	MipNeRF 360 [2]			LERF-Masked [44]			LLFF [27]		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Radiant Foam	29.92	0.83	0.21	22.73	0.79	0.38	24.60	0.74	0.34
Semantic Foam	29.79	0.90	0.17	22.72	0.79	0.38	24.59	0.74	0.34

Table 3. **Reconstruction metrics –** We show that we maintain the reconstruction quality of Radiant Foam with our Semantic Foam model while simultaneously learning semantic segmentation.

5.1. Additional implementation details

Training. The training pipeline utilizes the Adam optimizer [23]. Similar to Radiant Foam [12], we directly optimize per-point position, density, and view-dependent color, the latter being represented via spherical harmonics (SH) of degree three, in conjunction with identity encodings. Optimization of point coordinates commences with an initial learning rate of $2e^{-4}$ and is annealed using a cosine schedule to a minimum rate of $2e^{-6}$. The initial learning rates for point density and spherical harmonics are set to $1e^{-1}$ and $5e^{-3}$, respectively. These rates are also decayed via a cosine annealing schedule to a final rate equal to 0.1 times the initial rate. Consistent with Radiant Foam, we initially optimize only the zero-order (ambient) component of the SH coefficients. Optimization for the high-order coefficients is subsequently introduced after a warmup period, spanning the first 25% of the total training iterations. For identity encodings, we start with a learning rate of $5e^{-3}$ and decay it to a final learning rate of $5e^{-4}$ following the same cosine annealing schedule.

Following Radiant Foam, once initialization and warm-up training are complete, we progressively increase the number of Voronoi sites, linearly expanding the point set until the target resolution is reached. The adjacency data structure is updated using the same schedule to ensure consistency throughout training. All experiments are run for 20k iterations, with the final 2k iterations refining only the radiance and density attributes while keeping point positions fixed.

For the total variation loss, we halt gradient propagation through the face area and clamp this value to a minimum of 1, ensuring a stable and consistently non-zero penalty across adjacent points. We likewise stop density gradients from flowing through the identity loss to avoid unintended density variations arising from noise in the segmentation masks. This strategy preserves the structural and geometric fidelity learned from image renderings while enabling optimization of identity encodings for scene semantics.

5.2. Reconstruction metrics

In Table 3, we compare our reconstruction metrics against the original Radiant Foam. Our results demonstrate that the proposed method preserves reconstruction fidelity to a nearly identical degree while simultaneously enabling the learning of semantic information.

	LabelGS [45] mIoU↑ / mAcc↑	Gaussian Gouping [44] mIoU↑ / mAcc↑	SAGA [4] mIoU↑ / mAcc↑	SemanticFoam mIoU↑ / mAcc↑
Garden	0.79 / 0.95	0.88 / 0.91	0.78 / 0.94	**0.94 / 0.96**
Bonsai	0.69 / 0.92	0.77 / 0.84	0.70 / **0.97**	**0.90** / 0.94
Room	0.64 / 0.93	0.56 / 0.69	**0.80 / 0.98**	0.62 / 0.91
Counter	0.55 / 0.93	0.74 / 0.92	**0.81 / 0.96**	0.75 / 0.91
Kitchen	0.84 / 0.97	0.83 / 0.91	**0.95 / 0.99**	0.91 / 0.94

Table 4. **Per-scene metrics –** Per-scene metrics – mIoU, and mAcc scores for MipNerf360 [2] scenes.

	LabelGS [45] mIoU↑ / mAcc↑	Gaussian Gouping [44] mIoU↑ / mAcc↑	SAGA [4] mIoU↑ / mAcc↑	SemanticFoam mIoU↑ / mAcc↑
Figurines	0.60 / 0.97	0.70 / 0.81	0.86 / **0.98**	**0.89** / 0.94
Ramen	0.29 / 0.91	0.88 / 0.96	0.54 / **0.99**	**0.87** / 0.89
Teatime	0.73 / 0.93	0.71 / 0.80	0.76 / **0.99**	**0.79** / 0.88

Table 5. **Per-scene metrics –** Per-scene metrics – mIoU, and mAcc scores for LERF-Masked [44] scenes.

	LabelGS [45] mIoU↑ / mAcc↑	Gaussian Gouping [44] mIoU↑ / mAcc↑	SAGA [4] mIoU↑ / mAcc↑	SemanticFoam mIoU↑ / mAcc↑
Fern	0.84 / 0.99	**0.90 / 0.98**	0.89 / 0.97	0.89 / 0.96
Flower	0.72 / 0.78	0.76 / 0.81	**0.94 / 1.00**	0.77 / 0.80
Fortress	**0.98 / 1.00**	**0.98 / 1.00**	**0.98 / 1.00**	**0.98 / 1.00**
Horns	0.82 / **1.00**	0.89 / 0.99	**0.85** / 0.99	0.86 / 0.99
Leaves	0.09 / 0.13	0.85 / 0.98	0.51 / 0.96	**0.97 / 1.00**
Orchids	**0.82 / 0.95**	0.80 / 0.94	0.42 / 0.44	0.60 / 0.77
Room	0.87 / **1.00**	0.98 / **1.00**	**0.99 / 1.00**	0.97 / 0.97
Trex	0.50 / 0.59	0.55 / 0.59	0.17 / 0.17	**0.64 / 0.93**

Table 6. **Per-scene metrics –** Per-scene metrics – mIoU, and mAcc scores for LLFF [27] scenes.

5.3. Per scene metrics

Tables 4 to 6 summarize the segmentation metrics collected for our evaluation of all considered techniques. These include results for Mip-NeRF360 [2], LERF-Masked [44] and LLFF [27] scenes.

5.4. Training time and memory consumption

In Table 7, we report the average training time, inference speed, and memory usage of Semantic Foam compared to Radiant Foam, evaluated on the LERF-Masked [27] dataset using an NVIDIA RTX A6000 GPU. Our model preserves the training and inference efficiency of Radiant Foam, while incurring an 18% increase in model size due to the inclusion of identity features required for the segmentation task.

	Training Time (mins ↓)		Model Size (MB ↓)		Inference Speed (FPS ↑)	
	Radiant Foam	Semantic Foam	Radiant Foam	Semantic Foam	Radiant Foam	Semantic Foam
Figurines	83.00	84.00	661.00	783.00	84.52	84.17
Ramen	80.00	80.00	655.00	778.00	59.75	59.79
Teatime	79.00	79.00	663.00	785.00	91.04	90.39
Average	80.67	81.00	659.67	782.00	78.44	78.11

Table 7. **Model and training statistics**

5.5. Additional Quantitative Comparisons

In this section, we compare our model with InstanceGaussian and GARField. InstanceGaussian is designed primarily for open-language grounded evaluation, which differs fundamentally from the click-based object evaluation framework utilized by our method and existing baselines. We adapt the evaluation protocol from InstanceGaussian to facilitate a direct comparison on the LERF-Masked dataset, as shown in Table 8. We also evaluate our method against GARField – a recent state-of-the-art NeRF-based semantic segmentation method – on the LERF-Masked dataset. As detailed in Table 8, our method surpasses InstanceGaussian and GARField in average mIoU metrics.

	Figurines	Ramen	Teatime	Average
InstanceGaussian	0.68/**0.97**	0.52/**0.92**	**0.79/0.99**	0.66/**0.98**
GARField	0.70/0.72	0.53/0.59	0.68/0.68	0.64/0.70
Ours	**0.89**/0.94	**0.87**/0.89	**0.79**/0.88	**0.85**/0.91

Table 8. **Additional quantitative comparisons (mIoU↑/mAcc↑)**

5.6. Additional qualitative results

In Figure 8, we present additional qualitative results that further illustrate the high fidelity of our extracted assets and their corresponding segmentation maps. While prior methods frequently produce dilated segmentations that incorporate background regions, our approach yields clean, well-localized masks, enabling high-fidelity object extraction.

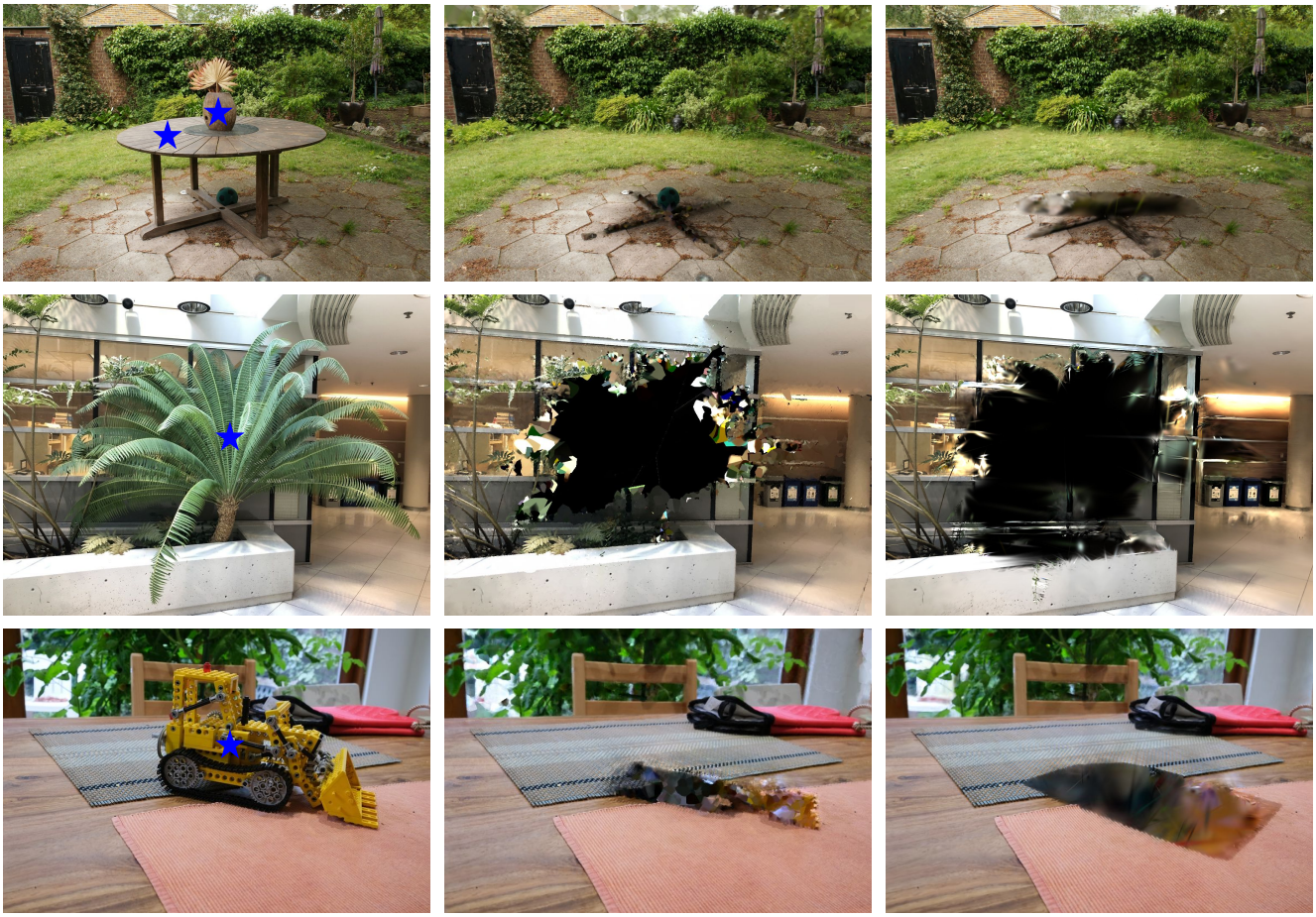
5.7. Additional scene editing results

In Figures 9 and 10, we compare the object insertion and deletion capabilities of our semantic foam representation against Gaussian Grouping. We restrict our comparison to Gaussian Grouping because SAGA and Label-GS do not provide open-source code for object-level editing. Owing to Radiant Foam's implicit surface formulation, our method enables seamless definition of 3D object masks without relying on additional post-processing steps such as the convex-hull construction required by Gaussian Grouping. As a consequence, Gaussian Grouping often over-deletes or introduces noise into the scene, since its object boundaries are restricted to convex shapes. For instance, as shown in Figure 10, our model accurately extracts only the table and pot from the Garden scene, whereas Gaussian Grouping also removes the nearby ball due to its inability to represent non-concave object masks.

Figure 8. **Extra Qualitative Results.** We present additional qualitative comparisons of our object - extraction results against Gaussian Grouping [44] and SAGA [4]. As demonstrated by the leaves and teatime extractions, Gaussian-based baselines often exhibit inconsistent segmentation boundaries; conversely, our approach generates sharp, accurately bounded masks that more faithfully preserve the integrity of the object's geometric structure.

Unedited GT | Our insertion | Gaussian Grouping insertion

Figure 9. **Scene editing (insertion) –** Comparison of object insertion between our semantic foam representation (middle) and Gaussian Grouping (right), with the (left) view showing the unedited reference image. Leveraging Radiant Foam's implicit surface formulation, our method defines accurate non-convex 3D object masks without requiring convex-hull post-processing. As shown, our approach cleanly inserts the toy and lego in the Kitchen and Fortress scenes respectively, whereas Gaussian Grouping inserts additional noise.



Unedited GT | Our deletion | Gaussian Grouping deletion

Figure 10. **Scene editing (deletion) –** Comparison of object deletion between our semantic foam representation (middle) and Gaussian Grouping (right), with the (left) view showing the unedited reference image (blue star denotes the object selected for deletion). Leveraging Radiant Foam's implicit surface formulation, our method defines accurate non-convex 3D object masks without requiring convex-hull post-processing. As illustrated, our approach accurately isolates the table and pot in the Garden scene, while Gaussian Grouping over-deletes and erroneously removes the nearby ball due to its convexity-restricted mask formulation.