Interactive Segmentation of Radiance Fields

1. Implementation Details

All the methods proposed in the paper have been implemented using PyTorch [8] branching off the code provided by DVGOv2 [11]. All experiments are performed using a commodity hardware equipped with AMD Ryzen 5800x and a NVIDIA RTX 3090.

The feature components of the radiance fields namely radiance latent vectors and the learnt DINO features are stored using VM decomposition proposed by TensoRF [3]. For radiance latent vectors, we use *VM-48* representation of TensoRF and for DINO features, we use *VM-64* variant of TensoRF. The segmentation masks and densities have been stored as a full voxel grids.

The DINO *ViT-b8* [2] model provides 768 features for each patch of 8×8 pixels in an image. We reduce the dimensionality of these features by doing a principal component analysis reducing the effective dimension to 64. This is consitent with the prior works [6, 12]. For each pixel, the feature is calculated by referring to the feature of the respective patch that pixel corresponds to.

We first pre-train the model for the volumetric density and radiance for $20,000$ iterations. Once the radiance field is stabilized on the VM-48 TensoRF representation, we introduce distillation using *student-teacher* strategy similar to that of [6, 12] on the VM-64 TensoRF variant. Upon adoption, the resultant VM-48 variant of TensoRF along with its shallow MLP represents the radiance field, and VM-64 consititute the distilled features. It is to be noted that the distilled features are not accompanied by a shallow MLP. The features are store at voxel lattice locations and tri-linearly interpolated to be compared and optimized against the DINO features without the involvement of any non-linearity. The adoption is done with $\lambda = 0.001$ for the weighted loss function for $5,000$ iterations. The loss is taken on the features and radiance together to maintain consistency.

We choose $K = 10$ when applying K-Means to the set of features selected from the user's brush stroke. For the bilateral search, the value of σ_ϕ and σ_s are set to 10.0 and the 1.0 respectively while the threshold value τ is 0.1.

2. Scene Editing

In this section, we explain the procedures that were followed for editing the 3D scenes post segmentation. The segmentation procedure provides a 3D bit map representing the segmented voxels. Utilization of an additional bitmap also assists in faster rendering as the voxels with segmentation mask values of 0 can easily be filtered out. Fig. 1 shows the additional results of scene editing.

2.1. Object Removal

For removing a segmented object from the scene, we alter the evaluation of the density for a 3D point. We simultaneously evaluate the bit map value b_x at the queried point. To segmented the object of interest (foreground), the effective density σ'_x is $\sigma_x * b_x$. Similarly, to render the background the effective density σ'_x is $\sigma_x * (1.0 - b_x)$.

2.2. Translation

If an object needs to be moved to another location, the ray queries lying inside the object's voxel space can be shifted to the desired location. Let t be the translation vector for the object to be moved, then the object's ray-point query changes as shown below.

$$\sigma'_x, rgb'_x = \sigma_x, rgb_x \; \forall \; b_x = 0$$
$$\sigma'_x, rgb'_x = \sigma_{x+t}, rgb_{x+t} \; \forall \; b_x = 1$$

2.3. Scene Composition

To perform scene composition, we follow a similar strategy used by D^{2}NeRF [13]. We alter the volumetric rendering equation to account for density and color from both the scenes as shown below:

$$\hat{C}(r) = \int_{t_n}^{t_f} T(t) \left(\sigma_1(t)c_1(t) + \sigma_2(t)c_2(t)\right) dt$$
$$T(t) = exp\left(-\int_{t_n}^{t} (\sigma_1(s) + \sigma_2(s))\right) ds$$

The results for scene composition have been shown in the main paper and Fig. 1 of the supplementary.

(a) Original Rendered Image (b) Removal of Pot (c) Composition (d) Style Transfer

Figure 1. *Seamless Progressive Scene Editing*: Image (a) is the reference rendered viewpoint. In (b), the pot has been removed. Image (c) shows scene composition. The JCB from KITCHEN scene has been placed on the top of the table in the GARDEN scene. Image (d) shows appearance editing of specific objects. We apply style transfer on just the JCB. For more details please refer to Sec. 2.

(a) Rendered View 1 (b) Rendered View 2

(c) Segmented Trex (d) Depth Map

Figure 2. *Finer Segmentation*: Images (a) and (b) show rendered views of T-Rex from the LLFF dataset [7]. Image (c) shows the segmented output of T-Rex scene. Our method achieves fine-grained segmentation of objects such as the rib-cage bones of T-Rex. However, on close observation, the region near the tail bones background bleeds in. This is due to the wall and the tail-bone lie at the similar depth as shown in the depth map (d). This can be mitigated by having more 3D information (better training views) or higher voxel grid resolution.

2.4. Appearance Editing

Here, we apply style transfer on an already composed scene. We first calculate a 3D bitmap for the JCB lego in the KITCHEN scene. Then, we generate a new set of stylized training images using the method proposed by [4, 5] using a reference image. The appearance latent vectors and the rendering MLP is fine-tuned according to the new training images while keeping the density and feature weights frozen. This transfers the style from a reference image to the 3D object.

3. Quantitative Analysis

To quantitatively compare our method on the LLFF Dataset [7], we hand-annotate the segmentation masks for the prominent objects in the CHESS TABLE , COLOR FOUNTAIN , STOVE and SHOE RACK scenes. Tab. 1 reports the segmentation metrics for the four scenes. In our method, to predict the segmentation mask, we threshold α to be greater than 0.1 while rendering. This removes the low volumetric density seeping in that contribute negligibly in the rendered visuals.

4. Region Growing: Bilateral Growth

In this section, we discuss the effect of bilateral filtering on the radiance fields and how it improves the final result. Even after employing an efficient feature-matching technique, we often obtain a high-confidence volumetric region with missing constituting parts. This is because the content search solely depends on feature distances while ignoring the spatial priors. To resolve this issue we resort to Bilateral search which exploits spatio-semantic domain priors resulting in accurate segmentation constituting all the desired regions of the semantic object. This is demonstrated in Fig. 3, where the initial high-confidence region misses the outer leaf of the dry plant. While the bilateral region is growing, we iteratively add more details into the extracted region, finally obtaining desired volumetric content. This content can be further used for various purposes as discussed in Sec. 2.

5. Evaluation strategies against SOTA techniques

5.1. N3F/DFF

As mentioned in the main document, we experiment with various thresholds in the case of N3F/DFF [6,12]. We report the quantitative metrics (Tab. 1) of our method against the best results of their methods. N3F/DFF don't produce good results for any threshold as shown in Fig. 5.

(a) Rendered Image

(b) High Confidence Region

(c) Iteration 1

(d) Iteration 2

Figure 3. *Region Growing*: Image (a) is the reference rendered viewpoint. Image (b) is the high confidence region which misses out frontal region of the dry-leaf when extracting the content. Image (c) shows the result obtained after the first iteration of bilateral filtering, which captures most of the desired region of the leaf. Image (d) is the result of the bilateral filtering applied for the second time to include intricate details such as strands around the dry-leaf.

Scene	Metric	N3F	Ours (Patch)	Ours (Stroke)
CHESS TABLE	Mean IoU ↑	0.344	0.864	**0.912**
	Accuracy ↑	0.820	0.985	**0.990**
	mAP ↑	0.334	0.874	**0.916**
COLOR FOUNTAIN	Mean IoU ↑	0.871	**0.927**	**0.927**
	Accuracy ↑	0.979	**0.989**	**0.989**
	mAP ↑	0.871	**0.927**	**0.927**
STOVE	Mean IoU ↑	0.416	**0.827**	0.819
	Accuracy ↑	0.954	**0.992**	**0.992**
	mAP ↑	0.387	**0.824**	0.817
SHOE RACK	Mean IoU ↑	0.589	0.763	**0.861**
	Accuracy ↑	0.913	0.965	**0.980**
	mAP ↑	0.582	0.773	**0.869**

Table 1. This table denotes the Mean IoU (Intersection Over Union), Accuracy and Mean Average Precision measurements for the four LLFF scenes shown in the main paper. The ground truth segmentation masks have been hand-annotated for comparison.

(a) With Steel Balls

(b) Without Steel Balls

Figure 4. *Removal of Steel Balls*: We use the MipNeRF360 [1] formulation in voxel space for unbounded 360 degree scenes. This gives fewer number of voxels to the background objects compared to the central volume of interest. In this scene, we remove the steel balls appearing in the background region of the scene.

NVOS		Ours(NVOS Stroke)		Our best	
mIOU	mAcc	mIOU	mAcc	mIOU	mAcc
70.1	92.0	83.75	96.4	90.8	98.2

Table 2. Quantitative metrics(*mIOU and mAcc)* of NVOS against Ours using NVOS provided strokes and additional strokes using our interactive feedback tool

5.2. NVOS

To make a fair comparison against NVOS [9], we utilize the masks provided by NVOS and evaluate the quantitative numbers on their dataset. We observe that our method out performs NVOS both qualitatively and quantitatively as shown Fig. 7 and Tab. 2 even when using their strokes. Using our own interactive tool with additional strokes achieve much better results.

6. Interactive Segmentation

Our method provides interactive segmentation capabilites to the user with the incorporation of positive and negative brush strokes similar to GrabCut [10].

Upon the addition of a new positive stroke, a new segmentation mask b_p is calculated using the procedure described in the main paper. The user has the option to grow this new region using bilateral filtering until not required. The new segmentation mask b_{new} is given by $b \cup b_p$.

When the user adds a negative stroke, a new segmentation mask b_n is calculated. Similar to a positive stroke, the

(a) CHESS TABLE (b) COLOR FOUNTAIN (c) STOVE (d) SHOE RACK

Figure 5. *N3F/DFF Results*: In this figure we show result of DFF/N3F [6, 12] on different thresholds and we reported the best of their method in main document. It can be seen that despite varying the thresholds the result is poorly segmented. The background objects are starting to bleed into the foreground. For the results of our method on the same scenes, please refer to the main paper.

user has the option to grow this region using bilateral filtering until not required. The new segmentation mask b_{new} is given by $b \cap (b \cap b_n)'$ (X' denotes the complement of X).

7. Critical Analysis

7.1. DINO Features

The teacher DINO features calculated on the training set of images are for patches of size 8x8. This method associates a total of 64 pixels to the same feature vector. As shown in Fig. 6, the teacher features appear to be in low resolution due to this. When performing the teacher-student training using the joint loss function, the features learnt by the student are finer in detail due to assistance from volumetric density. Hence, the student surpasses the teacher during distillation. This is evident from Fig. 6 as features are allocated with distinct boundaries in the voxel space.

Step	Time Taken
Pre-training radiance field	7 mins
Training feature field	2.5 mins
K-Means Clustering	2 secs
3D Feature Query	1 secs
Bilateral Region Growing	0.3 secs

Table 3. Timings of different steps of the ISRF pipeline

7.2. Finer Segmentation

Our method can segment out fine-grained details such as the ribs of a T-Rex as shown in Fig. 2. However, it requires accurate 3D information to achieve this. In the T-Rex scene, the tail-bones cannot be distinguished from the wall behind, since the training set images do not cover views which indicate the separation. Therefore, the optimized model containing the wall and the tail bones lie at similar depths as shown in Fig. 2d. Use of additional images covering more viewpoints can circumvent this issue.

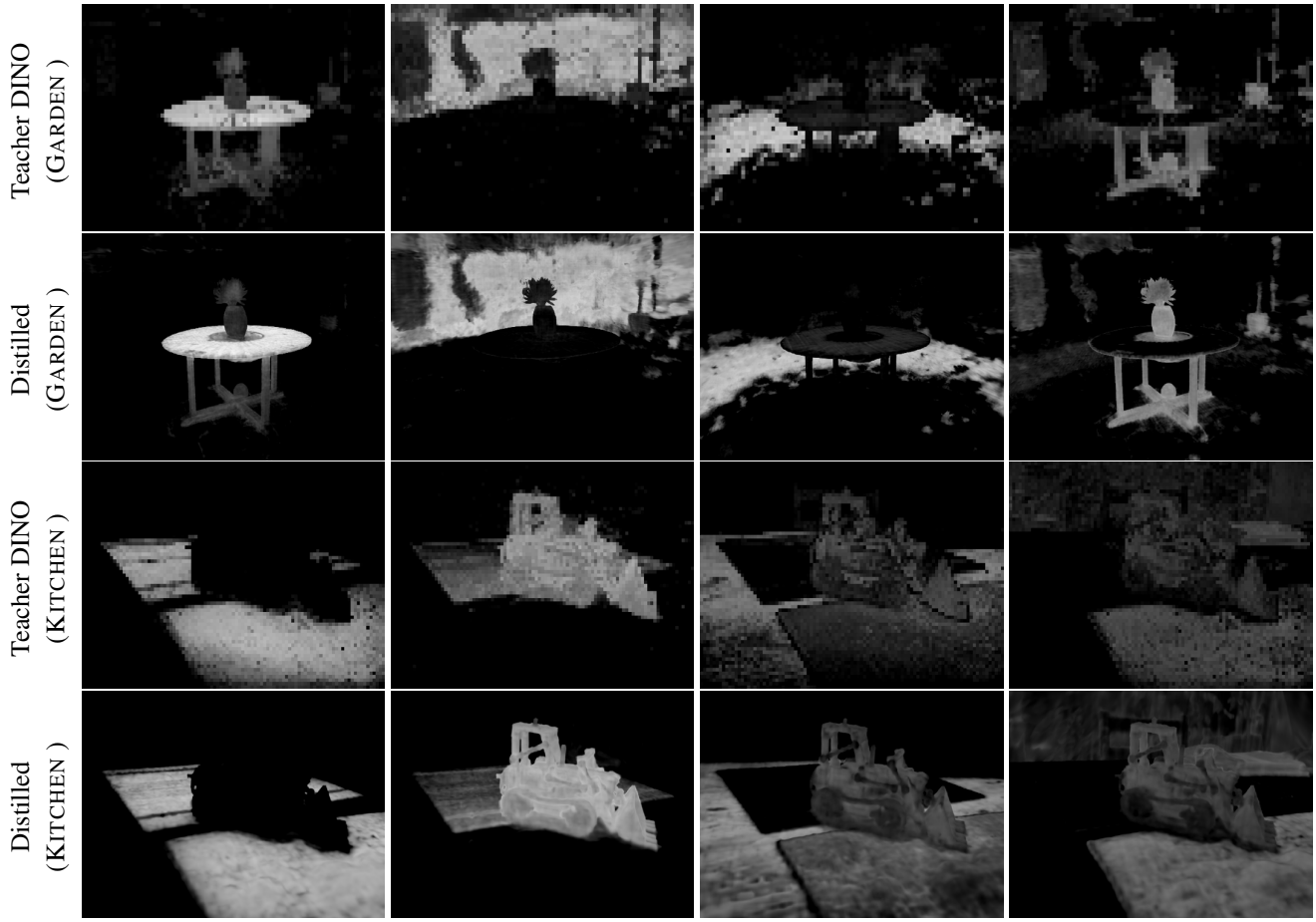

Figure 6. *Student Surpasses Teacher*: The 4 columns of this figure shows the DINO features used as teacher vs the ones learnt by student post optimization. Since, the student learns finer features than the teacher due to assistance from the volumetric density, we can claim that the student surpasses the teacher. This is consistent with the prior art N3F and DFF.

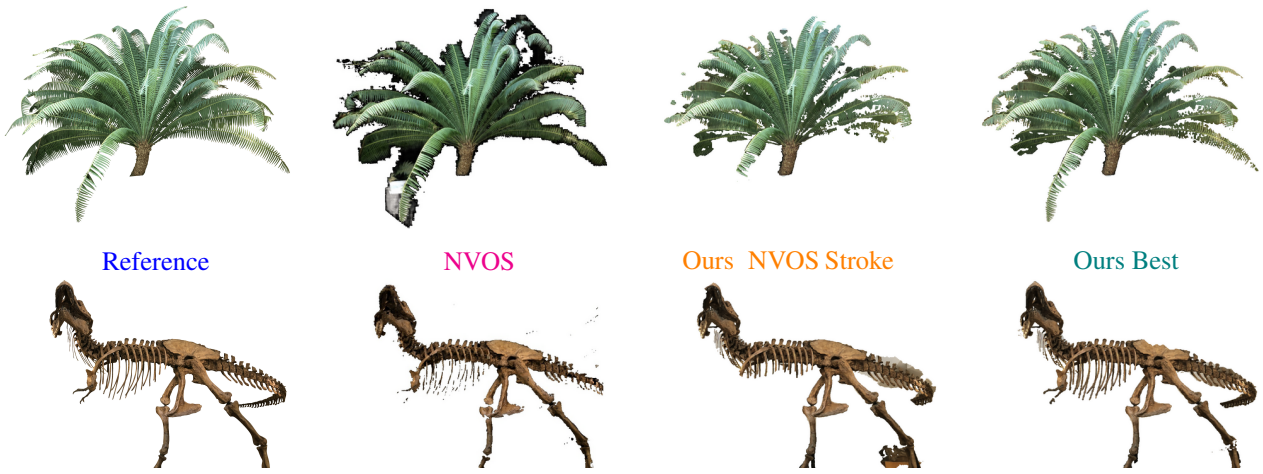

Figure 7. *left to right:* Reference segmentation using NVOS professionally segmented mask, Result of NVOS [9], Our result using NVOS stroke, Our result using additional strokes. The quantitative comparisons are mentioned in the main document where our method performs better than NVOS even when using NVOS strokes. Please zoom using *Adobe Acrobat/Okular* reader to see the details.

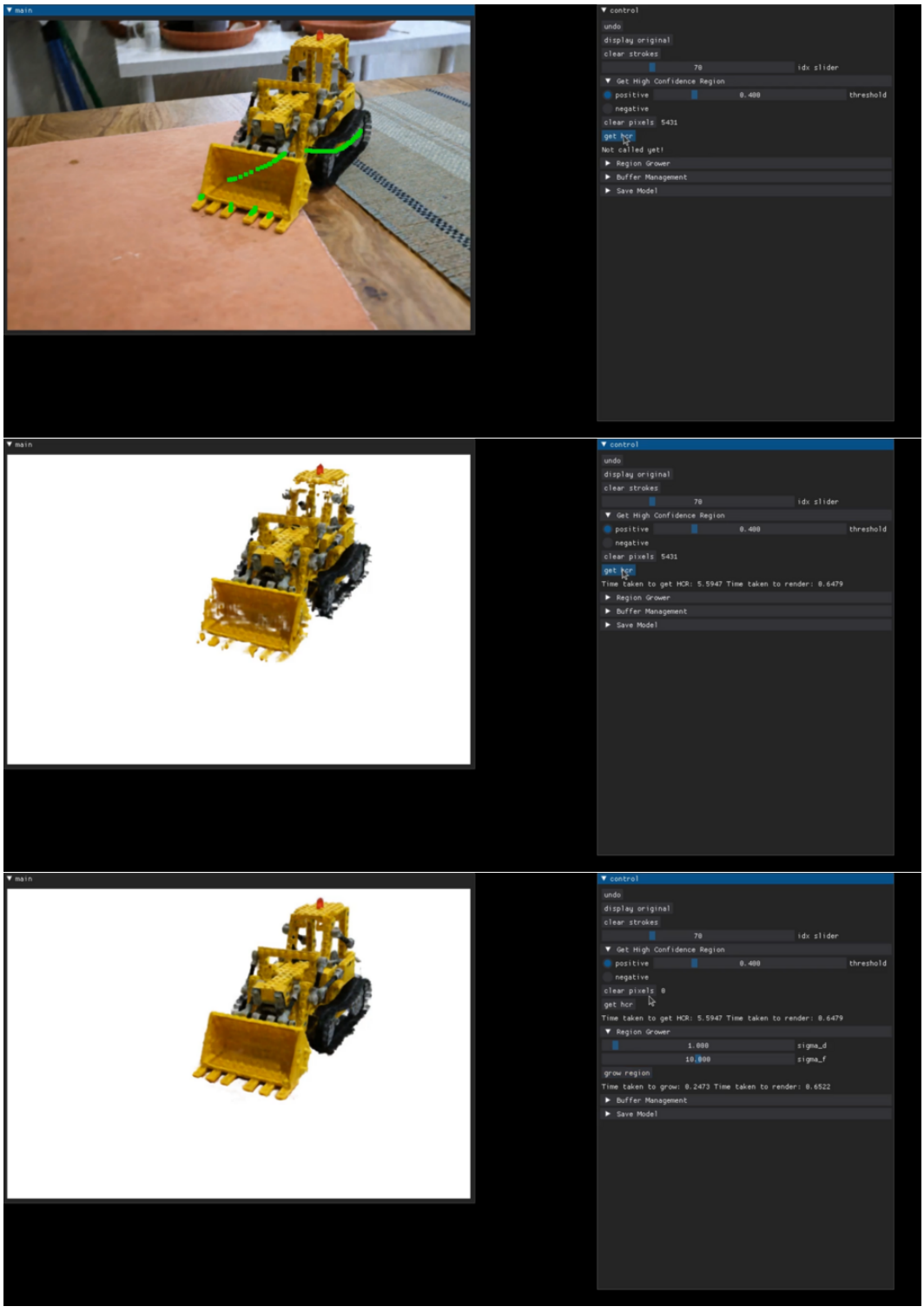

Figure 8. *Interactive GUI Tool:* We also release an easy-to-use interactive GUI tool which can be used to draw strokes and segment radiance fields.

References

[1] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1

[3] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. TensoRF: Tensorial Radiance Fields. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 1

[4] Rahul Goel, Dhawal Sirikonda, Saurabh Saini, and P. J. Narayanan. StyleTRF: Stylizing Tensorial Radiance Fields. In *Proceedings of the Thirteenth Indian Conference on Computer Vision, Graphics and Image Processing*, ICVGIP '22, 2022. 2

[5] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 2

[6] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing NeRF for Editing via Feature Field Distillation. In *Adv. Neural Inform. Process. Syst.*, 2022. 1, 2, 4

[7] 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 Trans. Graph.*, 2019. 2

[8] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 1

[9] Zhongzheng Ren, Aseem Agarwala, Bryan Russell, Alexander G. Schwing, and Oliver Wang. Neural Volumetric Object Selection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3, 5

[10] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "GrabCut": interactive foreground extraction using iterated graph cuts. *ACM SIGGRAPH 2004 Papers*, 2004. 3

[11] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Improved Direct Voxel Grid Optimization for Radiance Fields Reconstruction. *arXiv,abs/2206.05085*, 2022. 1

[12] Vadim Tschernezki, Iro Laina, Diane Larlus, and Andrea Vedaldi. Neural Feature Fusion Fields: 3D distillation of self-supervised 2D image representations. In *International Conference on 3D Vision (3DV)*, 2022. 1, 2, 4

[13] Tianhao Wu, Fangcheng Zhong, Andrea Tagliasacchi, Forrester Cole, and Cengiz Oztireli. D^2NeRF: Self-Supervised Decoupling of Dynamic and Static Objects from a Monocular Video. In *Adv. Neural Inform. Process. Syst.*, 2022. 1