SANeRF-HQ: Segment Anything for NeRF in High Quality

Supplementary Material

1. Implementation Details

We use torch-ngp [6] as our initial NeRF implementation. When we use 3D points as prompts in evaluation, the views containing less than k visible points get filtered out automatically and will not be used to train the object field, where k is a hyperparameter, depending on the total number of input points.

For both the SAM feature field and the object field, we use a hash grid as in [7] with 16 levels and feature dimension of 8 per level. The lowest and highest level are of resolution 16 and 2^{19}, respectively. We use a 5-layer 256-hidden dimensional MLP with skip connections and Layer Normalization after the feature field hash grid, and a 3-layer 256 hidden dimensional MLP with skip connections after the object field hash grid. In addition to the features from their respective hash grid, both MLPs also take the features from the density field as input, where feature MLP also takes the viewing directions as input. The initial radiance and density field, the SAM feature field, and the object field are trained for 15,000, 5,000, and 600 iterations, respectively. All models are trained on an NVIDIA RTX 4090 GPU.

Ray-Pair RGB loss is included after 300 iterations of warm-up. We use error maps downsampled by 4 times compared to original training images for Ray-Pair RGB loss sampling. In each iteration, we update the error maps using the training ray batch, and for every 200 iterations, we perform a full update for all error map pixels. During sampling, we independently sample initial rays on each error map weighted by their errors, reproject them onto each view, and subsequently sample 32 additional rays in each $N \times N$ patch centered at the reprojected pixels randomly. Here we choose $N = 8$ or 16. A subset of 20 rays are then sampled from each set as references in the Ray-Pair RGB loss.

2. Efficiency Evaluation

To provide a more comprehensive understanding of the two methods storing SAM features mentioned in Section 3.1, we evaluate the efficiency of the feature distillation method and the caching method. We randomly sample three scenes from the Mip-NeRF 360 dataset as reference. By default, the pre-trained NeRF renders images at 512×512 as input to the SAM encoder. Under a batch size of 4,096 and a maximum iteration of 5,000, it requires on average 666.0 seconds to train the feature field for a single scene, which can then render feature maps at 64×64 resolution from any viewpoints at 22.4 frames per second (FPS). In contrast, the caching method can encode the images to feature maps at 3.78 FPS while using extra memory to store the SAM feature maps (64×64, around 4.1MB each frame). Different from encoding, the decoding process is much faster, at 168.9 FPS with the pre-computed feature maps.

3. Comparison with Instance Segmentation Methods

We also compare our method with some instance segmentation methods. The instance segmentation methods in NeRF mentioned in our related works do not require user prompts and can automatically generate segmentation of salient objects in NeRF. These methods also leverage 2D segmentation methods for NeRF training but they mainly focus on the challenge of 3D consistency. Despite their different configurations and issues of concern, we still provide the comparison with these automatic end-to-end pipelines, showing that our prompt-based method can produce comparable results to these state-of-the-art auto-segmentation methods. Instance-NeRF [4] is a training-based methods so we only compare with it on 3D-FRONT dataset. Figure 1 and Table 1 illustrates the visual results and quantitative comparison respectively. For Panoptic Lifting [5] and Contrastive Lift [2], we also compare on the scenes they mentioned in the papers to ensure the fairness. Results are shown in Figure 2 and Table 2.

We use the objects in our evaluation sets as targets and choose the object that has the largest IoU with the target object as the predicted results of the instance segmentation methods. Notice that we only compare with those methods on the datasets mentioned in their papers, since they do not leverage SAM to achieve zero-shot generalization.

Metrics	Ours	Instance-NeRF
Acc.↑	98.7	99.2
mIoU.↑	89.9	92.8

Table 1. **Comparison with Instance-NeRF on 3D-FRONT**.

Metrics	Ours	Panoptic Lifting	Contrastive Lift
Acc.↑	99.6	94.3	94.1
mIoU.↑	91.1	84.5	81.5

Table 2. **Comparison with Panoptic Lifting and Contrastive Lift.** The results are on the data mentioned in their papers.

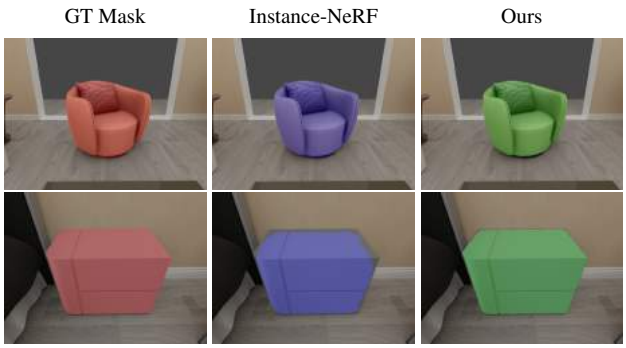

Figure 1. **Qualitative Comparison with Instance-NeRF.** Zoom in for details especially along the segmentation boundaries.

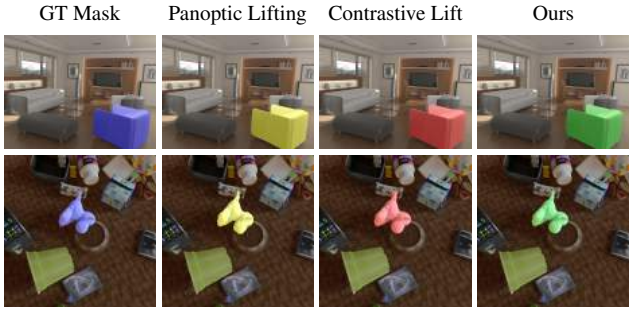

Figure 2. **Qualitative Comparison with Panoptic Lifting and Contrastive Lift**. Zoom in for details.

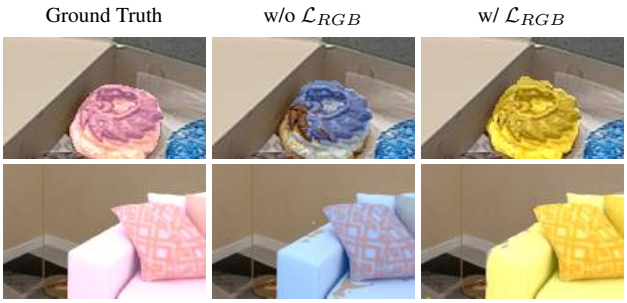

Figure 3. **Qualitative Results of the Ray-Pair RGB Loss.** The Ray-Pair RGB loss can help to recover local regions and make the results more solid.

4. Extending to Dynamic NeRFs

We present a preliminary demonstration in Figure 4 on the easy extension of our method to 4D dynamic NeRF representations. We use HyperReel [1] as our reference NeRF representation and only supply user prompts for the first frame of each camera. The prompts are fed into SAM to retrieve initial masks, whose bounding boxes are used as the prompts for the next frame. This process repeats until masks are acquired from all video frames, after which we proceed to object field training as in previous static scene cases. The scene is from the Neural 3D Video dataset [3].

5. More Qualitative Results

We demonstrate the qualitative results of the Ray-Pair RGB loss in Figure 3. The loss helps fill in the missing interior and boundaries of the masks by enforcing a local match between the similarity in labels, and the similarity in appearance.

We also provide extra qualitative comparisons between our method and other zero-shot 3D segmentation methods mentioned in the main paper. The results are given in Figures 5, 6, 7, 8. Please watch the video for more qualitative results.

6. Limitations

Though our method works well in most cases, it relies on NeRF and SAM, and its performance might be impacted by scene complexity and NeRF quality. On the other hand, the Ray-Pair RGB loss may not handle all circumstances especially given neighboring objects with identical colors and shading. Nevertheless, we present some results of our method on relatively challenging scenes to show that it may still robustly handle some of these cases, where the target objects are relatively small, in the background, partially occluded, or adjacent to other objects with similar appearance. The results are in Figure 9 and 10. We leave relevant potential improvements as future work.

(a) $t = 0s$ (b) $t = 0.25s$ (c) $t = 0.5s$ (d) $t = 0.75s$ (e) $t = 1s$ (f) $t = 1.25s$ (g) $t = 1.5s$

Figure 4. **Demonstration of Applying SANeRF-HQ to Dynamic NeRFs.** The first row are the NeRF RGB images over time, and the second row are the masks from SANeRF-HQ, which is also dynamic. Our method can be easily adapted to dynamic NeRFs and still retains reasonable performance. The implementation is based on HyperReel, and the *cook spinach* scene shown is from the Neural 3D Video dataset.

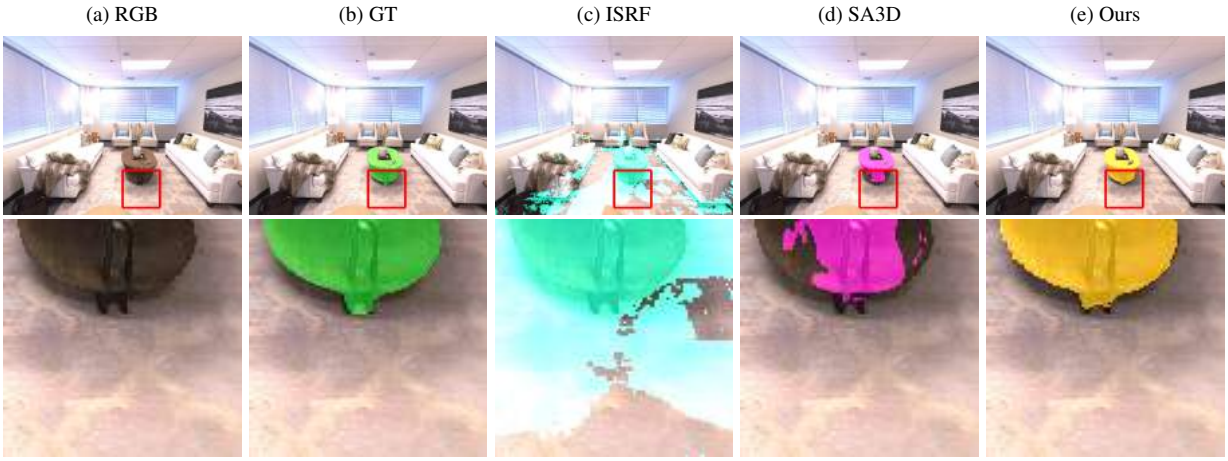

Figure 5. **Comparison with SA3D and ISRF on the Replica Room.** Data is from the Others subset. SANeRF-HQ can maintain the object structure while excludes the background.

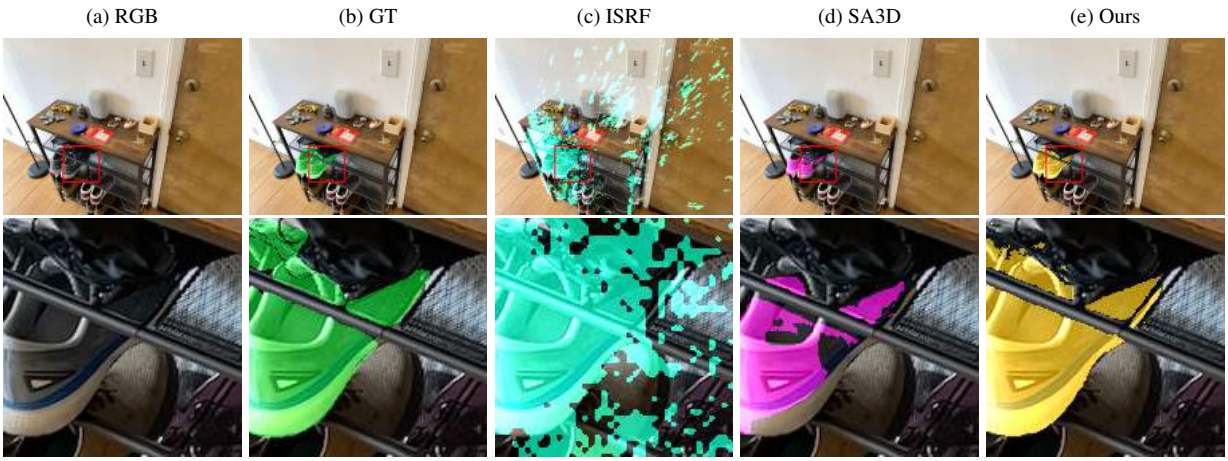

Figure 6. **Comparison with SA3D and ISRF on the Shoe Rack.** Data is from the LERF subset. Our method can reproduce the segmentation details even with some occlusion.

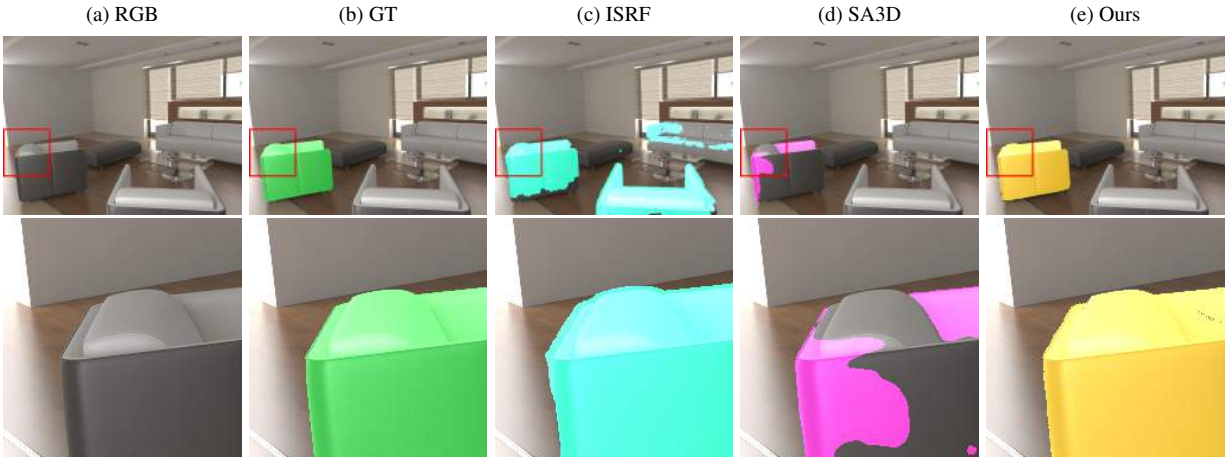

Figure 7. **Comparison with SA3D and ISRF on Hypersim.** Data is from the Others subset. ISRF contains too many false positives, while SA3D cannot cover the whole object.

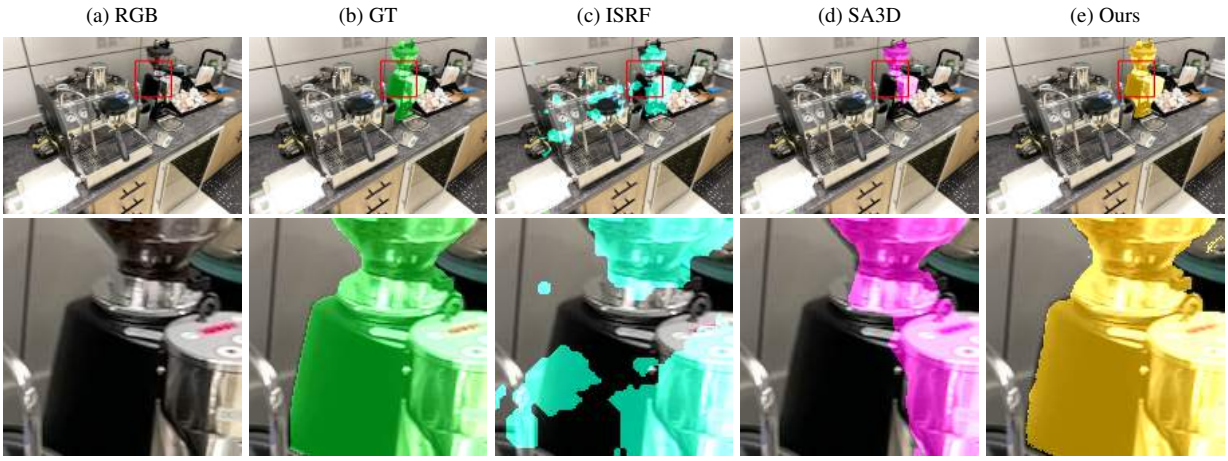

Figure 8. **Comparison with SA3D and ISRF on the Espresso.** Data is from the LERF subset. Our method produces the most reasonable segmentation in the distant, complex setting.

Figure 9. **Examples of More Complex Scenarios.** SANeRF-HQ can effectively segment target objects that are in the background, relatively small, and partially occluded.

Figure 10. **Examples of Objects with Similar Color.** Our method can still distinguish these objects and produce reasonable results in the presence of neighbouring objects with similar appearance, where the Ray-Pair RGB loss is less helpful but remains robust.

References

[1] Benjamin Attal, Jia-Bin Huang, Christian Richardt, Michael Zollhoefer, Johannes Kopf, Matthew O'Toole, and Changil Kim. HyperReel: High-Fidelity 6-DoF Video with Ray-Conditioned Sampling. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[2] Yash Bhalgat, Iro Laina, João F Henriques, Andrew Zisserman, and Andrea Vedaldi. Contrastive Lift: 3D Object Instance Segmentation by Slow-Fast Contrastive Fusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 1

[3] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, et al. Neural 3D Video Synthesis from Multi-view Video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5521–5531, 2022. 2

[4] Yichen Liu, Benran Hu, Junkai Huang, Yu-Wing Tai, and Chi-Keung Tang. Instance Neural Radiacne Field. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 1

[5] Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulò, Norman Müller, Matthias Nießner, Angela Dai, and Peter Kontschieder. Panoptic Lifting for 3D Scene Understanding With Neural Fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9043–9052, 2023. 1

[6] Jiaxiang Tang. Torch-ngp: a PyTorch Implementation of Instant-ngp, 2022. https://github.com/ashawkey/torch-ngp. 1

[7] Thomas Müller and Alex Evans and Christoph Schied and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM TOG*, 41(4):102:1–102:15, 2022. 1