

---

# KaRF: Weakly-Supervised Kolmogorov-Arnold Networks-based Radiance Fields for Local Color Editing

---

## Supplementary Material

In this supplementary document, we provide further details on our method and experimental results. We offer extended comparisons with other methods, including the IReNe method and a 3D Gaussian Splatting-based editing method, and a breakdown of computation times. We also showcase additional capabilities of KaRF, such as style transfer, and present a wider range of visual results. Finally, we discuss the limitations of our work and its potential societal impacts.

### 1 Additional Results

We present extended qualitative results on six scenes (two from each of the three datasets). For these experiments, we only provide a single coarse input mask for forward-facing scenes and three for 360° scenes. As shown in Figure 1, our method successfully generates highly fine-grained local recoloring results despite the sparse supervision.

### 2 Additional Preliminary

**NeRF.** NeRF [12] uses an MLP to learn a mapping from positional encoding  $\gamma(\mathbf{x})$  and view direction  $\gamma(\mathbf{d})$  to per-point color  $\mathbf{c}$  and density  $\sigma$ :

$$F_{\Theta}(\gamma(\mathbf{x}), \gamma(\mathbf{d})) \rightarrow (\mathbf{c}, \sigma), \quad (1)$$

where  $F_{\Theta}$  represents a learnable MLP. For each pixel, a camera ray (parameterized as  $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ , where  $\mathbf{o}$  is the camera origin and  $\mathbf{d}$  is the direction) is cast. Then, for each point  $t_i$  sampled along this ray  $\mathbf{r}$ , its color  $\mathbf{c}$  and density  $\sigma$  are integrated through volume rendering to compose the final color of that pixel, thereby generating the rendered image  $\hat{\mathbf{C}}$ :

$$\hat{\mathbf{C}} = \sum_{i=1}^N T_i (1 - \exp(-\sigma_i \delta_i)) \mathbf{c}_i, \quad \text{where } T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right), \quad (2)$$

where  $N$  denotes the number of sampling points and  $\delta_i$  is the distance between sample  $i$  and sample  $i + 1$ .

### 3 Additional Method

In our local recoloring stage, We leverage the 2D sparse weight maps  $\mathbf{W}$  as supervisory signals, enabling the generated results to effectively reduce the impact of primary color variations on non-primary colors, as illustrated in the top part of Figure 2. And Our palette  $\hat{\mathbf{P}}$  also allows for the input of randomly initialized color values, as shown in the bottom part of Figure 2.

To directly clarify the input-output relationships for each stage:

**Segmentation Stage (View-Invariant):** Inputs: 3D position + prior features. Output: consistent segmentation maps. This stage learns the intrinsic geometry and semantic class of the scene, which is independent of the viewing angle.

Table 1: Quantitative comparison with NeRF segmentation methods.

Scenes	MVSeg [13]		SA3D [2]		KaRF	
	IoU(%) $\uparrow$	Acc(%) $\uparrow$	IoU(%) $\uparrow$	Acc(%) $\uparrow$	IoU(%) $\uparrow$	Acc(%) $\uparrow$
Orchids	92.7	98.8	87.9	97.8	<b>93.2</b>	<b>98.9</b>
Leaves	94.9	99.7	<b>97.5</b>	<b>99.9</b>	97.2	99.9
Fern	94.3	99.2	97.3	99.6	<b>97.6</b>	<b>99.7</b>
Room	95.6	99.4	90.4	98.6	<b>98.3</b>	<b>99.8</b>
Horns	92.8	98.7	95.4	99.2	<b>95.4</b>	<b>99.2</b>
Fortress	97.7	99.7	<b>98.4</b>	<b>99.8</b>	98.2	99.7
Fork	87.9	99.5	<b>89.8</b>	<b>99.6</b>	84.4	99.3
Truck	85.2	95.1	96.1	98.7	<b>96.9</b>	<b>98.9</b>
Lego	74.9	99.2	90.9	99.7	<b>91.3</b>	<b>99.7</b>
Mean	90.7	98.8	93.7	99.2	<b>94.7</b>	<b>99.5</b>

Table 2: Background MSE( $\downarrow$ ) of local recoloring for each scene using KaRF, PaletteNeRF with LSeg, ICENeRF, and LAENeRF.

Scenes	PNF(with LSeg) [6]	ICENeRF [7]	LAENeRF [14]	KaRF
Flower	0.0018	0.0003	0.0017	<b>0.0002</b>
Horns	0.0136	0.0213	0.0023	<b>0.0003</b>
Fortress	0.0014	0.0010	0.0021	<b>0.0002</b>
Hotdog	0.0029	-	0.0037	<b>0.0001</b>
Lego	0.0025	-	0.0022	<b>0.0002</b>
Chair	0.0039	-	0.0031	<b>0.0005</b>
Bonsai	0.0025	-	0.0036	<b>0.0002</b>
Room	0.0028	-	0.0036	<b>0.0002</b>
Kitchen	0.0027	-	0.0026	<b>0.0003</b>

Local Recoloring Stage (View-Dependent): Inputs: 3D position + view direction + prior features. Output: consistent weight maps and a palette. Composites colors by location and viewing angle to reconstruct view-dependent effects (*e.g.*, gloss, shading, reflections).

As shown in Figure 3, our visualizations confirm that for a fixed 3D point, diffuse weights stay stable as the viewing angle changes. Conversely, light and dark weights (*e.g.*, white weight and black weight) vary significantly with view direction, which is precisely how view-dependent highlights and shadows are formed.

## 4 Additional Experimental Details

### 4.1 Segmentation Details

In the qualitative segmentation evaluation, our method outperforms both SA3D [2] and LAENeRF [14] in terms of both clarity and detail of the segmentation masks, while maintaining excellent multi-view consistency. Furthermore, following the SPIn-NeRF [13] benchmark protocol, in Figure 4, we provide additional examples of segmentation mask annotations from KaRF alongside the resulting generated masks. It can be observed that our method accurately learns the geometric information of the scene under guidance from the reference views. As shown in Table 1, in the quantitative segmentation evaluation, our method achieved the best results for both Intersection over Union (IoU) and Accuracy (Acc) across multiple scenes. Furthermore, it demonstrated optimal average performance, showcasing its superior segmentation capabilities and a profound understanding of scene geometry. The performance deviation in the fork scene is attributed to the treatment of the fork head as a single, unified region in the ground truth.

### 4.2 Local Recoloring Details

In the local recoloring stage, KaRF merely requires space-level palette base colors and weights as input, which are inconsistent and imprecise. It then generates palettes and weights consistent with the color distribution of the scene and exhibits multi-view consistency, as shown in Figure 5, fully demonstrating the advantages of our method under weak supervision.

Table 3: Per-scene LPIPS and RMSE for local recoloring using KaRF, PaletteNeRF with LSeg, and LAENeRF.

Consistency	Dataset	Scenes	PNF(with LSeg) [6]		LAENeRF [14]		KaRF	
			LPIPS↓	RMSE↓	LPIPS↓	RMSE↓	LPIPS↓	RMSE↓
Short-range	LLFF [11]	Flower	0.088	0.034	0.083	0.024	<b>0.076</b>	<b>0.021</b>
		Horns	0.147	0.077	0.141	0.070	<b>0.131</b>	<b>0.066</b>
		Fortress	0.110	0.087	0.106	0.089	<b>0.093</b>	<b>0.087</b>
		Hotdog	<b>0.246</b>	0.515	0.250	0.516	0.247	<b>0.513</b>
	Blender [12]	Lego	0.237	0.113	0.239	0.115	<b>0.228</b>	<b>0.106</b>
		Chair	0.135	0.089	0.138	0.083	<b>0.128</b>	<b>0.082</b>
		Bonsai	0.235	0.097	0.243	0.097	<b>0.225</b>	<b>0.086</b>
	Mip360 [1]	Room	0.251	0.119	0.249	<b>0.100</b>	<b>0.236</b>	0.102
		Kitchen	0.198	0.093	<b>0.192</b>	0.088	0.193	<b>0.082</b>
	Mean		0.183	0.136	0.182	0.131	<b>0.173</b>	<b>0.127</b>
Long-range	LLFF [11]	Flower	0.187	0.079	0.184	0.054	<b>0.181</b>	<b>0.049</b>
		Horns	0.312	0.203	0.307	0.185	<b>0.307</b>	<b>0.182</b>
		Fortress	0.206	<b>0.201</b>	0.208	0.202	<b>0.193</b>	0.204
		Hotdog	0.378	0.652	0.385	0.655	<b>0.377</b>	<b>0.648</b>
	Blender [12]	Lego	0.381	0.236	0.382	0.237	<b>0.377</b>	<b>0.222</b>
		Chair	0.300	<b>0.186</b>	0.301	0.248	<b>0.296</b>	0.213
		Bonsai	0.517	0.239	0.543	0.237	<b>0.490</b>	<b>0.225</b>
	Mip360 [1]	Room	0.578	0.296	0.587	0.271	<b>0.568</b>	<b>0.270</b>
		Kitchen	0.492	0.263	0.493	0.252	<b>0.466</b>	<b>0.234</b>
	Mean		0.372	0.262	0.377	0.260	<b>0.362</b>	<b>0.250</b>

Table 4: The PSNR and SSIM for local recoloring using KaRF, PaletteNeRF with LSeg, and LAENeRF.

Dataset	PNF(with LSeg) [6]		LAENeRF [14]		KaRF	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
LLFF [11]	36.26	0.978	30.25	0.961	33.52	0.966
Blender [12]	35.38	0.987	33.23	0.978	38.95	0.988
Mip360 [1]	35.04	0.986	32.53	0.980	31.63	0.952
Mean	35.68	0.984	32.00	0.973	34.70	0.969

We conduct an additional comparison with IReNe [10], as illustrated in the Figure 6. It can be observed that IReNe exhibits color bleeding outside the segmented regions. Moreover, this method cannot correctly represent the colors of specular reflections, as seen on the seat cushion of the chair scene. It is noteworthy that the results for IReNe were obtained from its publication [10] and official website.

Regarding the metrics reported in quantitative evaluation, Figure 7 presents nine example foreground masks drawn from different datasets. Subsequently, we compare the MSE results before and after local recoloring across various scenes for PaletteNeRF [6] equipped with the LSeg module [8], ICENeRF [7], and LAENeRF [14], as shown in Table 2. Evidently, the values of our method are one order of magnitude lower than those of other methods across all scenes.

To evaluate multi-view consistency, we select pairs of views with intervals of 1 and 7 under short-range and long-range conditions, respectively. Table 3 presents the LPIPS [16] and RMSE results across various scenes after local recoloring with KaRF, PaletteNeRF [6] equipped with the LSeg module [8], and LAENeRF [14]. As can be seen, KaRF achieves the best performance in terms of consistency.

Furthermore, to demonstrate the reconstruction quality of palette-based local editing, we utilize PSNR and SSIM to compare the quality of local regions synthesized by our method with PaletteNeRF and LAENeRF after palette synthesis (where the palette has not undergone any color editing). Specifically, we select 10 views from one scene in each of the three datasets. Using the foreground masks generated during the segmentation stage, we segment the images composited by each method to obtain composited images containing only the local foreground. Subsequently, we calculate metrics against the input images to highlight the reconstruction quality of the segmented regions after color layer decomposition and composition, as shown in the Table 4. The related works ICENeRF and

Table 5: The impact of  $\mathcal{L}_{\text{Palette}}$ .

W/o $\mathcal{L}_{\text{Palette}}$		W/ $\mathcal{L}_{\text{Palette}}$	
PSNR $\uparrow$	SSIM $\uparrow$	PSNR $\uparrow$	SSIM $\uparrow$
28.95	0.940	<b>31.72</b>	<b>0.958</b>

Table 6: The impact of GRBFKAN.

MLP		B-Spline KAN		GRBF KAN	
PSNR $\uparrow$	SSIM $\uparrow$	PSNR $\uparrow$	SSIM $\uparrow$	PSNR $\uparrow$	SSIM $\uparrow$
27.65	0.934	30.29	0.945	<b>31.63</b>	<b>0.952</b>

IReNe are not open-sourced, and neither are palette-based methods, thus they do not have a palette reconstruction process.

### 4.3 Ablation Study Details

**Impact of  $\gamma(d)$ .** We present an ablation study on adding view dependence in Figure 8. By amplifying the image contrast 4x to highlight specular effects, the results show that view-dependent weights achieve a significant improvement in capturing complex lighting when compared to a view-invariant model.

**Impact of  $\mathcal{L}_{\text{Palette}}$ .** It can be observed in Table 5 (We select Lego scene to evaluate) that the model without the palette-adaptive reconstruction loss (w/o  $\mathcal{L}_{\text{Palette}}$ ) exhibits a noticeable deviation in the color reproduction of local regions when compared to the original scene.

**Details.** In our GRBFKAN ablation study, we replace the GRBFKAN layer in our residual adaptive gating KAN network structure with B-spline KAN and Linear layer for training. And the composition quality of the palettes and weights extracted by the three aforementioned methods is shown in the Table 6. We select Kitchen scene to evaluate. Therefore, the palettes extracted by GRBF KAN are of higher quality and greater precision.

Furthermore, in the top part of Figure 9, we present the average training process of one scene selected from each of the three datasets, and the results indicate that GRBFKAN exhibits lower loss and faster convergence speed. On the other hand, the overall structure with residual connections demonstrates significant improvement in terms of convergence speed and loss, as shown in the bottom part of Figure 9.

As part of the ablation study for our proposed Residual Adaptive Gating KAN structure, we derive explicit formulaic representations under four configurations: stacking only KAN layers (w/o All), introducing gating KAN alone (w/ Gate), introducing gating and the adaptive operator  $\mathcal{G}$  without residual connections (w/o Res), and including all components (w/ All).

- w/o All:

$$\mathbf{f}'_h = \mathbf{f}_h. \quad (3)$$

- w/ Gate:

$$\mathbf{f}'_h = \text{SiLU}(\mathbf{f}_g) \cdot \mathbf{f}_h. \quad (4)$$

- w/o Res:

$$\mathbf{f}'_h = \text{SiLU}(\beta \mathbf{f}_g) \cdot \mathbf{f}_h. \quad (5)$$

- w/ All:

$$\mathbf{f}'_h = \text{Sigmoid}(\alpha) \cdot \mathbf{f}_q + \text{SiLU}(\beta \mathbf{f}_g) \cdot \mathbf{f}_h, \quad (6)$$

## 5 Comparison with 3D Gaussian Splatting-based editing method

As shown in Figure 10, we conduct a comparative evaluation against the latest conditional editing method [4], which employs 3D Gaussian splatting. Since [4] is not open-sourced, we derive its results directly from the original publication. It can be observed that our method demonstrates superior performance and delivers more precise effects in local color edits.

## 6 Time Comparisons

We use the Fern scene from the LLFF dataset [11] to provide an exemplary comparison of processing times. When using a pre-trained radiance field, PaletteNeRF [6] requires approximately 12 minutes

for recoloring a selected object, LAENeRF [14] completes the task in around 5.5 minutes, and ICE-G [4] takes about 21 minutes according to their original publication. In our method, the pre-trained and frozen NeRF [12] is only used to provide prior knowledge and does not participate in the subsequent local color editing task. To more effectively leverage this prior knowledge, we adopt KAN [9] as our foundational architecture. The KAN network significantly enhances the nonlinear representational capacity of the model through the integration of multiple nonlinear activation functions, thereby delivering superior performance in complex scenarios. However, this improvement in expressiveness inevitably introduces additional computational overhead, resulting in somewhat increased processing times. In practical applications, our proposed method typically completes local color editing tasks within approximately 10 minutes.

While our method takes longer than LAENeRF, its processing time is favorable when compared to other SOTA editing methods. KaRF is faster than both PaletteNeRF and the recent 3DGS-based method ICE-G. This indicates that the processing time of KaRF is within a reasonable range, offering a practical balance between speed and quality. And this suggests good potential for deployment in semi-interactive scenarios.

## 7 User Study Details

Among our 44 participants (21 female, 23 male), 39 possessed a foundational understanding of deep learning and were familiar with the local color editing task. Additionally, 12 participants had knowledge of NeRF.

## 8 Style Transfer

To achieve artistic stylization effects, we design an overall pipeline as shown in Figure 11. A 2D stylization method [5] is first used to generate an initial stylized image. Then, leveraging the consistent masks obtained in the segmentation stage, we refine the results to generate localized stylization. These localized results are subsequently passed into our local recoloring module, where layer decomposition and weight training are performed to generate palettes and weights that matches the target style while maintaining multi-view consistency. Finally, based on these weights, we can precisely achieve the desired local stylized recoloring effects. We evaluate the style transfer performance on the Blender dataset [12], as illustrated in Figure 12.

## 9 More Visual Results

We present additional visual results of KaRF in Figure 13 to validate the effectiveness of our approach.

## 10 Limitations

Although KaRF can accurately perform local color editing of NeRF, some challenges remain. Similar to [14, 1], NeRF often sacrifice geometric fidelity by utilizing samples behind the surface to enhance the visual realism of non-Lambertian effects. This characteristic poses a challenge for our work on recoloring radiance fields, particularly when dealing with visually detailed, real-world unbounded scenes, as it can lead to a decrease in the quality of the recolored results. Another limitation of KaRF lies in its scalability: despite the expressive power of KAN [9] in capturing nonlinear features, its learnable node structure introduces extra training cost, which may affect performance in large-scale or real-time scenarios.

## 11 Societal Impacts

Local color editing of NeRF offers powerful tools for creative expression and design, enabling artists and professionals to easily modify 3D scenes for entertainment, architecture, or commerce. However, the ability to realistically alter scenes also raises significant concerns about the potential for creating convincing misinformation and deepfakes, which could be used maliciously.

## References

- [1] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *CVPR*, 2022.

- [2] Jiazhong Cen, Zanwei Zhou, Jiemin Fang, chen yang, Wei Shen, Lingxi Xie, Dongsheng Jiang, XiaoPeng Zhang, and Qi Tian. Segment anything in 3d with nerFs. In *NeurIPS*, 2023.
- [3] Cheng-Kang Ted Chao, Jason Klein, Jianchao Tan, Jose Echevarria, and Yotam Gingold. Locopalettes: Local control for palette-based image editing. *Computer Graphics Forum*, 42(4):e14892, 2023.
- [4] Vishnu Jaganathan, Hannah Hanyun Huang, Muhammad Zubair Irshad, Varun Jampani, Amit Raj, and Zsolt Kira. Ice-g: Image conditional editing of 3d gaussian splats, 2024.
- [5] Nicholas Kolkin, Michal Kucera, Sylvain Paris, Daniel Sykora, Eli Shechtman, and Greg Shakhnarovich. Neural neighbor style transfer, 2022.
- [6] Zhengfei Kuang, Fujun Luan, Sai Bi, Zhixin Shu, Gordon Wetzstein, and Kalyan Sunkavalli. Palettenerf: Palette-based appearance editing of neural radiance fields. In *CVPR*, 2022.
- [7] Jae-Hyeok Lee and Dae-Shik Kim. Ice-nerf: Interactive color editing of nerfs via decomposition-aware weight optimization. In *ICCV*, 2023.
- [8] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *ICLR*, 2022.
- [9] Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljacic, Thomas Y. Hou, and Max Tegmark. KAN: Kolmogorov–arnold networks. In *ICLR*, 2025.
- [10] Alessio Mazzucchelli, Adrian Garcia-Garcia, Elena Garces, Fernando Rivas-Manzanique, Francesc Moreno-Noguer, and Adrian Penate-Sanchez. Irene: Instant recoloring of neural radiance fields. In *CVPR*, 2024.
- [11] 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 TOG*, 2019.
- [12] 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 *ECCV*, 2020.
- [13] Ashkan Mirzaei, Tristan Aumentado-Armstrong, Konstantinos G. Derpanis, Jonathan Kelly, Marcus A. Brubaker, Igor Gilitschenski, and Alex Levinshtein. Spin-nerf: Multiview segmentation and perceptual inpainting with neural radiance fields. In *CVPR*, 2022.
- [14] Lukas Radl, Michael Steiner, Andreas Kurz, and Markus Steinberger. Laenerf: Local appearance editing for neural radiance fields. In *CVPR*, 2024.
- [15] Jianchao Tan, Jose Echevarria, and Yotam Gingold. Efficient palette-based decomposition and recoloring of images via rgbxy-space geometry. *ACM TOG*, 37(6), 2018.
- [16] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

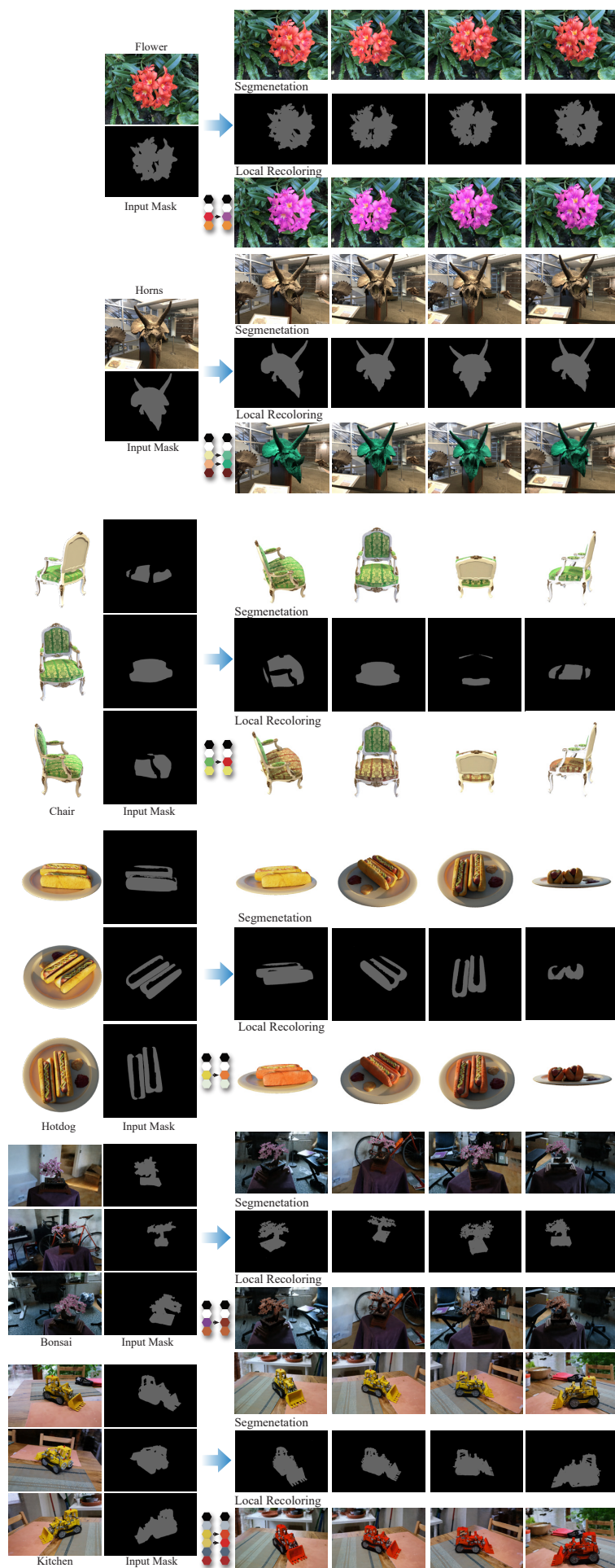


Figure 1: Additional results of KaRF.

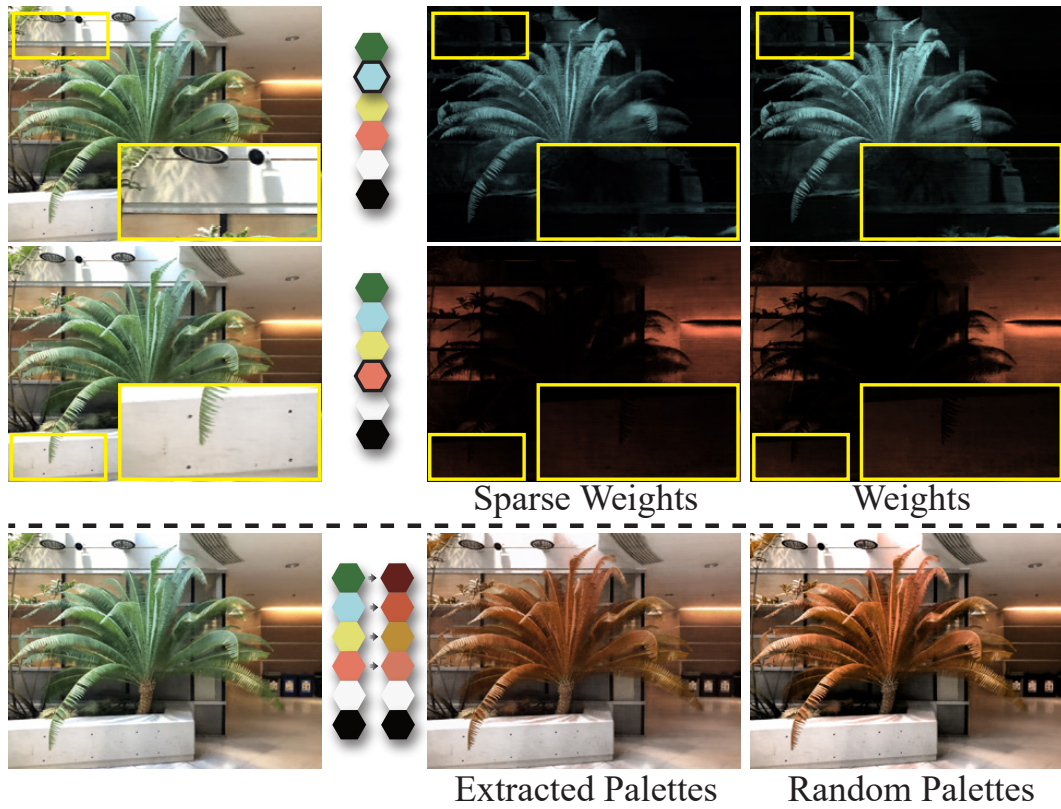


Figure 2: Above: Comparison between the sparse weights we generated based on [3] and the weights we generated based on [15]. Below: Comparison between the recoloring results using extracted palettes and those using random palettes.

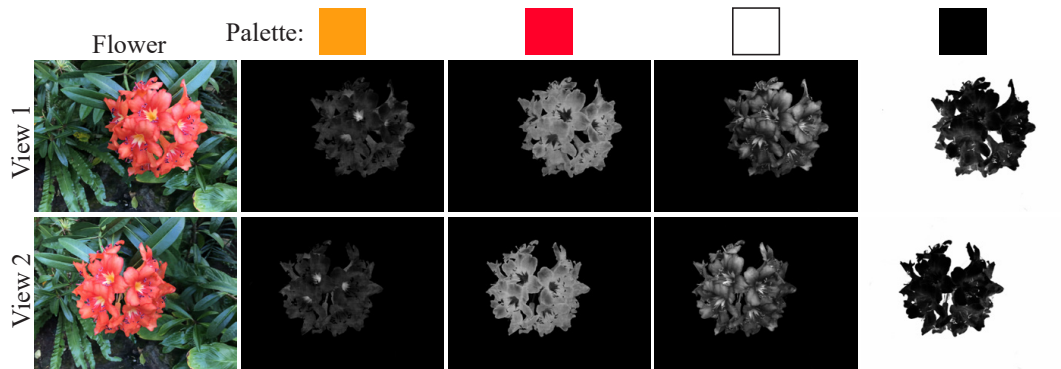


Figure 3: White weight and black weight vary significantly with view direction in the flower scene.

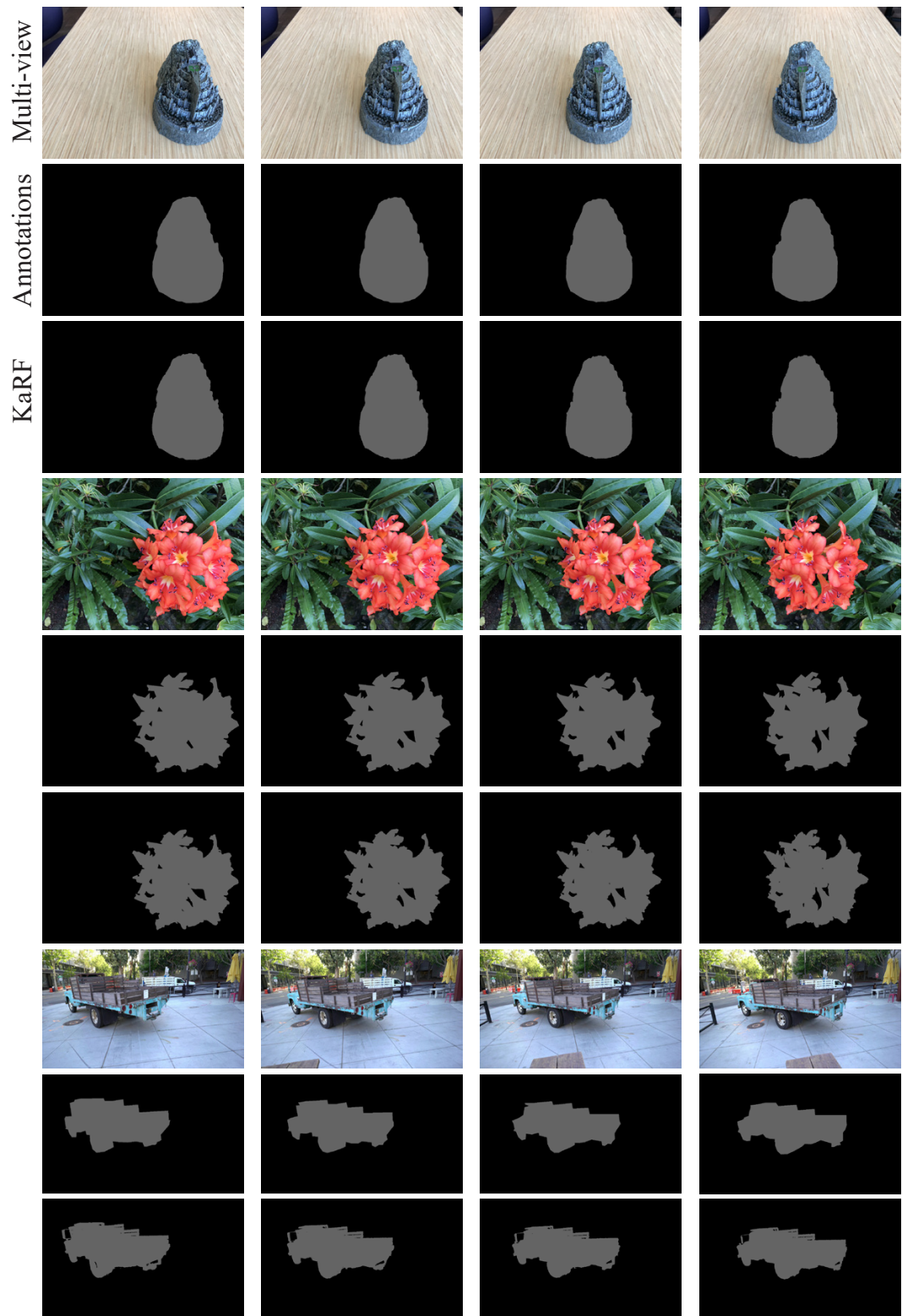


Figure 4: Additional spatial-level annotations generated by SAM [2] and corresponding results generated by KaRF.

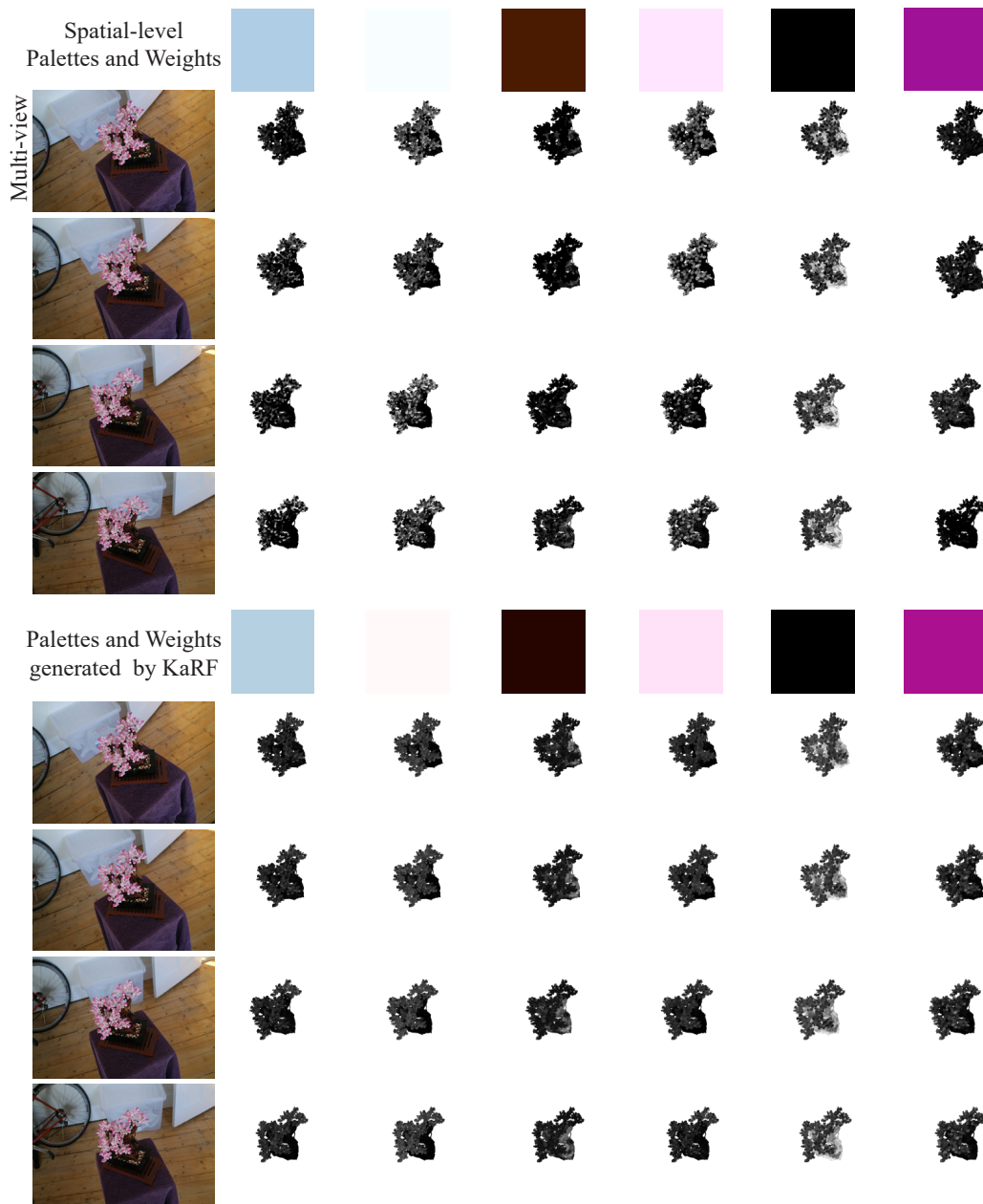


Figure 5: Additional spatial-level palettes and weights generated by LocoPalette [3] and corresponding results generated by KaRF.

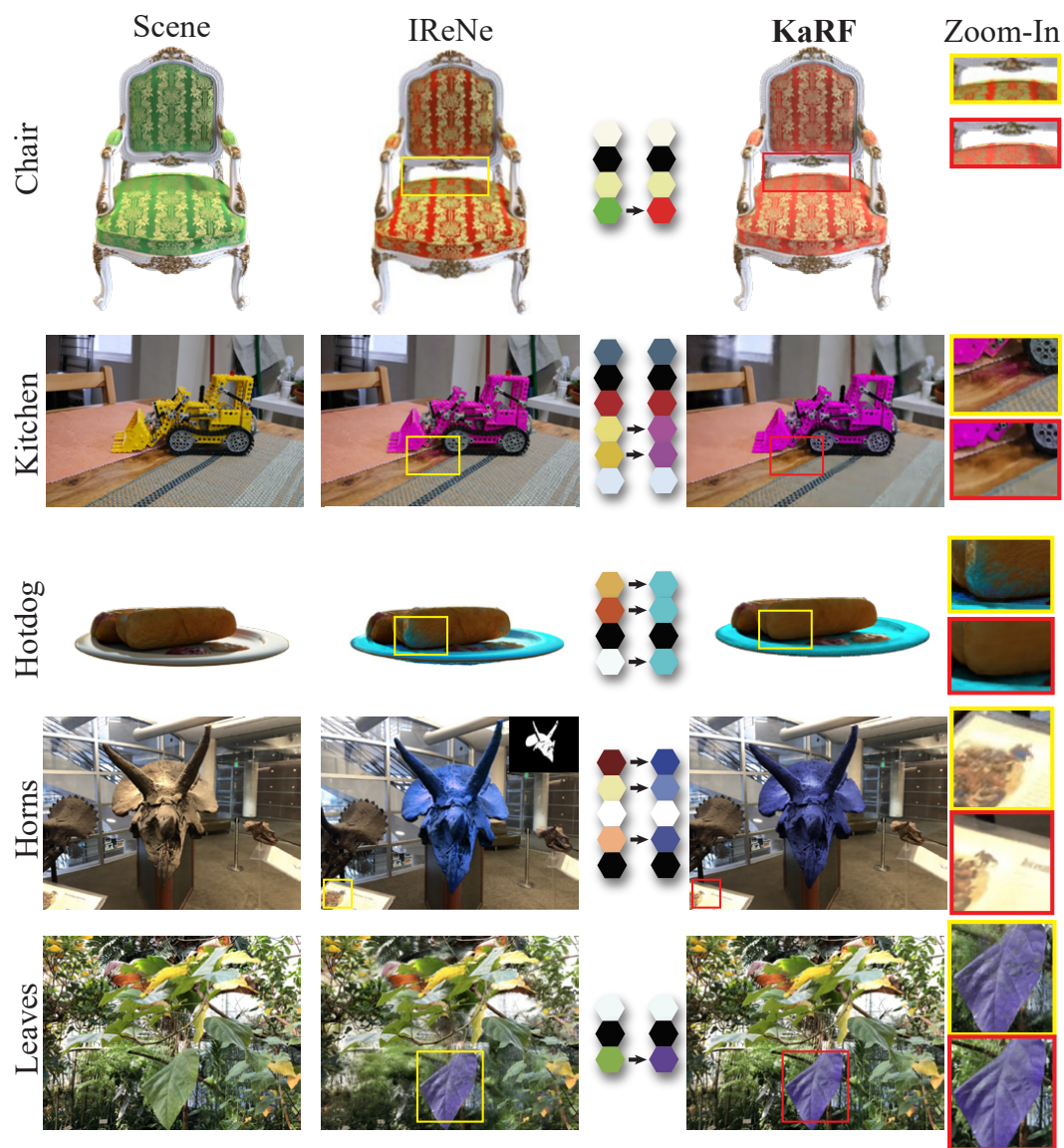


Figure 6: Qualitative comparison with IReNe [10]. Zoom-in views are highlighted.

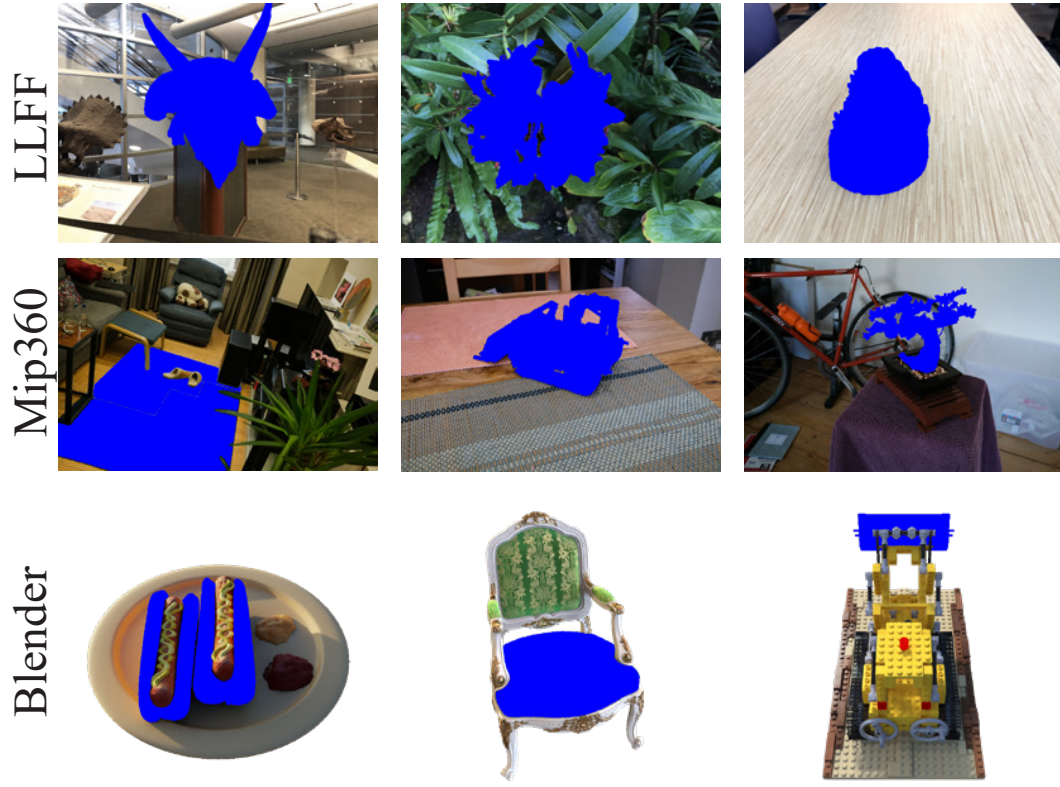


Figure 7: Foreground masks in local recoloring experiments.

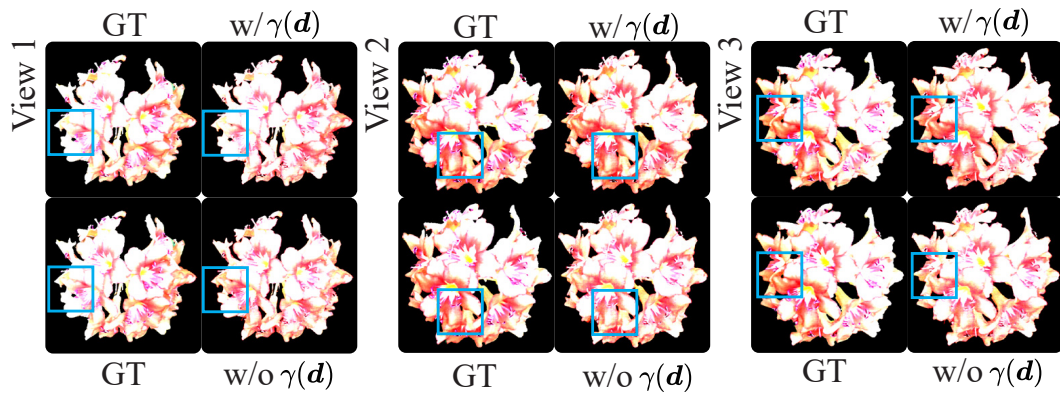


Figure 8: Impact of  $\gamma(d)$ . The blue boxes highlight the differences.

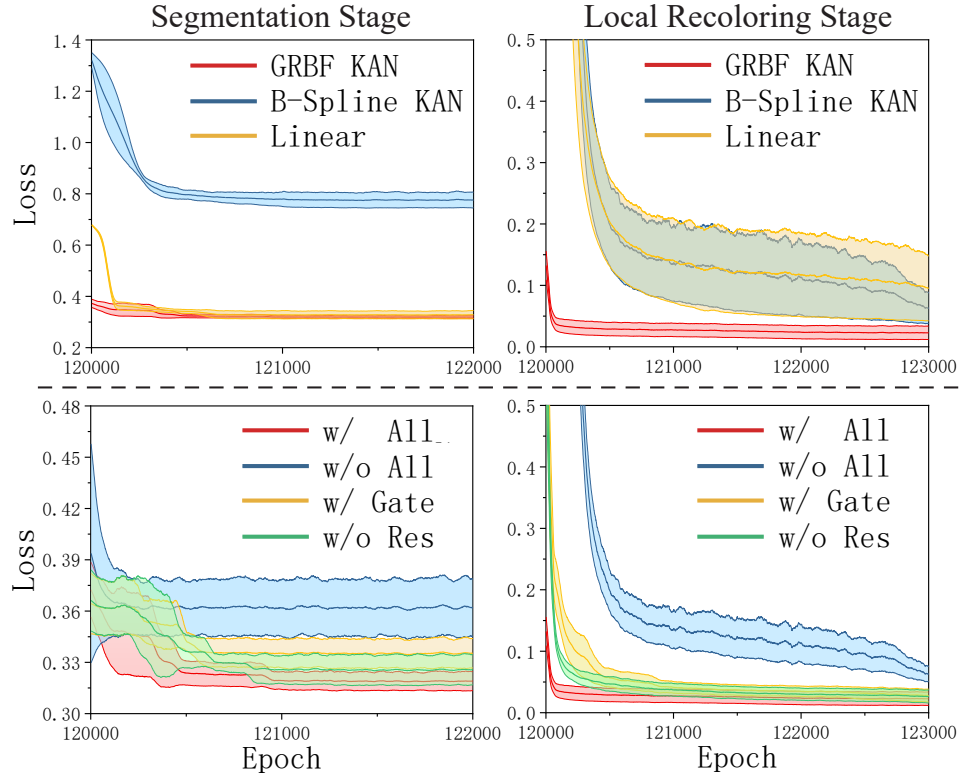


Figure 9: The impact of GRBFKAN and residual adaptive gating KAN on loss during the segmentation and local recoloring stages.

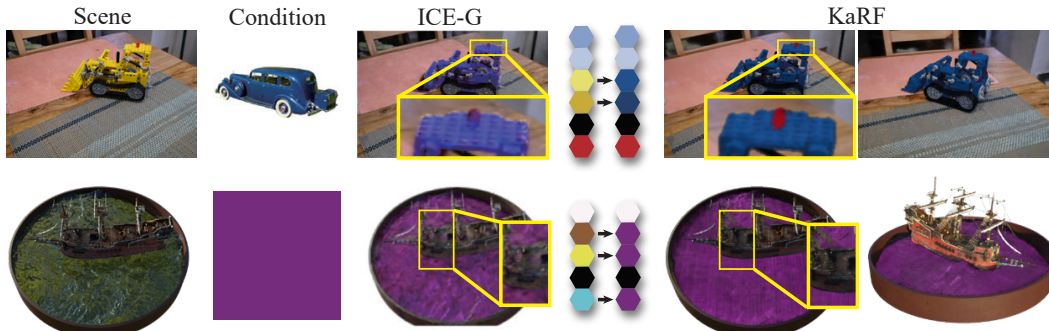


Figure 10: Comparison with 3D Gaussian Splatting-based editing method. Zoom-in views are highlighted within the yellow boxes.

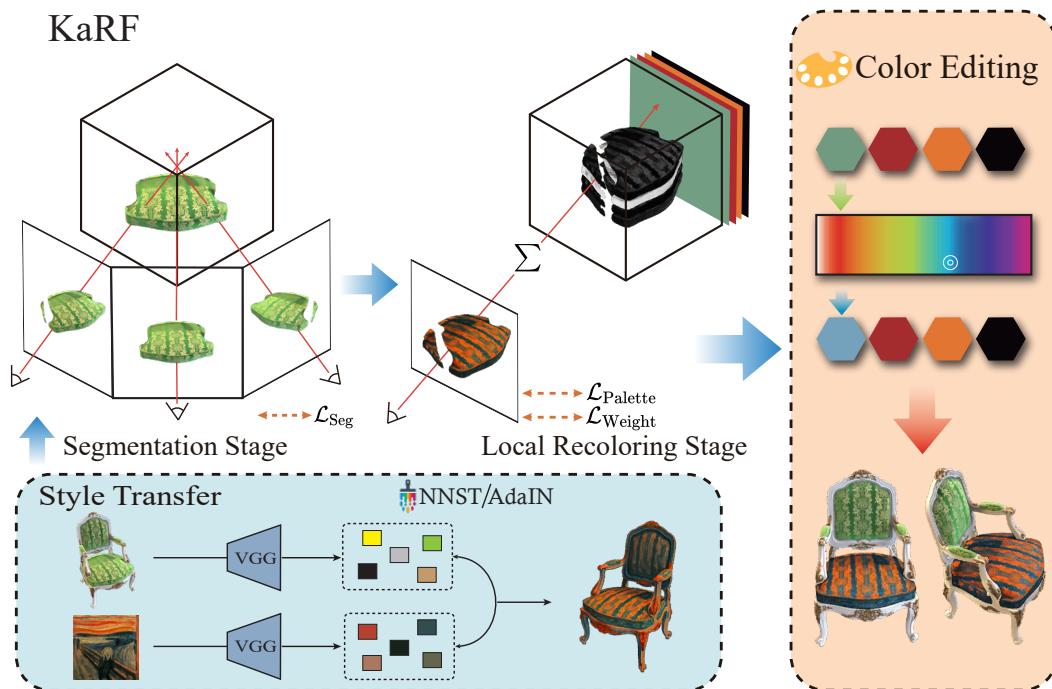


Figure 11: An overall style transfer pipeline of KaRF.

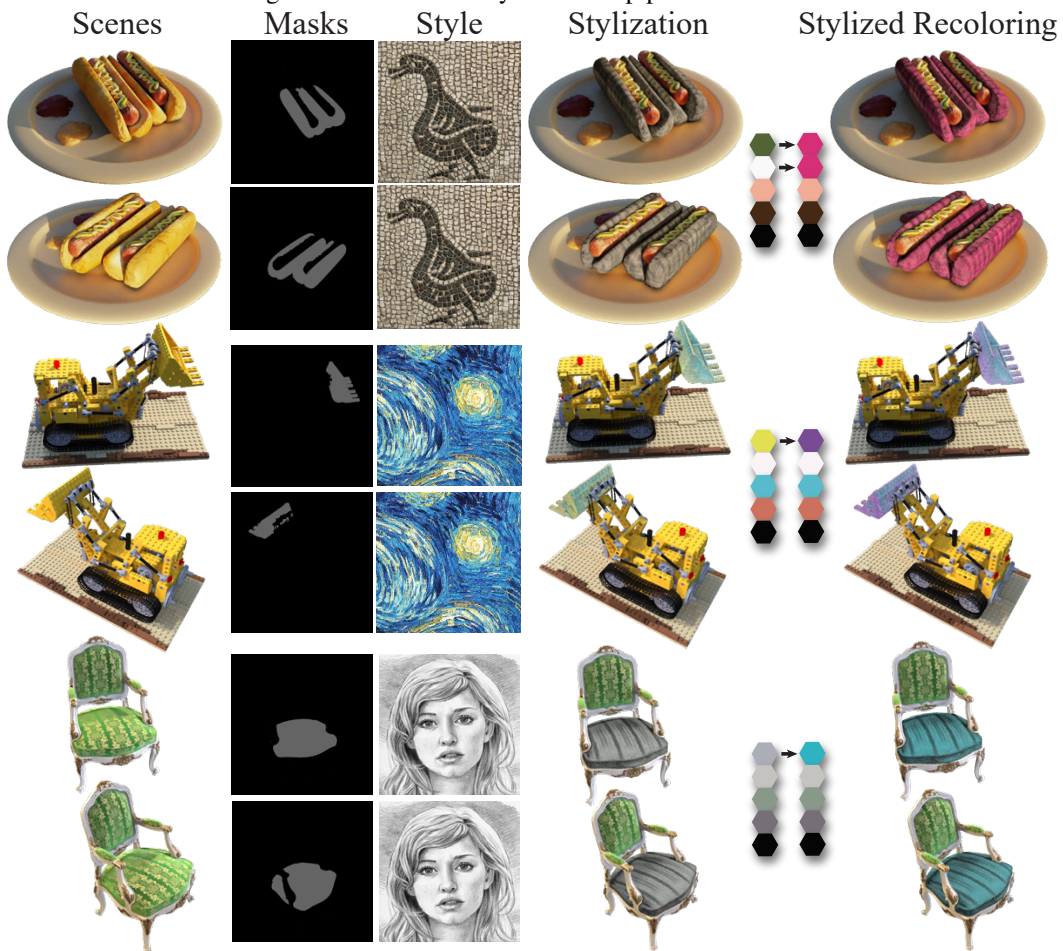


Figure 12: Style transfer results of KaRF.

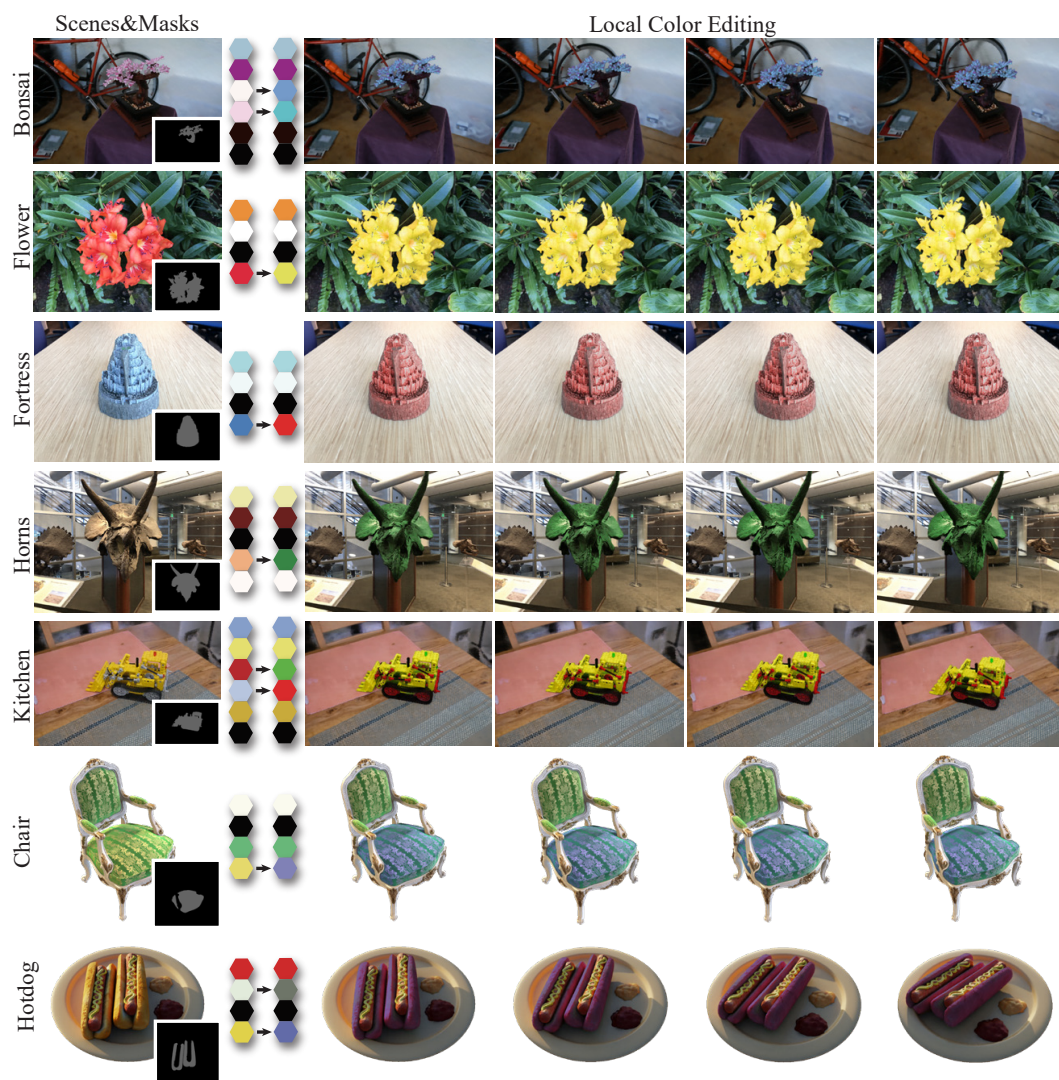


Figure 13: Additional visual results of local color editing.