsic components. Limited to memory, the batch size of the rays is 1024. Tab. B1 shows acceptable clustering and the total training time of our method. Note that PhySG [72] and InvRender [75] can only perform on object-specific scenes, for they rely on good geometry.

Train and Test Time. For Blender Object, IntrinsicNeRF based on NeRF [43] renders 25 images with 400x400 resolution for clustering. For Replica Scene, IntrinsicNeRF based on SemanticNeRF [77] renders 180 images with 320x240 resolution for clustering. Implementing IntrinsicNeRF based on faster NeRF extensions is our future work.

Dataset	Method	Train	Test	Cluster
Blender	NeRFactor [74]	5.7d	5.65s	-
Blender	PhySG [72]	5.2h	2.92s	-
Blender	InvRender [75]	11.4h	14.25s	-
Blender	NeRF [43]	5.4h	4.59s	-
Blender	Ours	6.5h	5.35s	39s
Replica	Semantic-NeRF [77]	13.5h	2.50s	-
Replica	Ours	17.5h	2.79s	220s

Table B1: **Time Comparison.** We show the total training time, the average synthesis time of each frame, and the average clustering time of our method, where **the clustering is performed every 10K training iterations.** All run on a single RTX3090.

Figure C2: **Qualitative Comparison Results of Reflectance and Rendering with Previous work on Bleneder Objects.** The top 4 rows represent the sample of our dataset and the bottom 4 rows represent the sample of Invrender dataset. Our method can perform reflectance estimation and novel view synthesis on both datasets well, while Invrender [75] fails to do that on our dataset. N/A means failure.

C. More Experimental Results

C.1. Comparison Results on Blender Object

We present the detailed quantitative results on Tab. C2 and Tab. C3, compared with intrinsic decomposition methods and neural rendering methods. Our full model is superior to existing traditional intrinsic decomposition methods such as USI3D [35], IIW [3], CGIntrinsic [31] and reaches comparable results with Invrender [75] in intrinsic decomposition on Invrender dataset, shown in Fig. C2. Furthermore, our intrinsic neural radiance field scene representation enhances reconstructing objects with complex shapes and textures on our dataset, while Invrender fails to make it. The qualitative results of IntrinsicNeRF on Blender Object are shown in Fig. C3. However, our method also falls into some local optima in lego tracks (see Fig. C5), due to the inherent property of the intrinsic decomposition, failing to handle the black regions. Meanwhile, when the scenario does not conform to unsupervised prior, it will struggle to obtain the correct decomposition results, as shown in Fig. C2 (Hotdog, Chair in Ours column).

C.2. Comparison Results on Replica Scene

Tab. C4 shows the complete quantitative results on Replica Scene for novel view synthesis and semantic segmentation. We achieve comparable results with Semantic-NeRF [77] while giving the ability to model the underlying properties of scenes. Fig. C4 shows the qualitative results

Method	Albedo (Lego)					View Synthsis (Lego)			Albedo (Ficus)					View Synthsis (Ficus)		
	PSNR ↑	SSIM ↑	LPIPS ↓	MSE ↓	LMSE ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	MSE ↓	LMSE ↓	PSNR ↑	SSIM ↑	LPIPS ↓
IIW [3]	21.3080	0.8840	0.1255	0.0075	0.0355	-	-	-	19.4159	0.9145	0.0803	0.0110	0.1330	-	-	-
CGIntrinsic [31]	18.6028	0.8683	0.1454	0.0123	0.0363	-	-	-	22.0665	**0.9408**	0.0513	0.0052	0.1298	-	-	-
USI3D [35]	18.2291	0.8822	0.1282	0.0146	0.0332	-	-	-	16.2838	0.9253	0.0746	0.0230	0.0995	-	-	-
NeRFactor [74]	**22.5591**	**0.9250**	**0.0875**	**0.0034**	**0.0262**	17.6665	0.8263	0.1504	19.6809	0.9107	0.0488	0.0104	0.0874	21.3010	0.9053	0.0678
PhySG [72]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Invrender [75]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
NeRF [43]	-	-	-	-	-	**29.5691**	**0.9331**	**0.0268**	-	-	-	-	-	**29.4080**	**0.9609**	**0.0155**
baseline	11.9473	0.7669	0.2399	0.0522	0.2398	29.4163	0.9326	0.0280	23.0957	0.9229	0.0420	0.0045	0.1158	29.3302	0.9597	0.0158
baseline+w/prior	18.3652	0.8832	0.1515	0.0136	0.0615	29.1918	0.9300	0.0313	19.3838	0.9232	0.0606	0.0112	0.0933	29.0722	0.9588	0.0170
Ours	19.0001	0.9046	0.1288	0.0116	0.0647	29.1526	0.9283	0.0308	**23.3383**	0.9402	**0.0325**	**0.0042**	**0.0676**	28.9046	0.9576	0.0175

Method	Albedo (Chair2)					View Synthsis (Chair2)			Albedo (Drums)					View Synthsis (Drums)		
	PSNR ↑	SSIM ↑	LPIPS ↓	MSE ↓	LMSE ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	MSE ↓	LMSE ↓	PSNR ↑	SSIM ↑	LPIPS ↓
IIW [3]	24.2352	0.9410	0.0913	0.0035	0.0133	-	-	-	17.1604	0.8918	0.1553	0.0188	0.1091	-	-	-
CGIntrinsic [31]	15.9210	0.9070	0.1363	0.0259	0.0265	-	-	-	17.1604	0.8918	0.1553	0.0188	0.1091	-	-	-
USI3D [35]	23.0661	0.9303	0.1092	0.0045	0.0108	-	-	-	16.8267	0.8835	0.1588	0.0188	0.0711	-	-	-
NeRFactor [74]	21.5867	0.9266	0.1680	0.0056	0.0203	25.5135	0.8919	0.1285	**21.9491**	0.9059	**0.1176**	**0.0059**	**0.0438**	20.6880	0.8733	0.1185
PhySG [72]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Invrender [75]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
NeRF [43]	-	-	-	-	-	**30.1428**	**0.9448**	**0.0301**	-	-	-	-	-	**24.4357**	**0.9205**	**0.0590**
baseline	11.0799	0.8387	0.2025	0.0810	0.1802	30.0731	0.9436	0.0304	13.3059	0.8301	0.2110	0.0426	0.2036	24.2220	0.9172	0.0614
baseline+w/prior	27.1114	0.9406	0.0897	0.0015	0.0067	29.7973	0.9406	0.0368	18.9980	0.9089	0.1845	0.0117	0.0537	24.1918	0.9188	0.0625
Ours	**28.0020**	**0.9486**	**0.0731**	**0.0011**	**0.0054**	29.6453	0.9388	0.0383	19.9305	**0.9133**	0.1555	0.0093	0.0518	24.0949	0.9182	0.0620

Table C2: **Quantitative Evaluations on Our dataset.** Bold indicates best and underline indicates second best. - means failure.

Method	Albedo (Jugs)					View Synthsis (Jugs)			Albedo (Chair)					View Synthsis (Chair)		
	PSNR ↑	SSIM ↑	LPIPS ↓	MSE ↓	LMSE ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	MSE ↓	LMSE ↓	PSNR ↑	SSIM ↑	LPIPS ↓
IIW [3]	15.2941	0.9105	0.1188	0.0320	0.0238	-	-	-	25.8220	**0.9337**	0.0620	0.0019	0.0091	-	-	-
CGIntrinsic [31]	19.2596	0.9313	0.1066	0.0086	0.0220	-	-	-	21.1657	0.9140	0.0855	0.0070	0.0098	-	-	-
USI3D [35]	18.4617	0.9242	0.0780	0.0147	0.0249	-	-	-	24.5503	0.9290	0.0744	0.0020	**0.0070**	-	-	-
NeRFactor [74]	19.1639	0.9275	0.0911	0.0116	**0.0215**	26.0967	0.9492	0.0430	22.0620	0.9208	0.1287	0.0014	0.0089	22.1625	0.9294	0.0876
PhySG [72]	24.6498	0.9427	0.0790	0.0034	0.0860	24.6221	0.9544	0.0609	24.9832	0.9168	0.0877	0.0024	0.0262	25.7197	0.9320	0.0710
Invrender [75]	24.8413	**0.9508**	**0.0361**	0.0033	0.0427	29.5990	0.9654	0.0266	**29.4776**	0.9285	**0.0574**	**0.0010**	0.0089	31.3660	0.9444	0.0464
NeRF [43]	-	-	-	-	-	**35.4846**	0.9796	0.0165	-	-	-	-	-	32.5685	0.9436	0.0427
baseline	21.6691	0.8750	0.0773	0.0065	0.4158	35.2488	**0.9800**	**0.0155**	14.8468	0.8679	0.1271	0.0277	0.1151	**34.1195**	**0.9522**	**0.0312**
baseline+w/prior	19.1960	0.9249	0.1136	0.0117	0.0331	35.0930	0.9769	0.0212	22.5096	0.9232	0.0875	0.0042	0.0156	32.7608	0.9445	0.0424
Ours	**25.7546**	0.9471	0.0661	**0.0025**	0.0308	35.0342	0.9769	0.0213	23.7306	0.9278	0.0854	0.0027	0.0110	32.6955	0.9441	0.0415

Method	Albedo (Air balloons)					View Synthsis (Air balloons)			Albedo (Hotdog)					View Synthsis (Hotdog)		
	PSNR ↑	SSIM ↑	LPIPS ↓	MSE ↓	LMSE ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	MSE ↓	LMSE ↓	PSNR ↑	SSIM ↑	LPIPS ↓
IIW [3]	22.4801	**0.9276**	**0.0571**	0.0040	**0.0087**	-	-	-	24.5176	0.9512	0.1009	0.0014	0.0062	-	-	-
CGIntrinsic [31]	20.6844	0.9083	0.0888	0.0066	0.0192	-	-	-	19.5237	0.9299	0.1176	0.0294	0.0054	-	-	-
USI3D [35]	19.2599	0.9119	0.0725	0.0088	0.0185	-	-	-	20.7564	0.9418	0.1297	0.0061	0.0084	-	-	-
NeRFactor [74]	17.5734	0.8770	0.1701	0.0063	0.0416	20.7204	0.9018	0.1096	20.8677	0.9372	0.1517	0.0044	0.0121	23.0737	0.9305	0.0885
PhySG [72]	22.7754	0.9080	0.0974	0.0035	0.0328	26.1276	0.9475	0.0781	21.0910	0.9248	0.1729	0.0042	0.0134	25.2207	0.9213	0.1115
Invrender [75]	**25.2053**	0.9155	0.0716	0.0026	0.0263	27.6636	0.9493	0.0779	**25.7069**	0.9570	**0.0637**	0.0020	0.0123	28.9192	0.9497	0.0513
NeRF [43]	-	-	-	-	-	**32.8084**	**0.9676**	**0.0224**	-	-	-	-	-	**34.2531**	**0.9697**	**0.0287**
baseline	15.2960	0.8601	0.1399	0.0241	0.1820	32.5626	0.9666	0.0251	13.4718	0.8517	0.1762	0.0432	0.0690	34.0833	0.9693	0.0292
baseline+w/prior	21.2049	0.9049	0.1148	0.0036	0.0214	32.3400	0.9661	0.0254	24.0375	0.9581	0.1184	0.0024	0.0042	33.7700	0.9678	0.0325
Ours	21.9558	0.9116	0.1036	**0.0023**	0.0235	32.2197	0.9648	0.0269	25.6160	**0.9620**	0.0967	**0.0008**	**0.0038**	34.0375	0.9662	0.0325

Table C3: **Quantitative Evaluations on Invrender dataset.** Bold indicates best and underline indicates second best. - means failure.

of IntrinsicNeRF on Replica Scene.

C.3. Ablation Studies

We show more ablation study results in Fig. C5 on Blender Object and in Fig. C7 on Replica Scene. The reflectance estimated by the baseline method is more stochastic and unstable. While adding the intrinsic prior, the network output is plausible. The adaptive reflectance iterative clustering method can make the reflectance regions of the same material cluster together but may lose some distinguishable boundaries in Replica Scene. We also show the quantitative comparison results of Blender Object in Tab. C2 and Tab. C3. The comparison results demonstrate that unsupervised prior and clustering can help to improve the intrinsic decomposition, but may decrease the performance of view synthesis slightly. Fig. C7 shows hierarchical clustering method can retain the boundaries and still yields more plausible results.

C.4. Applications

We show the applicability of IntrinsicNeRF on real-time scene recoloring, illumination variation, and editable novel view synthesis. We have also developed a convenient editing software, to facilitate the user to perform object or scene editing, shown in Fig. C6.

Real-Time Scene Recoloring. The reflectance predicted by the IntrinsicNeRF network is saved as [Semantic category, reflectance category], and the last iteration of the hierarchical iterative clustering method will save the reflectance categories in all semantic categories of the whole scene. Therefore, the [Semantic category, reflectance category] label can be used to quickly find the reflectance value of each pixel point. Based on this representation, we can perform scene recoloring in real-time, just by simply modifying the color of a certain reflectance category, the reflectance values of all pixels in the multi-view images belonging to that

Method	Office 0				Office 1				Office 2				Office 3			
	PSNR ↑	SSIM ↑	LPIPS ↓	mIoU↑	PSNR ↑	SSIM ↑	LPIPS ↓	mIoU↑	PSNR ↑	SSIM ↑	LPIPS ↓	mIoU↑	PSNR ↑	SSIM ↑	LPIPS ↓	mIoU↑
Semantic-NeRF [77]	33.9807	0.9294	0.0631	0.9802	35.6869	0.9516	0.0689	0.9816	30.8175	0.9296	0.0755	0.9777	30.2418	0.9238	0.0694	0.9678
Ours	33.9734	0.9292	0.0666	0.9793	35.4500	0.9532	0.0680	0.9809	30.2827	0.9231	0.0843	0.9753	29.9553	0.9179	0.0741	0.9619

Method	Office 3				Room 0				Room 1				Room 2			
	PSNR ↑	SSIM ↑	LPIPS ↓	mIoU↑	PSNR ↑	SSIM ↑	LPIPS ↓	mIoU↑	PSNR ↑	SSIM ↑	LPIPS ↓	mIoU↑	PSNR ↑	SSIM ↑	LPIPS ↓	mIoU↑
Semantic-NeRF [77]	31.4142	0.9154	0.1039	0.9531	27.2094	0.8108	0.1669	0.9712	28.5790	0.8215	0.1719	0.9802	29.8863	0.8814	0.1331	0.9681
Ours	30.9201	0.9106	0.1098	0.9537	27.0812	0.8063	0.1698	0.9680	28.1852	0.8048	0.2056	0.9769	29.7873	0.8809	0.1343	0.9651

Table C4: **Quantitative Evaluations on Replica Scene.** We achieve comparable results with Semantic-NeRF in novel view synthesis and semantic segmentation.

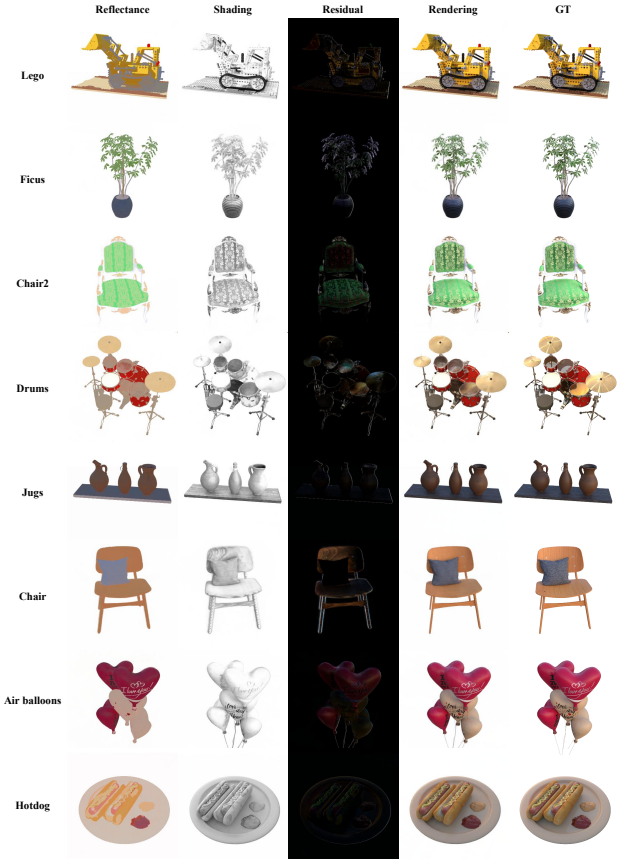

Figure C3: **Qualitative Results of IntrinsicNeRF on Blender Object.** From left to right are reflectance, shading, residual term, rendering result, and original image. In addition to the Lambertian assumption, our method can also simulate specular reflections or metallic materials.

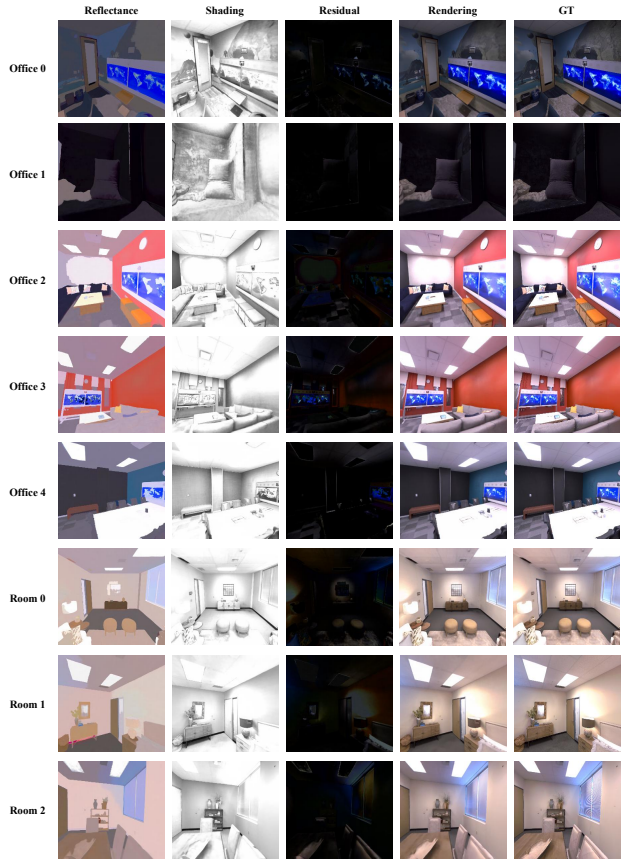

Figure C4: **Qualitative Results of IntrinsicNeRF on Replica Scene.** From left to right are reflectance, shading, residual term, rendering result, and original image. In addition to the Lambertian assumption, our method can also simulate specular reflections or metallic materials.

category can be modified at the same time, and then the recolored images can be reconstructed using the modified reflectance with the original shading and residual through Eq. 2. Fig. C8 shows the scene recoloring samples on Blender Object and Replica Scene. Our method can support semantic recoloring with a simple user click and selected modified color. We also perform scene recoloring on the real-world data to show the generalization ability of our method, shown in Fig. C11.

Illumination Variation. Since our IntrinsicNeRF can decompose residual terms besides Lambertian assumptions, which may be properties such as specular illumination, we can adjust its overall brightness directly by a multiplicative factor. Specifically, users only need to adjust the sliding buttons of the video editing software and the overall brightness will be modified. We can enhance the light or diminish

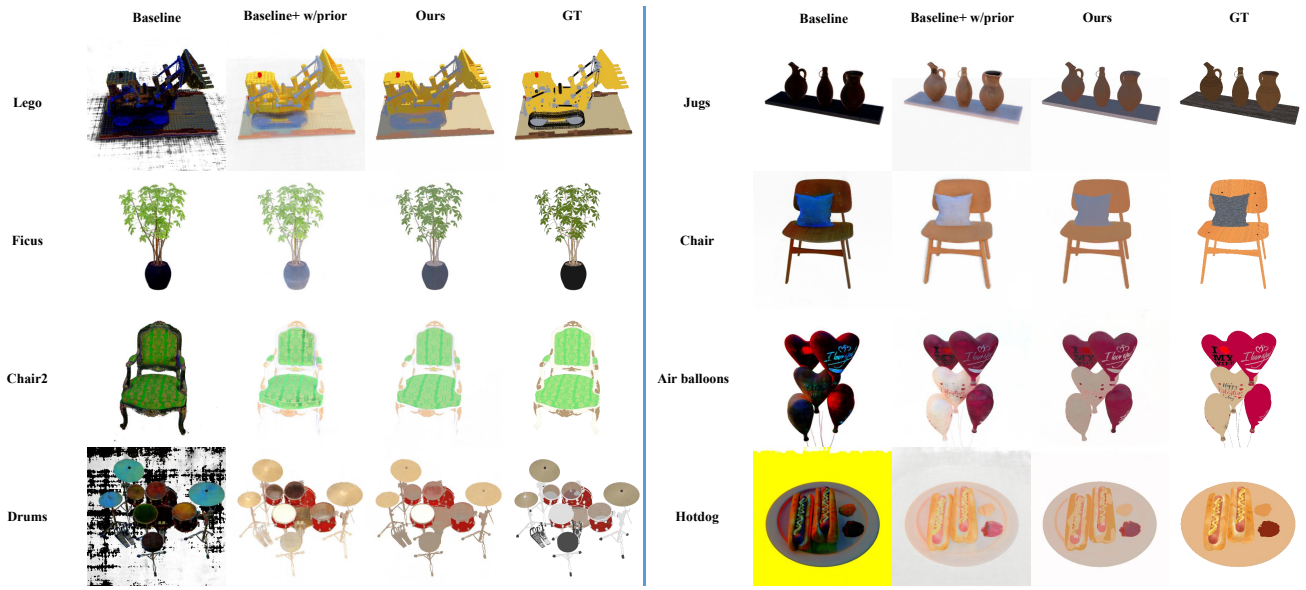

Figure C5: **Ablation study of Reflectance Estimation on Blender Object Dataset.** Left: our dataset, right: Invrender dataset. The reflectance estimation of the baseline method is stochastic and unstable, while the intrinsic prior makes the optimization of the network traceable. Our final model achieves more plausible albedo results.

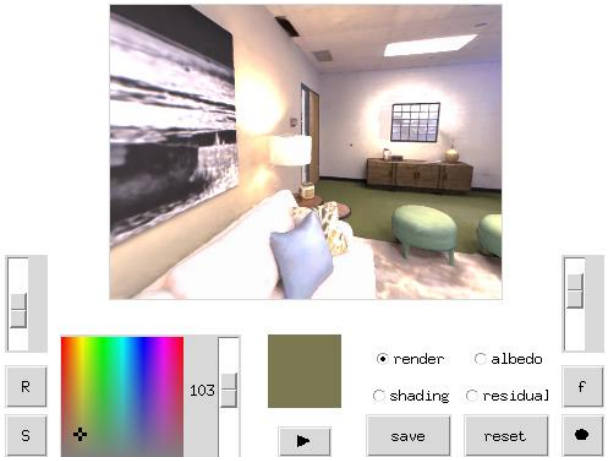

Figure C6: **Video Editing Software.** The software includes a palette for albedo, a sliding bar for shading, residual layers, as well as buttons for playing or recording view synthesis, reset, etc.

it, to see the effect of different light intensities, as shown in Fig. C9. We also perform illumination variation on the real-world data to show the generalization ability of our method, shown in Fig. C12.

Editable Novel View Synthesis. Our IntrinsicNeRF gives the NeRF [43] the ability to model additional fundamental properties of the scene, and the original novel view synthesis functionality is retained. As shown in Fig. C10, the effects of our video editing application above such as scene recoloring can be applied to the editable novel view synthesis, maintaining consistency. We also perform editable view synthesis on the real-world data to show the generalization ability of our method, shown in Fig. C13. Please refer to the supplementary video for more details.

Video Editing Software. As shown in Fig. C6, we visualize the interface of our video editing software, which contains controls for the color palette for the albedo layer, two sliding bars for shading and residual layers, as well as buttons for playing or recording view synthesis and reset, etc. Due to IntrinsicNeRF with hierarchical clustering and indexing representation, our software can support real-time augmented video editing.

D. Future Work

Although the intrinsic neural radiance fields give NeRF the ability to model the basic properties of scenes (object-level and scene-level) (e.g., albedo, shading, illumination, etc.), IntrinsicNeRF retains other shortcomings of NeRF. Given the high degree of integration of our approach with NeRF, NeRF extensions can be seamlessly incorporated into our IntrinsicNeRF, such as NeRF in the wild [43, 11, 57], dynamic NeRF [48, 50, 30, 67], fast NeRF [45, 16, 9, 69], NeRF with generalization [70, 62, 10, 25], generative NeRF [60, 53], NeRF with panoptic segmentation [24, 66], NeRF-based SLAM [56, 79, 44], Geometry and Texture Editing with NeRF [13] etc, which will be helpful to the

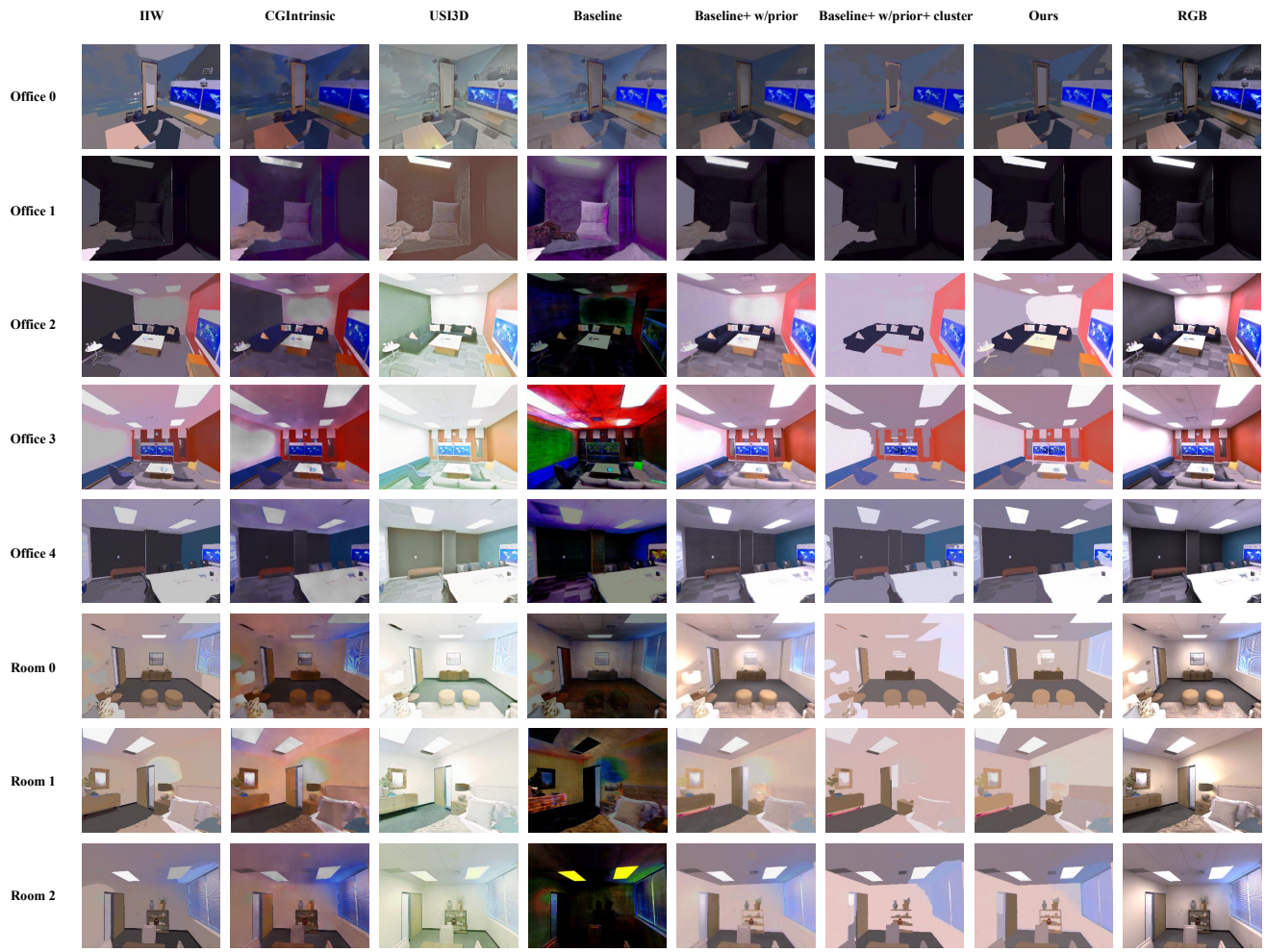

Figure C7: **Qualitative Reflectance Comparisons with Previous Methods on Replica Scene.** Experiments demonstrate the progressive facilitation effect of our different variants. Compared with previous methods, our final method achieves more plausible and consistent albedo estimation results, retaining the boundaries of objects, please refer to the supplementary video.

research community. In addition, although IntrinsicNeRF can yield plausible results in most scenarios, it always faces the challenge that complex scenarios, such as lego, chair, and hotdog, do not conform to unsupervised intrinsic prior. A refinement method based on intrinsic decomposition prediction is required.

Another more interesting direction is how to unify intrinsic decomposition and inverse rendering to construct a hierarchical representation of the intrinsic properties of the scene.

Since our approach yields consistent intrinsic video decomposition results, IntrinsicNeRF can improve the performance of the intrinsic decomposition method by providing more datasets with pseudo-Ground-Truth labels for the intrinsic decomposition task. We leave this as future work.

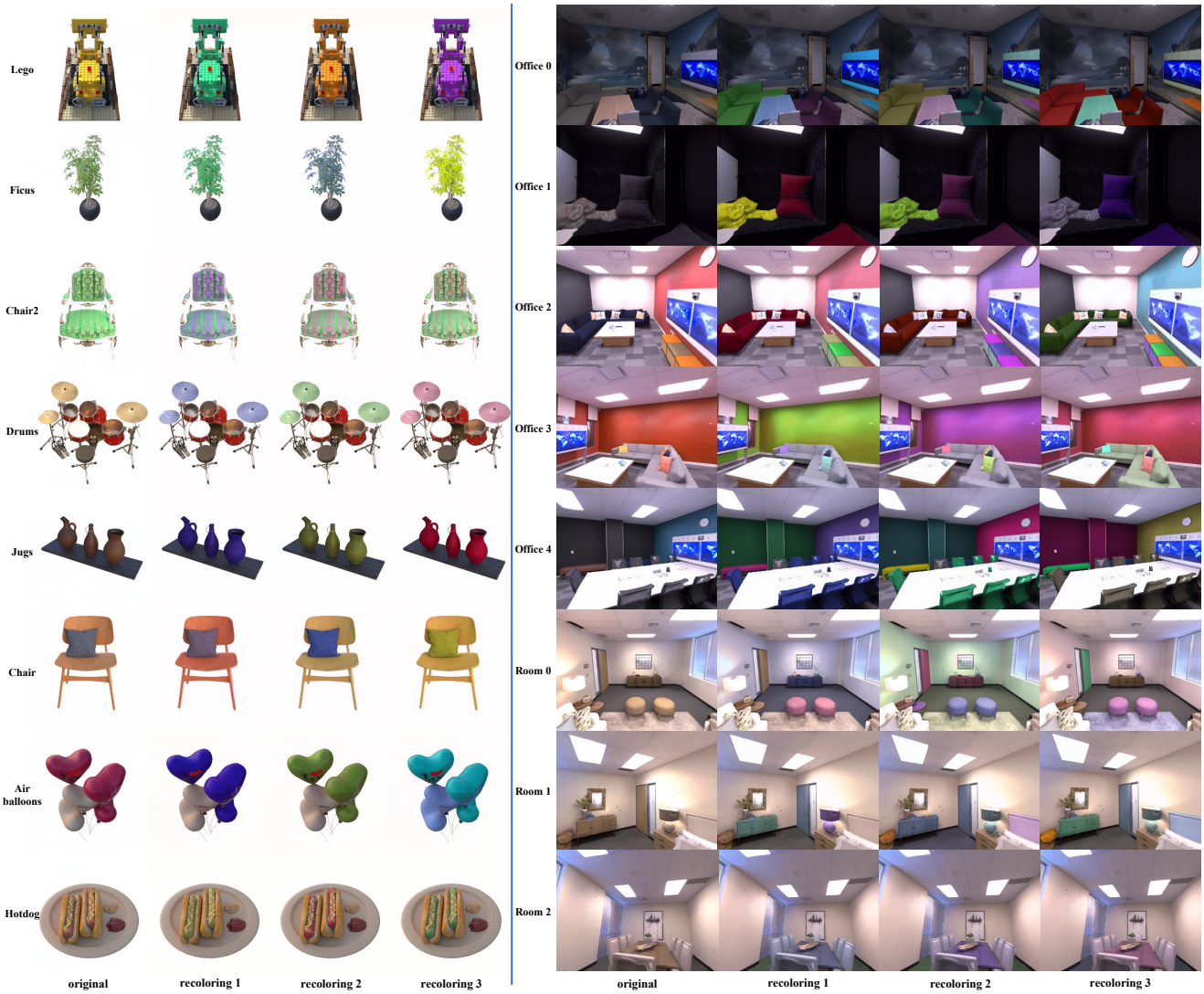

Figure C8: **Real-Time Scene Recoloring on Synthetic Data.** Our approach allows for real-time region-level scene recoloring on synthetic data with a simple user click and selected modified color.

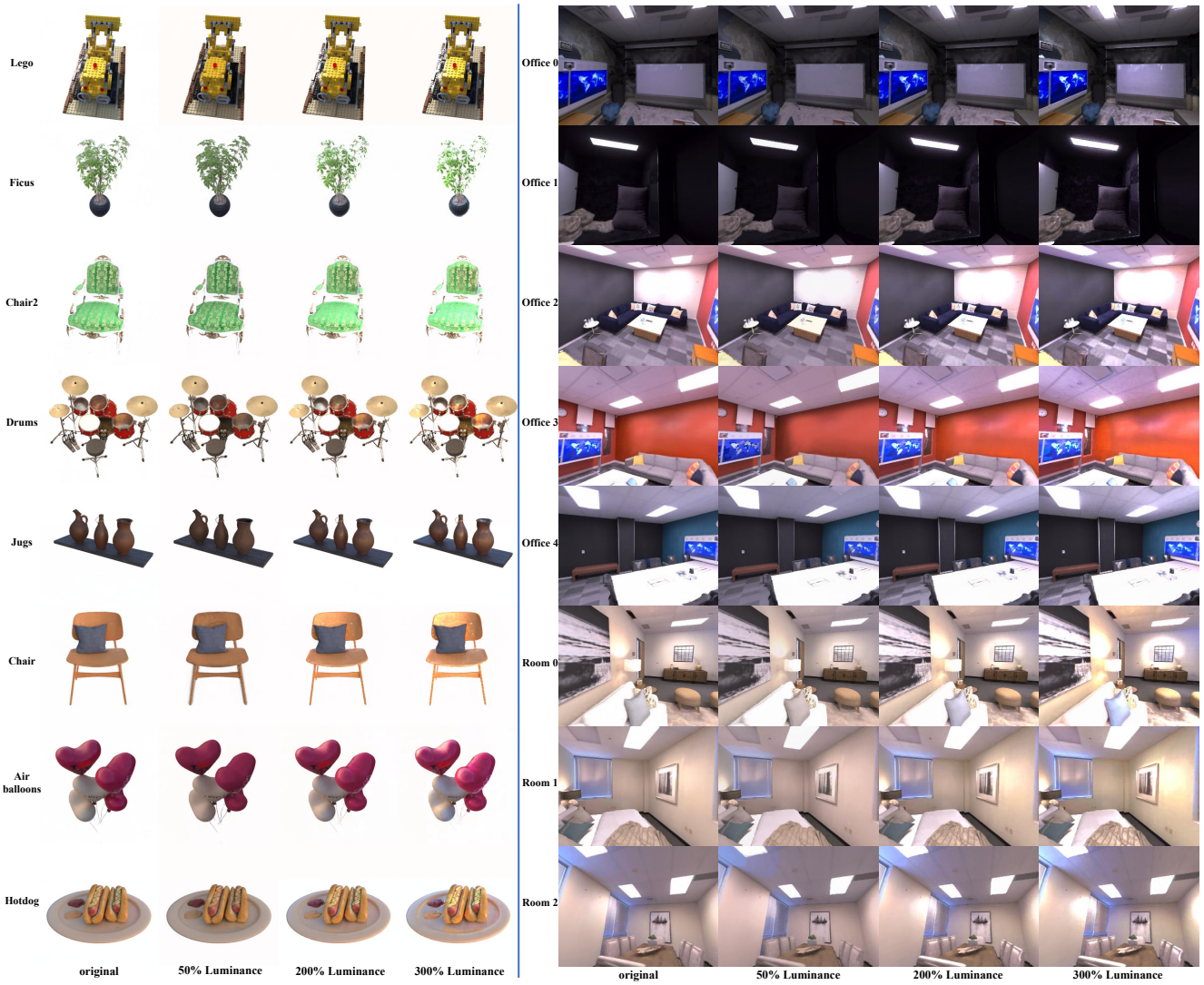

Figure C9: **Illumination Variation on Synthetic Data.** Left: Blender Object Dataset, Right: Replica Scene. We can adjust the brightness of the illumination, which can be applied to the ceiling, sofa, walls, and doors (such as Room 0). Please refer to the supplementary video.

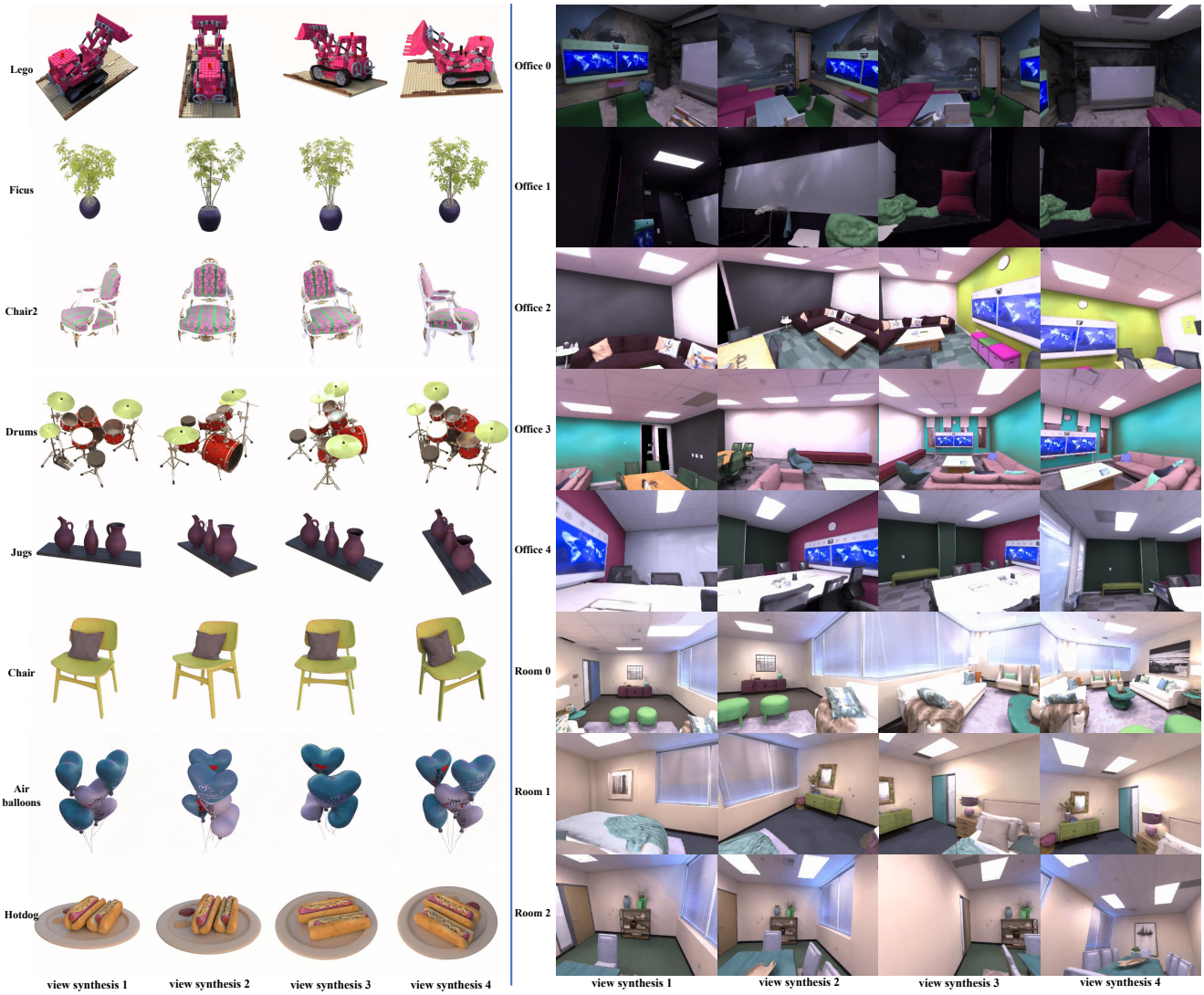

Figure C10: **Editable Novel View Synthesis on Synthetic Data.** Our method can support real-time video augmented editing applications with editable novel view synthesis. Here, we show the view synthesis results with scene recoloring. For more details, please refer to the supplementary video.

Figure C11: **Real-Time Scene Recoloring on Real-World Data.** Our approach allows for real-time region-level scene recoloring on real-world data with a simple user click and selected modified color.

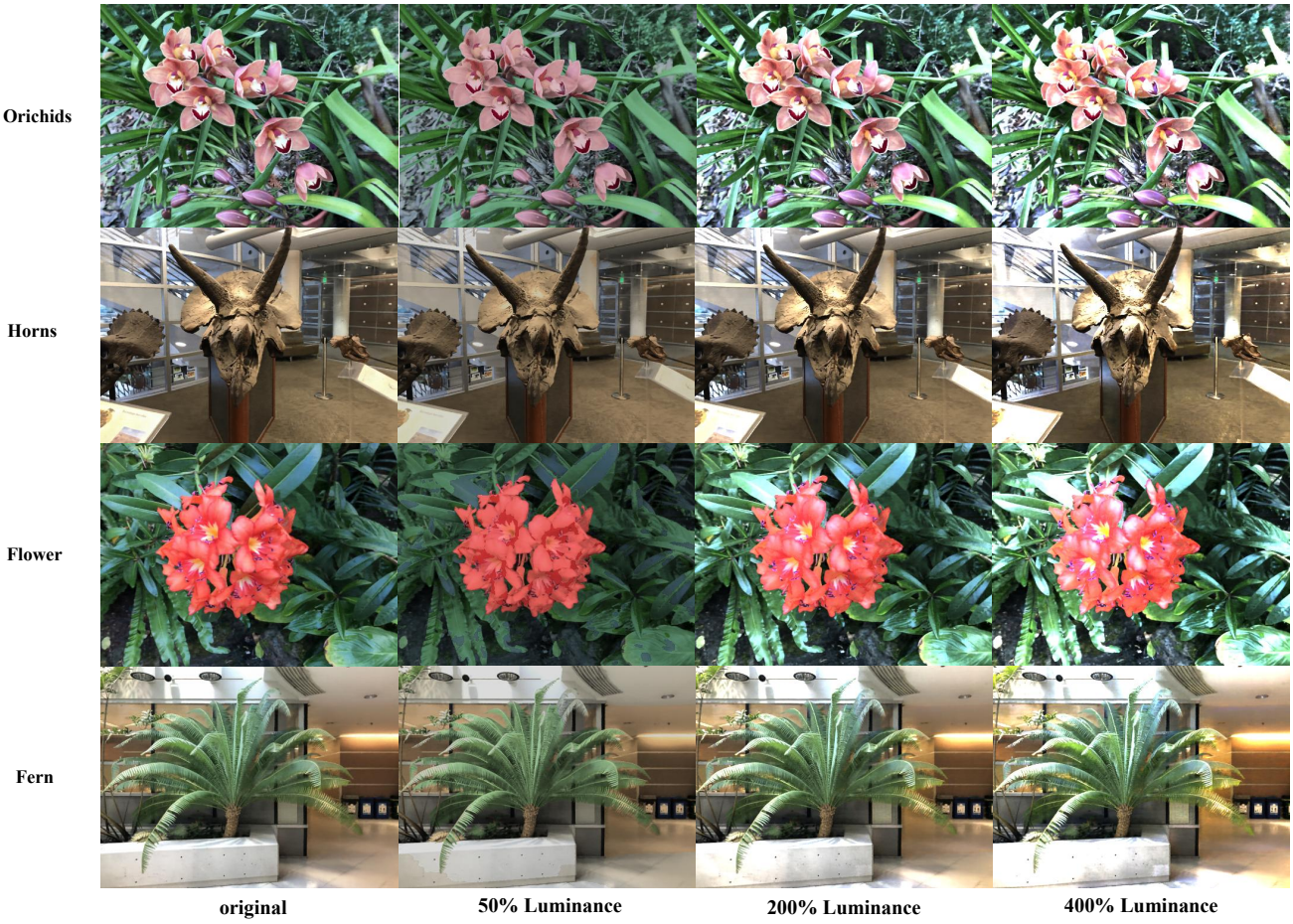

Figure C12: **Illumination Variation on Real-World Data.** We can adjust the brightness of the illumination on real-world data. Please refer to the supplementary video.

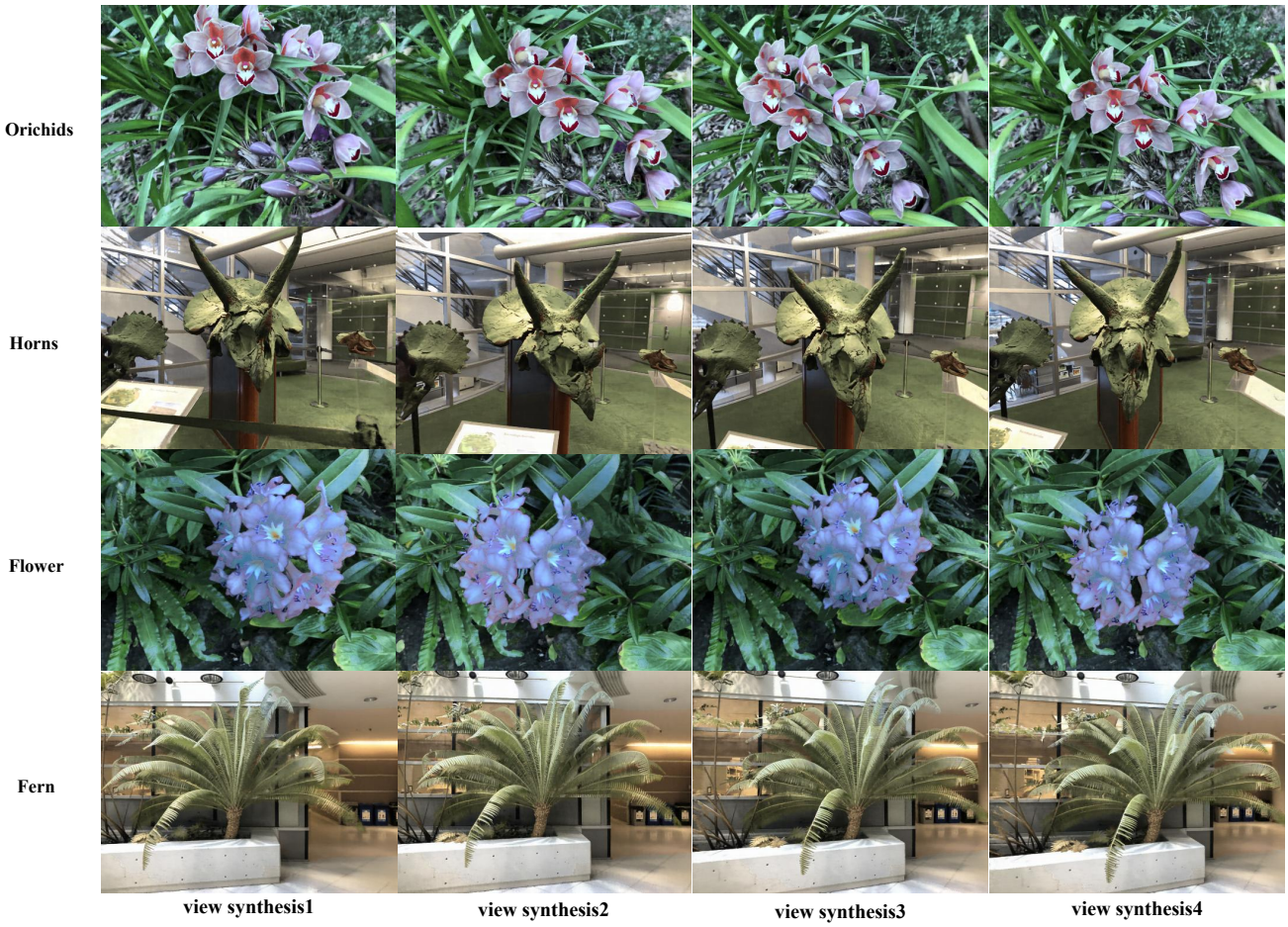

Figure C13: **Editable Novel View Synthesis on Real-World Data.** Our method can support real-time augmented editing applications with editable novel view synthesis.