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

Supplementary Document

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
https://zju3dv.github.io/intrinsic_nerf

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

A. Semantic Branch in IntrinsicNeRF

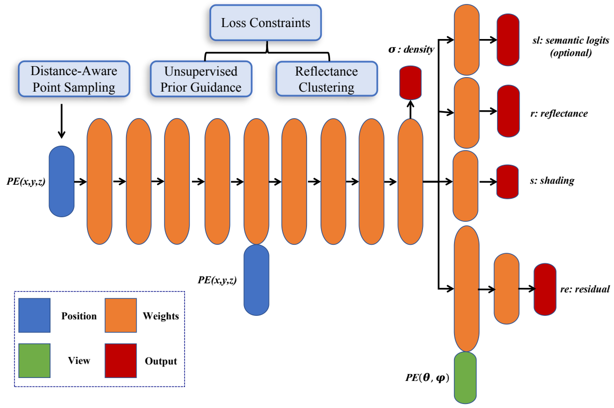


Figure A1: **IntrinsicNeRF Network.** IntrinsicNeRF takes 3D position $\mathbf{x}$=(x, y, z) as input, and outputs view-independent volume density σ, semantic logits sl, reflectance r, and shading s. While the residual term re additionally depends on the viewing direction $\mathbf{r}$=(θ, ϕ). Distance-aware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

Inspired by [8], 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 [8], semantic segmentation is formalized as a view-independent function that recognized each pixel $\mathbf{x}$ as a semantic label distribution with softmax semantic logits $sl(\mathbf{x})$:

$$sl = F_\Theta(\mathbf{x}), \tag{A1}$$

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 $\mathbf{r}$. Following Semantic-NeRF [8], we present semantic logits as multi-class probabilities with the cross-entropy loss:

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

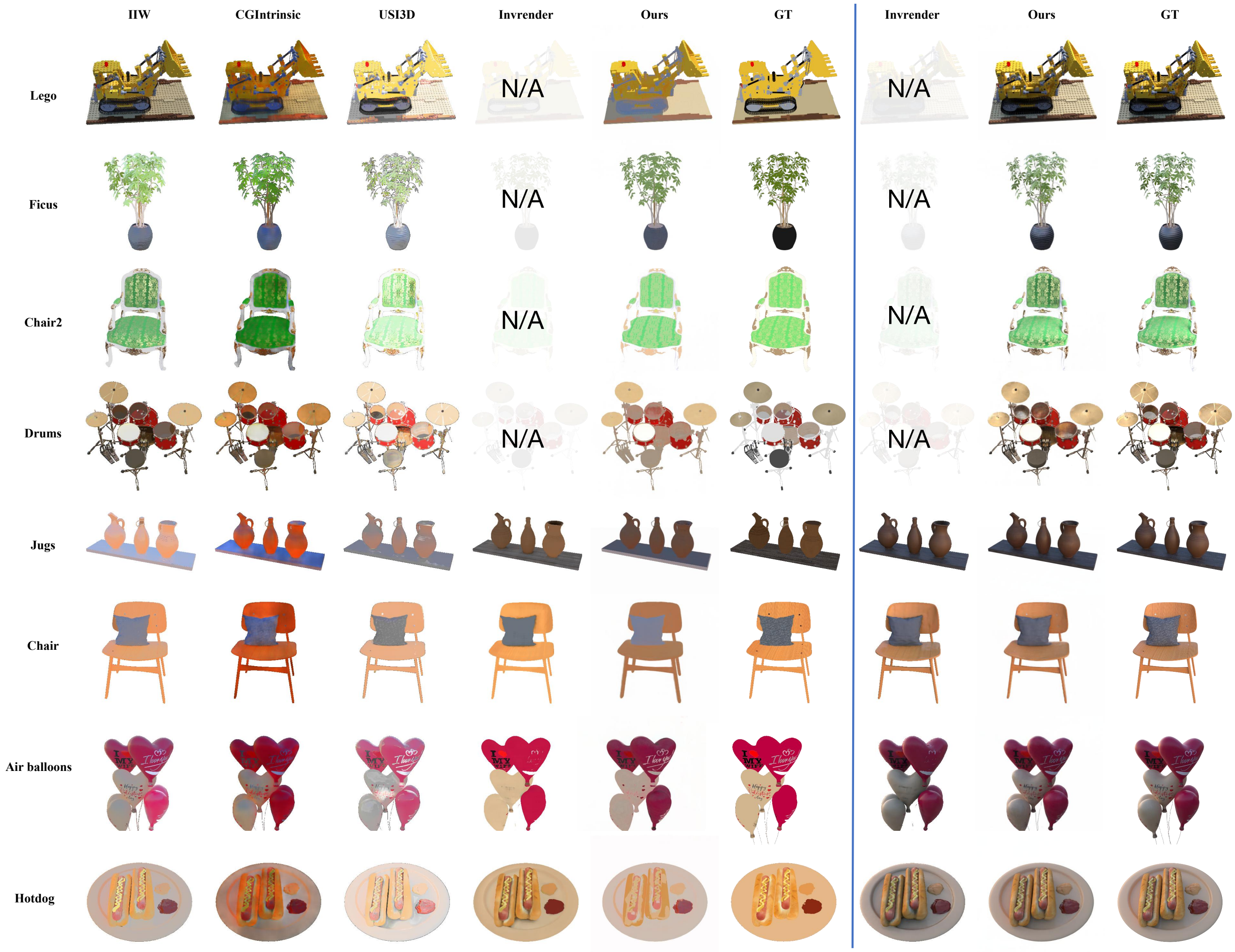
where p is the multi-class semantic probabilities of the ground truth semantic map, while $\hat{p}_c$ and $\hat{p}_f$ are the probabilities of coarse and fine predictions, respectively.

B. More Experimental Results

B.1. Comparison on the Blender Object Dataset

We present the detailed quantitative results on Tab. B1 and Tab. B2, compared with intrinsic decomposition methods and neural rendering methods. Our full model is superior to existing traditional intrinsic decomposition methods such as USI3D [3], IIW [1], CGIntrinsic [2] and reaches comparable results with Invrender [7] in intrinsic decomposition on Invrender dataset, shown in Fig. B2. Furthermore,

[*] Equal contribution.
[†] Corresponding author.



(a) Reflectance Comparison

(b) Rendering Comparison

Figure B2: **Qualitative Comparison Results of Reflectance and Rendering with Previous Work on the Blender Object Dataset.** The top 4 rows represent the sample of our dataset and the bottom 4 rows represent the sample of the Invrender dataset. Our method can perform reflectance estimation and novel view synthesis on both datasets well, while Invrender [7] fails to do that on our dataset. N/A means failure.

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 the Blender Object dataset are shown in Fig. B3. However, our method also falls into some local optima in Lego tracks (see Fig. B5), 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. B2 (Hotdog, Chair in Ours column).

B.2. Comparison on the Replica Scene Dataset

Tab. B3 shows the complete quantitative results on the Replica Scene dataset for novel view synthesis and semantic segmentation. We achieve comparable results with Semantic-NeRF [8] while giving the ability to model the underlying properties of scenes. Fig. B4 shows the qualitative results of IntrinsicNeRF on the Replica Scene dataset.

B.3. Ablation Studies

We show more ablation study results in Fig. B5 on the Blender Object dataset and in Fig. B7 on the Replica Scene dataset. 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 the Replica Scene dataset. We also show the quantitative comparison

	Reflectance (Lego)					View Synthesis (Lego)			Reflectance (Ficus)					View Synthesis (Ficus)		
Method	PSNR ↑	SSIM ↑	LPIPS ↓	MSE ↓	LMSE ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	MSE ↓	LMSE ↓	PSNR ↑	SSIM ↑	LPIPS ↓
IIW [1]	21.3080	0.8840	0.1255	0.0075	0.0355	-	-	-	19.4159	0.9145	0.0803	0.0110	0.1330	-	-	-
CGIntrinsic [2]	18.6028	0.8683	0.1454	0.0123	0.0363	-	-	-	22.0665	**0.9408**	0.0513	0.0052	0.1298	-	-	-
USI3D [3]	18.2291	0.8822	0.1282	0.0146	0.0332	-	-	-	16.2838	0.9253	0.0746	0.0230	0.0995	-	-	-
NeRFactor [6]	**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 [5]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Invrender [7]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
NeRF [4]	-	-	-	-	-	**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

	Reflectance (Chair2)					View Synthesis (Chair2)			Reflectance (Drums)					View Synthesis (Drums)		
Method	PSNR ↑	SSIM ↑	LPIPS ↓	MSE ↓	LMSE ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	MSE ↓	LMSE ↓	PSNR ↑	SSIM ↑	LPIPS ↓
IIW [1]	24.2352	0.9410	0.0913	0.0035	0.0133	-	-	-	17.1604	0.8918	0.1553	0.0188	0.1091	-	-	-
CGIntrinsic [2]	15.9210	0.9070	0.1363	0.0259	0.0265	-	-	-	17.1604	0.8918	0.1553	0.0188	0.1091	-	-	-
USI3D [3]	23.0661	0.9303	0.1092	0.0045	0.0108	-	-	-	16.8267	0.8835	0.1588	0.0188	0.0711	-	-	-
NeRFactor [6]	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 [5]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Invrender [7]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
NeRF [4]	-	-	-	-	-	**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 B1: **Quantitative Evaluations on Our dataset.** Bold indicates best and underline indicates second best. - means failure.

	Reflectance (Jugs)					View Synthesis (Jugs)			Reflectance (Chair)					View Synthesis (Chair)		
Method	PSNR ↑	SSIM ↑	LPIPS ↓	MSE ↓	LMSE ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	MSE ↓	LMSE ↓	PSNR ↑	SSIM ↑	LPIPS ↓
IIW [1]	15.2941	0.9105	0.1188	0.0320	0.0238	-	-	-	25.8220	**0.9337**	0.0620	0.0019	0.0091	-	-	-
CGIntrinsic [2]	19.2596	0.9313	0.1066	0.0086	0.0220	-	-	-	21.1657	0.9140	0.0855	0.0070	0.0098	-	-	-
USI3D [3]	18.4617	0.9242	0.0780	0.0147	0.0249	-	-	-	24.5503	0.9290	0.0744	0.0020	**0.0070**	-	-	-
NeRFactor [6]	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 [5]	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 [7]	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 [4]	-	-	-	-	-	**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

	Reflectance (Air balloons)					View Synthesis (Air balloons)			Reflectance (Hotdog)					View Synthesis (Hotdog)		
Method	PSNR ↑	SSIM ↑	LPIPS ↓	MSE ↓	LMSE ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	MSE ↓	LMSE ↓	PSNR ↑	SSIM ↑	LPIPS ↓
IIW [1]	22.4801	**0.9276**	**0.0571**	0.0040	**0.0087**	-	-	-	24.5176	0.9512	0.1009	0.0014	0.0062	-	-	-
CGIntrinsic [2]	20.6844	0.9083	0.0888	0.0066	0.0192	-	-	-	19.5237	0.9299	0.1176	0.0294	0.0054	-	-	-
USI3D [3]	19.2599	0.9119	0.0725	0.0088	0.0185	-	-	-	20.7564	0.9418	0.1297	0.0061	0.0084	-	-	-
NeRFactor [6]	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 [5]	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 [7]	**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 [4]	-	-	-	-	-	**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 B2: **Quantitative Evaluations on Invrender dataset.** Bold indicates best and underline indicates second best. - means failure.

results of the Blender Object dataset in Tab. B1 and Tab. B2. 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. B7 shows hierarchical clustering method can retain the boundaries and still yields more plausible results.

B.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. B6.

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 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 in the main paper. Fig. B8 shows the scene recoloring samples on the Blender Object dataset and the Replica Scene dataset. 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 gener-

	Office 0				Office 1				Office 2				Office 3			
Method	PSNR ↑	SSIM ↑	LPIPS ↓	mIoU↑	PSNR ↑	SSIM ↑	LPIPS ↓	mIoU↑	PSNR ↑	SSIM ↑	LPIPS ↓	mIoU↑	PSNR ↑	SSIM ↑	LPIPS ↓	mIoU↑
Semantic-NeRF [8]	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

	Office 3				Room 0				Room 1				Room 2			
Method	PSNR ↑	SSIM ↑	LPIPS ↓	mIoU↑	PSNR ↑	SSIM ↑	LPIPS ↓	mIoU↑	PSNR ↑	SSIM ↑	LPIPS ↓	mIoU↑	PSNR ↑	SSIM ↑	LPIPS ↓	mIoU↑
Semantic-NeRF [8]	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 B3: **Quantitative Evaluations on the Replica Scene Dataset.** We achieve comparable results with Semantic-NeRF in novel view synthesis and semantic segmentation.

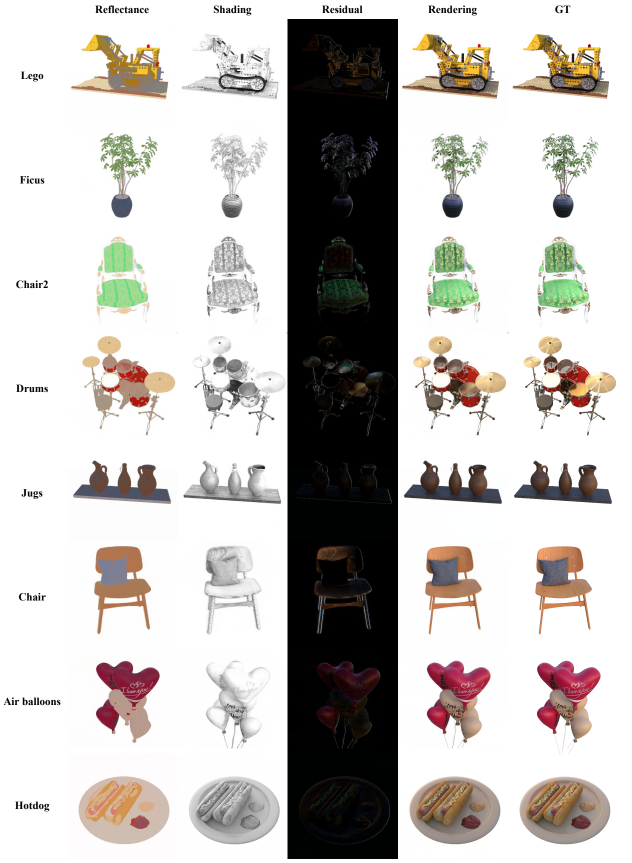


Figure B3: **Qualitative Results of IntrinsicNeRF on the Blender Object Dataset.** From left to right are reflectance, shading, residual term, rendering result, and original image. In addition to the Lambertian assumption, our method can also simulate glossy reflections or metallic materials.

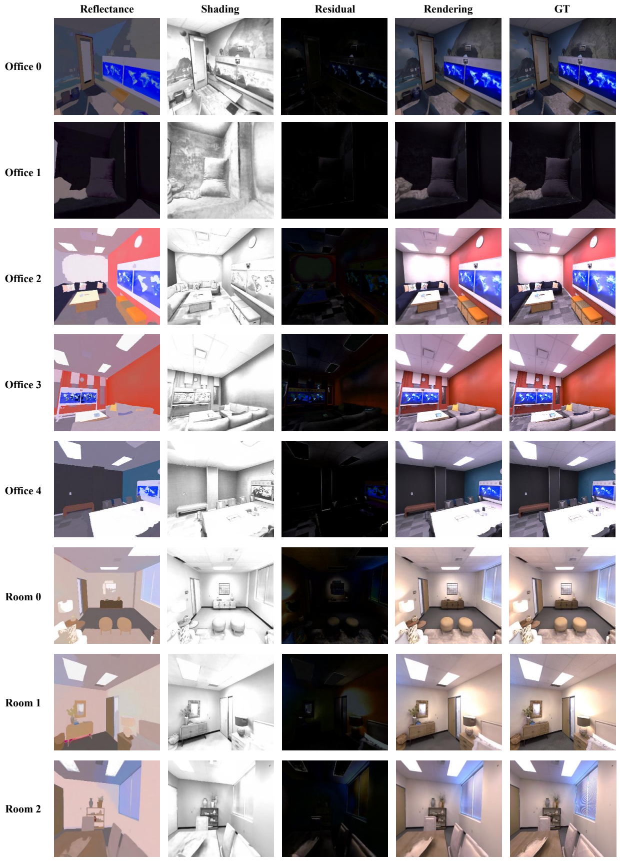


Figure B4: **Qualitative Results of IntrinsicNeRF on the Replica Scene Dataset.** From left to right are reflectance, shading, residual term, rendering result, and original image. In addition to the Lambertian assumption, our method can also simulate glossy reflections or metallic materials.

alization ability of our method, shown in Fig. B11.

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 it, to see the effect of different light intensities, as shown in Fig. B9. We also perform illumination variation on the real-world data to show the generalization ability of our method, shown in Fig. B12.

Editable Novel View Synthesis. Our IntrinsicNeRF gives the NeRF [4] the ability to model additional fundamental

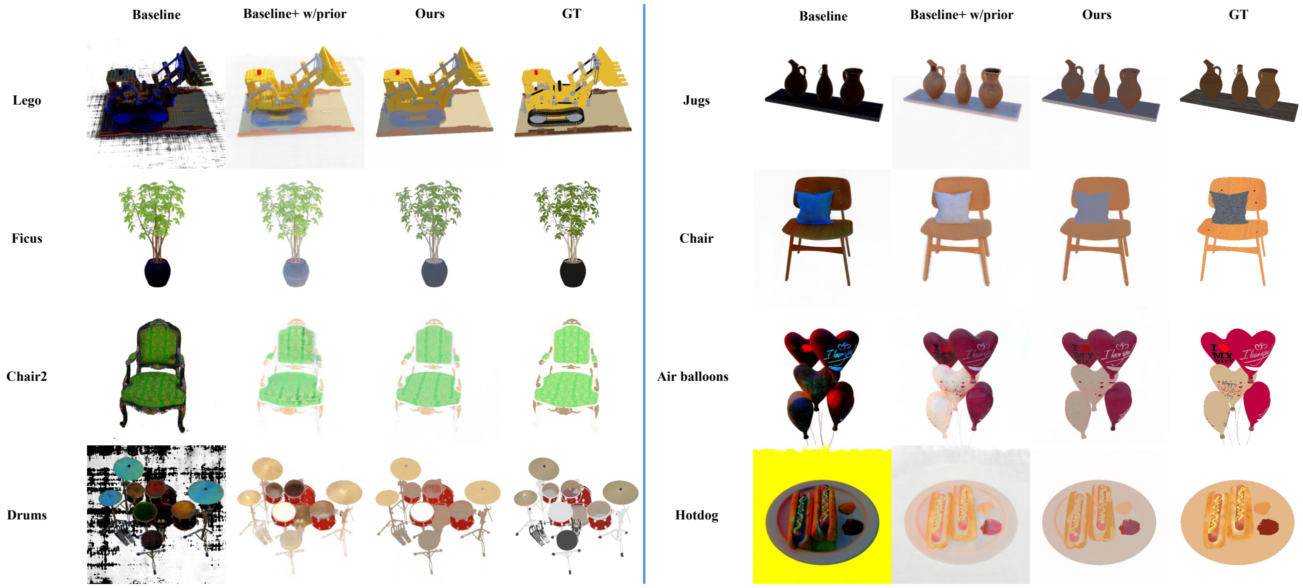


Figure B5: **Ablation study of Reflectance Estimation on the Blender Object Dataset.** Left: our dataset, right: Invrender dataset. The reflectance estimation of the baseline method is stochastic and unstable, while the intrinsic prior makes the optimization of the network traceable. Our final model achieves more plausible reflectance results.

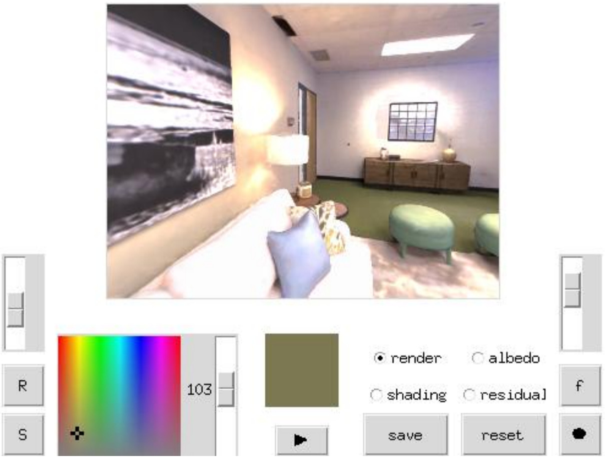


Figure B6: **Video Editing Software.** The software includes a palette for reflectance, a sliding bar for shading, residual layers, as well as buttons for playing or recording view synthesis, reset, etc.

properties of the scene, and the original novel view synthesis functionality is retained. As shown in Fig. B10, 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. B13. Please refer to the supplementary video for more details.

Video Editing Software. As shown in Fig. B6, we visualize the interface of our video editing software, which contains controls for the color palette for the reflectance 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.

References

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

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

[3] 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.

[4] 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 *Proceedings of the European conference on computer vision*, pages 405–421. Springer, 2020.

[5] Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. PhySG: Inverse Rendering with Spherical Gaussians for Physics-based Material Editing and Relighting.

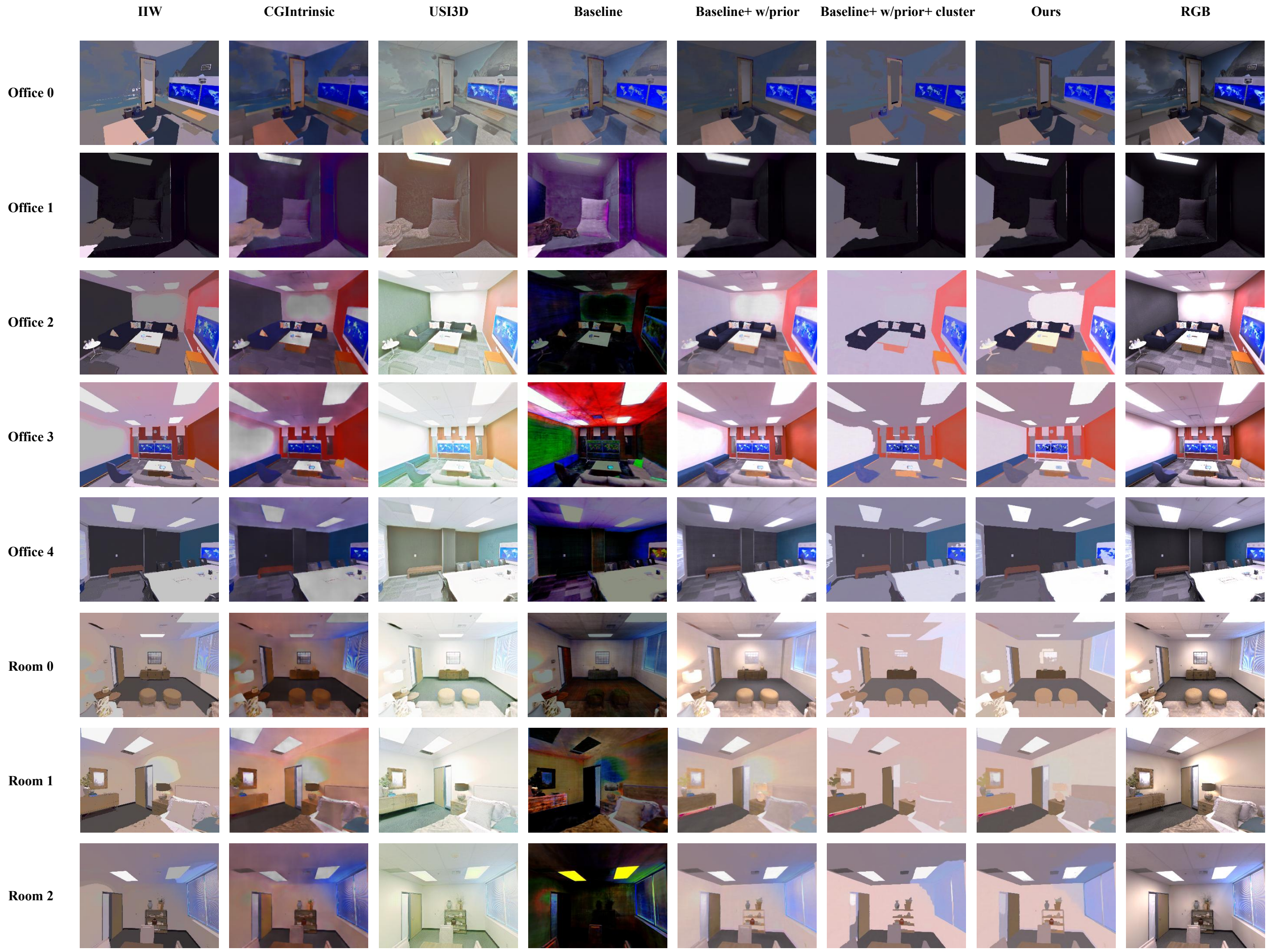


Figure B7: **Qualitative Reflectance Comparisons with Previous Methods on the Replica Scene Dataset.** Experiments demonstrate the progressive facilitation effect of our different variants. Compared with previous methods, our final method achieves more plausible and multi-view consistent reflectance estimation results, retaining the boundaries of objects, please refer to the supplementary video.

In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5453–5462, 2021.

[6] 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*, 40(6):1–18, 2021.

[7] 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.

[8] 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.

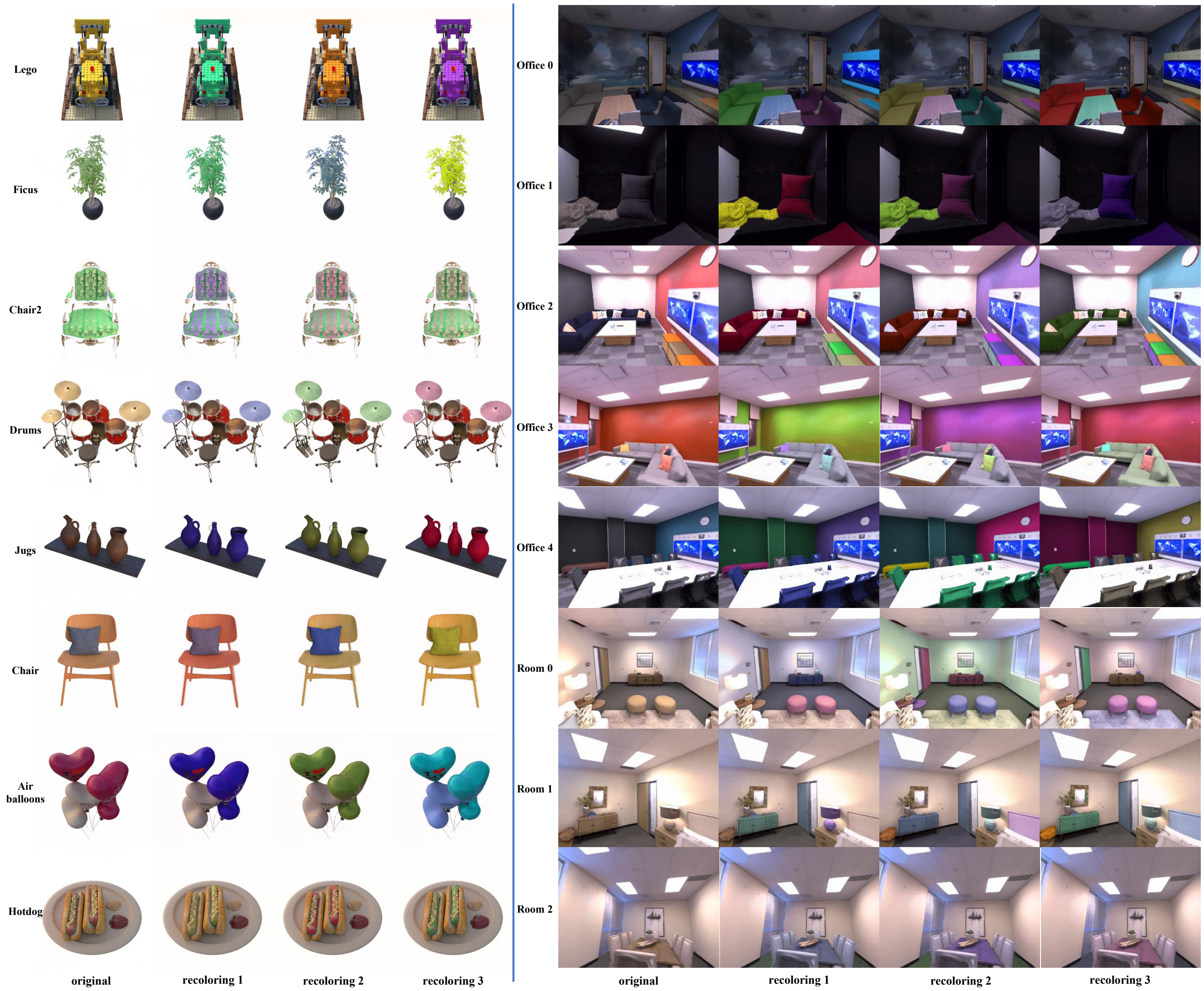


Figure B8: **Real-Time Scene Recoloring on Synthetic Data.** Our approach allows for real-time region-level scene recoloring on synthetic data with a simple user click and selected modified color.

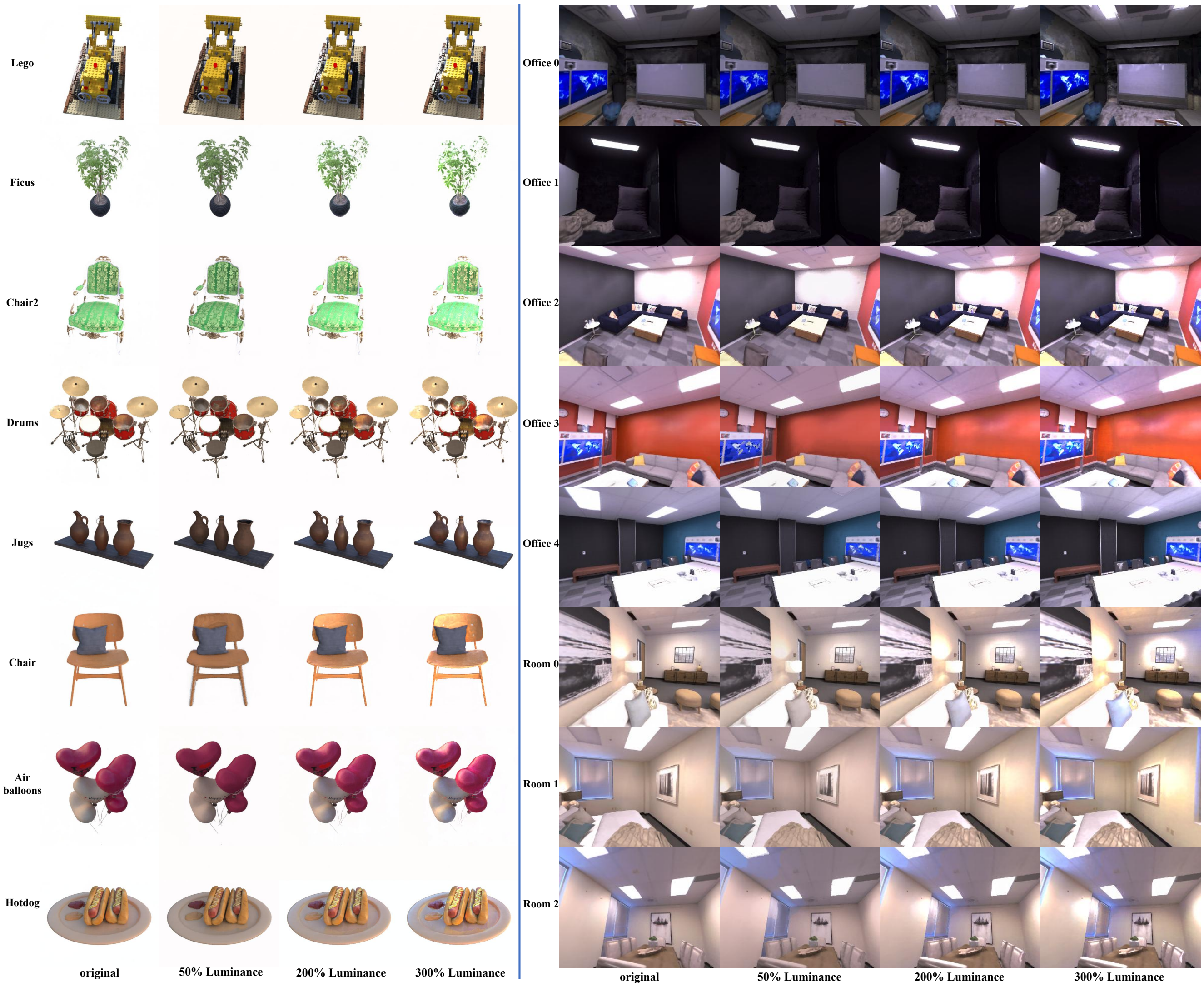


Figure B9: **Illumination Variation on Synthetic Data.** Left: Blender Object dataset, Right: Replica Scene dataset. We can adjust the brightness of the illumination, which can be applied to the ceiling, sofa, walls, and doors (such as Room 0). Please refer to the supplementary video.

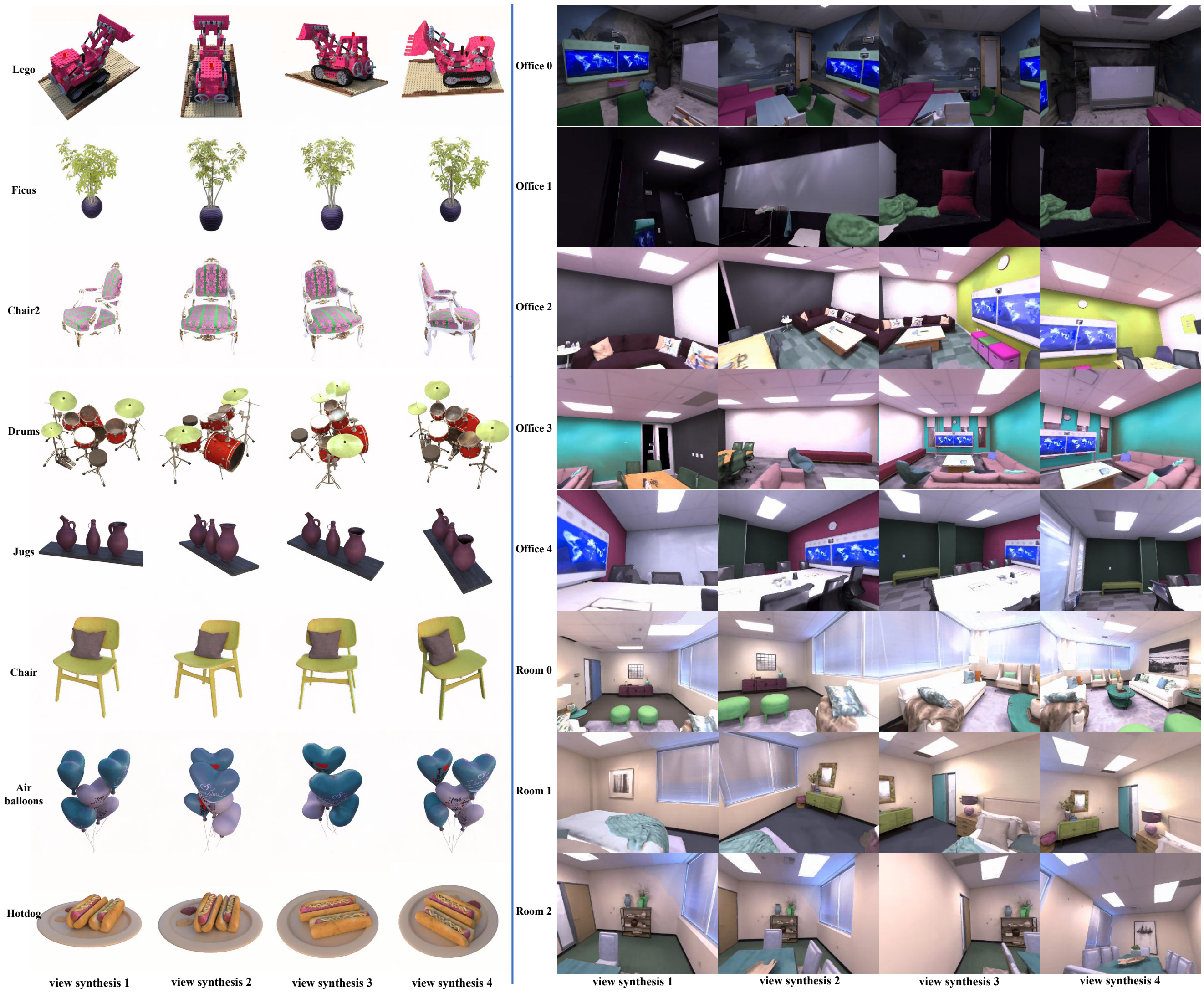


Figure B10: **Editable Novel View Synthesis on Synthetic Data.** Our method can support real-time augmented editing applications with editable novel view synthesis. Here, we show the view synthesis results with scene recoloring. For more details, please refer to the supplementary video.

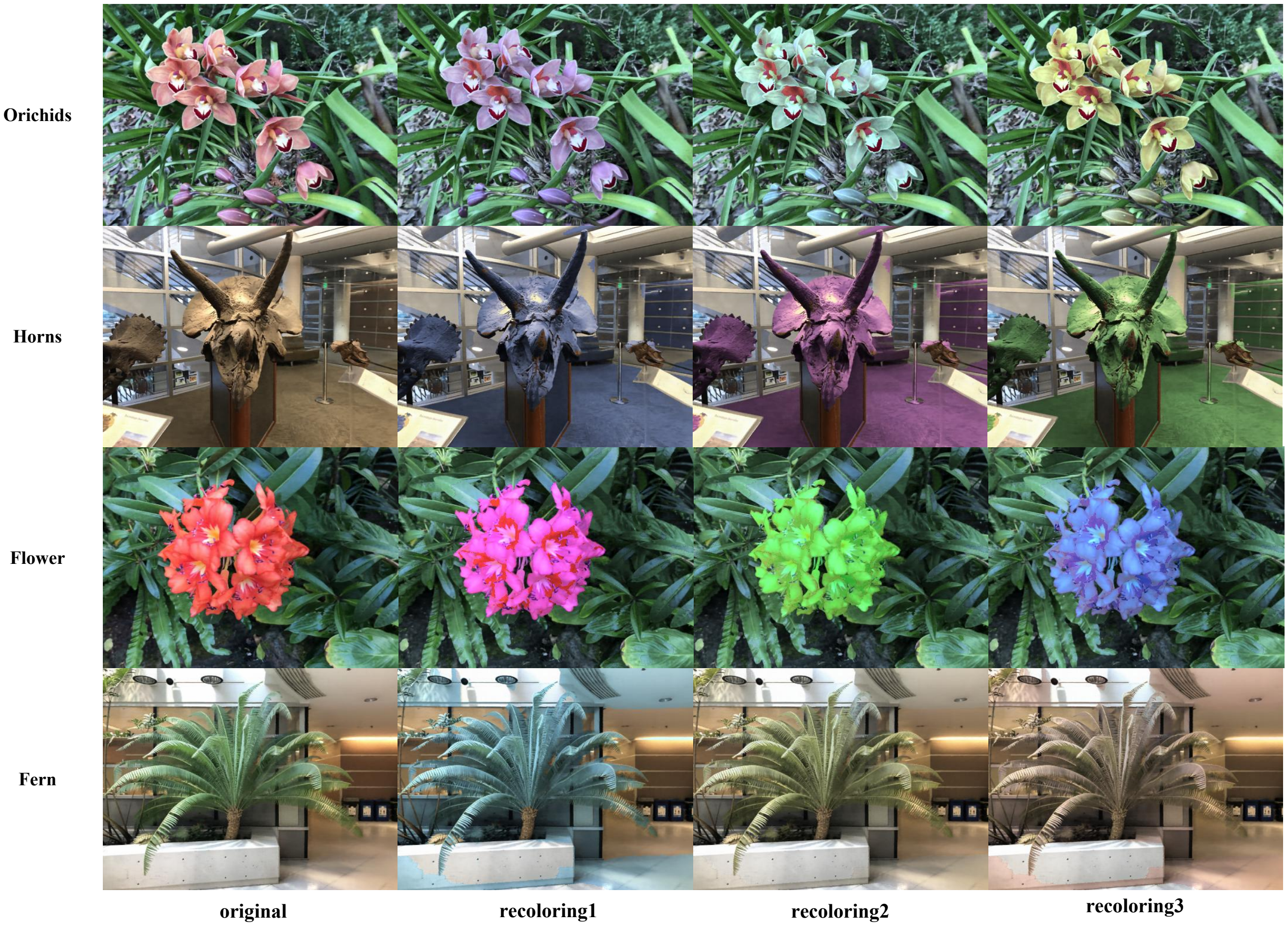


Figure B11: **Real-Time Scene Recoloring on Real-World Data.** Our approach allows for real-time region-level scene recoloring on real-world data with a simple user click and selected modified color.

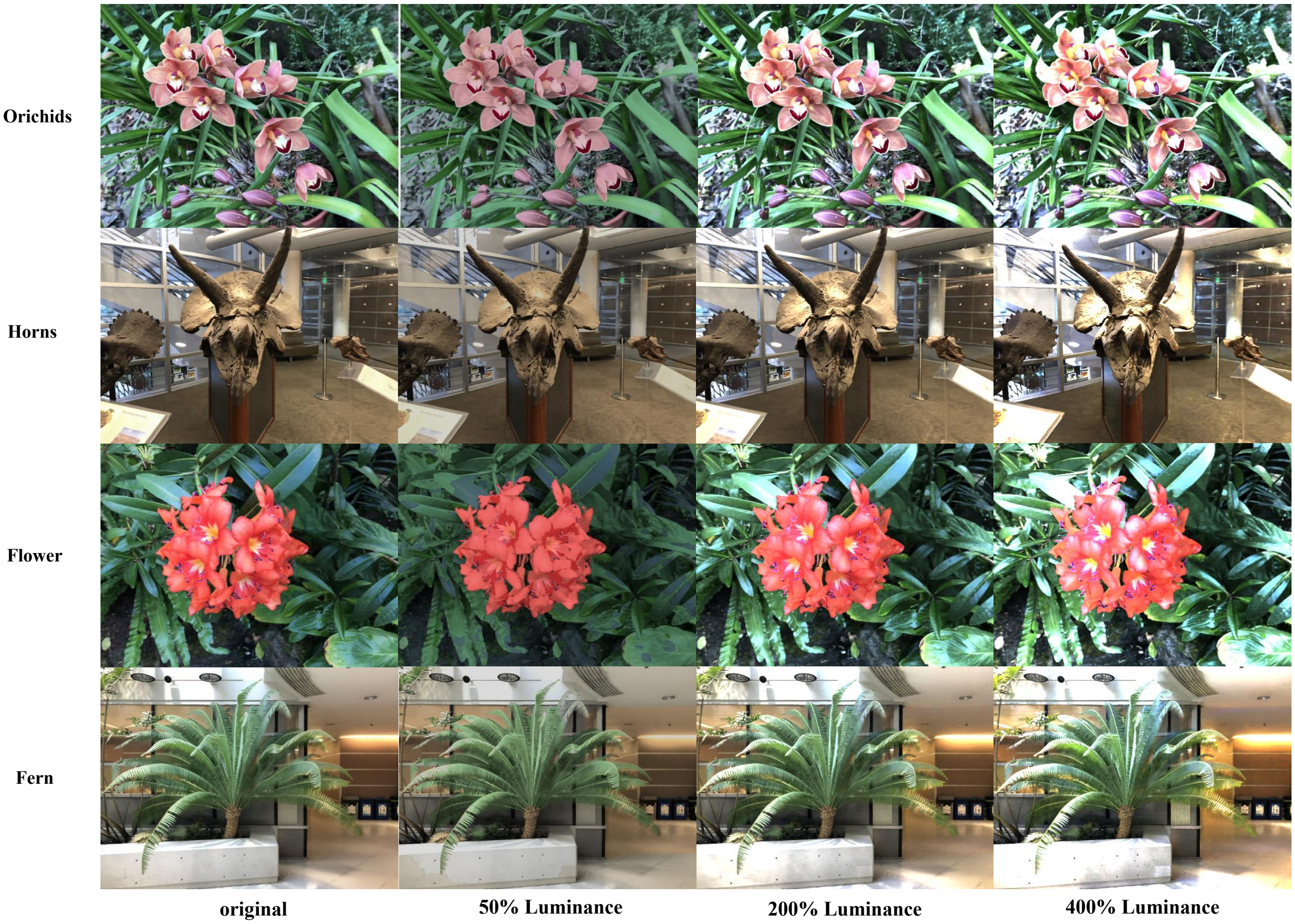


Figure B12: **Illumination Variation on Real-World Data.** We can adjust the brightness of the illumination on real-world data. Please refer to the supplementary video.

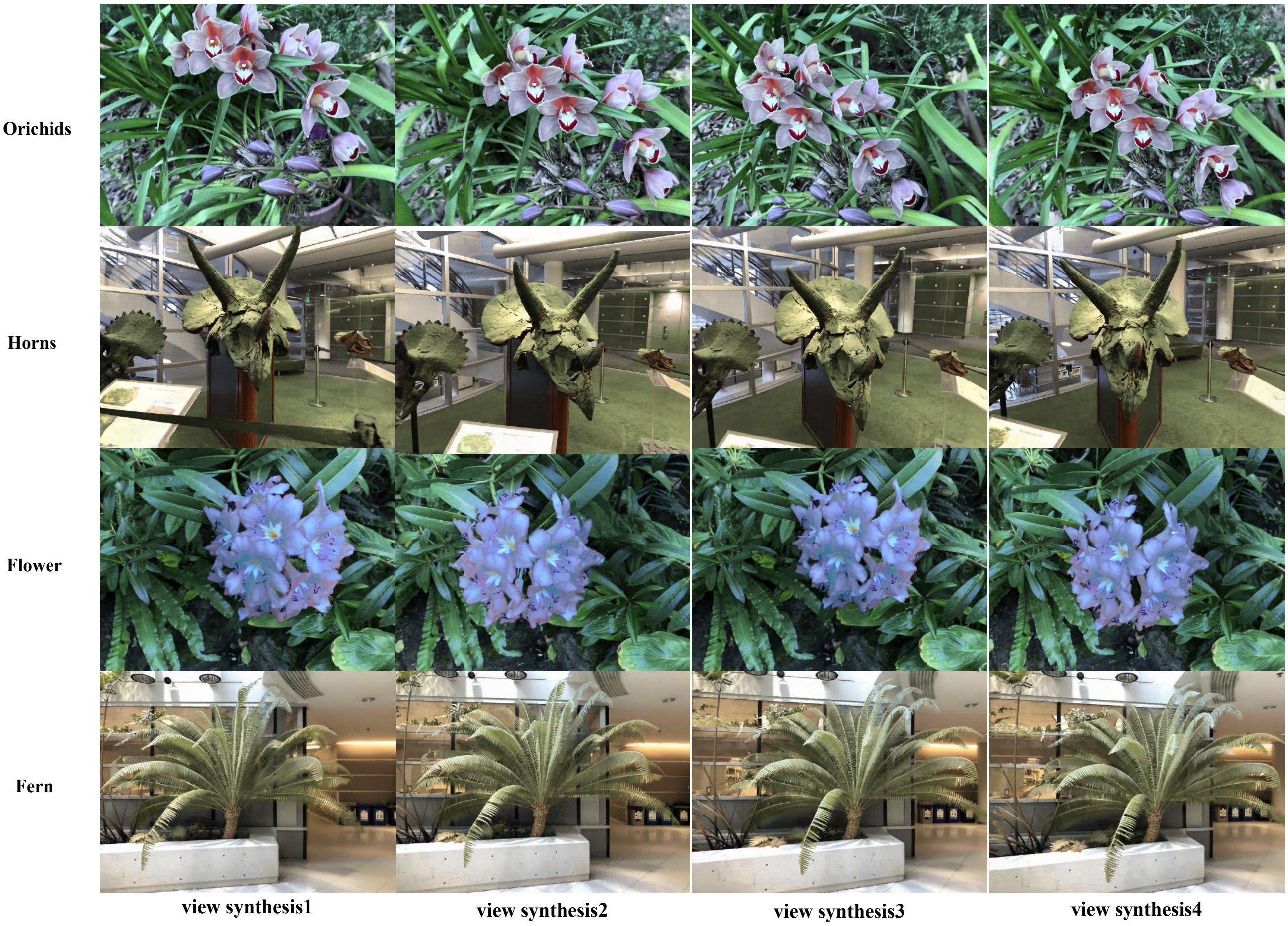


Figure B13: **Editable Novel View Synthesis on Real-World Data.** Our method can support real-time augmented editing applications with editable novel view synthesis.