InNeRF360: Text-Guided 3D-Consistent Object Inpainting on 360° Neural Radiance Fields

Supplementary Material

A. Additional Results

360-degree scenes. We encourage our readers to view the supplementary video for a comprehensive set of results from our evaluated datasets and to qualitatively assess our method. To quantitatively evaluate the output of InNeRF360 against ground truth inpainted scenes, we evaluate our method on captured real-world scenes. Fig. 10 demonstrates that even with occlusion between the plant and the object to remove, our method successfully generates clean inpainting with the rest of the scene unmodified.

To clarify our evaluation, we conduct quantitative experiments on the baselines *Ideal*, *ObjectNeRF-M*, *SPN-360-M* on our captured real-world datasets:*Starbucks*, *Glass Cat* and *4-Objects* The *ObjectNeRF-M* baseline involves training a NeRF with L2 loss outside the segmentation masks only. ObjectNeRF-M produces lower-quality inpainting results than our InNeRF360, evidencing that inpainting on 360-degree NeRF is more complicated than frontal-facing scenes. The implementation of SPN-360 is based on NeRFacto. Its segmentation masks and inpainting results are both generated by the method used in SPIn-NeRF so that we can compare them with our InNeRF360 results. This lets us contrast the performance of the complete methods. To address the concern for reliability, we provide evaluations with *SPN360-M* which uses our segmentation masks and SPIn-NeRF's inpainting technique. While SPN360-M outperforms SPN-360, it still falls short of our InNeRF360's performance. Moreover, it is less effective than NeRFactor combined with $\mathcal{L}_{\text{geom}}$, see results in Tab.1 of the main paper. To show the reliability of our numerical results, we provide the requested evaluation for *Ideal*, i.e., fitting a NeRF on ground truth scenes without the object(s) to be inpainted, which serves as an upper bound for the best inpainting results with the same inputs and NeRF architecture.

We capture ground truth datasets with the objects removed from the scene, with which we evaluate LPIPS [44] and Frechet Inception Distance (FID) [11] metric as metric. As indicated in Tab. 1 and Tab. 3, InNeRF360 outperforms the other two methods in terms of the similarity between the activations of the inpainted region and the ground truth.

Ablation on mask dilation. In Fig. 11, we demonstrate the importance of mask dilatation. We conducted experiments using different numbers of pixels on the contour for dilation, specifically with the set $\{11, 21, 41, 51, 101\}$, and determined that 51 pixels yield the best inpainting performance. Allowing for more contextual information in the 2D image inpainter encourages the inclusion of more background details, thereby enhancing consistency across different views. However, due to the limited 3D understanding of 2D image inpainters, further refinement of the initially inpainted scene is necessary.

Methods	Starbucks		Glass Cat		Multiple	
	LPIPS ↓	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓	FID ↓
Ideal	0.4016	130.79	0.3928	129.95	0.3829	124.82
ObjNeRF-M	0.6967	275.92	0.7048	288.91	0.6743	260.48
SPN360-M	0.6037	198.64	0.6542	253.74	0.6073	192.45
InNeRF360	**0.4523**	**153.46**	**0.4158**	**142.60**	**0.4264**	**147.49**

Table 3. Quantitative results of requested baselines Ideal, ObjectNeRF-M, SPN-360-M and our InNeRF360 on rw datasets.

Methods	LPIPS ↓	FID ↓
SPIn-NeRF	0.4971	149.41
Ours	**0.4764**	**129.54**

Table 4. Quantitative comparison with SPIn-NeRF on the quality of synthesized inpainting regions, averaged over the front-facing datasets that we evaluate.

Front-facing scenes. InNeRF360 works for front-facing scenes with only text instructions, and without hand-drawn masks as SPIn-NeRF. In Fig. 12, our method synthesizes inpainting content that is more perceptually consistent with the surroundings for the staircase without introducing artifacts, while SPIn-NeRF leaves the partial shadow of the box. On the bench scene, our method also gives a more perceptually robust synthesized texture to the fence. Fig. 13 show additional qualitative results of InNeRF360 on frontal datasets. Our method does not introduce visual artifacts to the inpainted regions, and we encourage the reader to inspect our supplementary video for better visualization.

For quantitative analysis, we evaluate our method on the SPIn-NeRF frontal datasets by comparing the synthesized contents in the bounding box region with the provided ground truth images, following the setup of SPIn-NeRF. As displayed in, Tab. 4, InNeRF360 synthesizes content that is closer to the ground truth data.

B. Dataset Details

We conducted experiments on ten 360-degree scenes from various datasets: *Bear*, *Vasedeck*, *Garden*, *Room*, *Bulldozer* and *Floating Tree*, as well as four captured scenes, Cup, Starbucks, Glass Cat and 4-Objects specif-

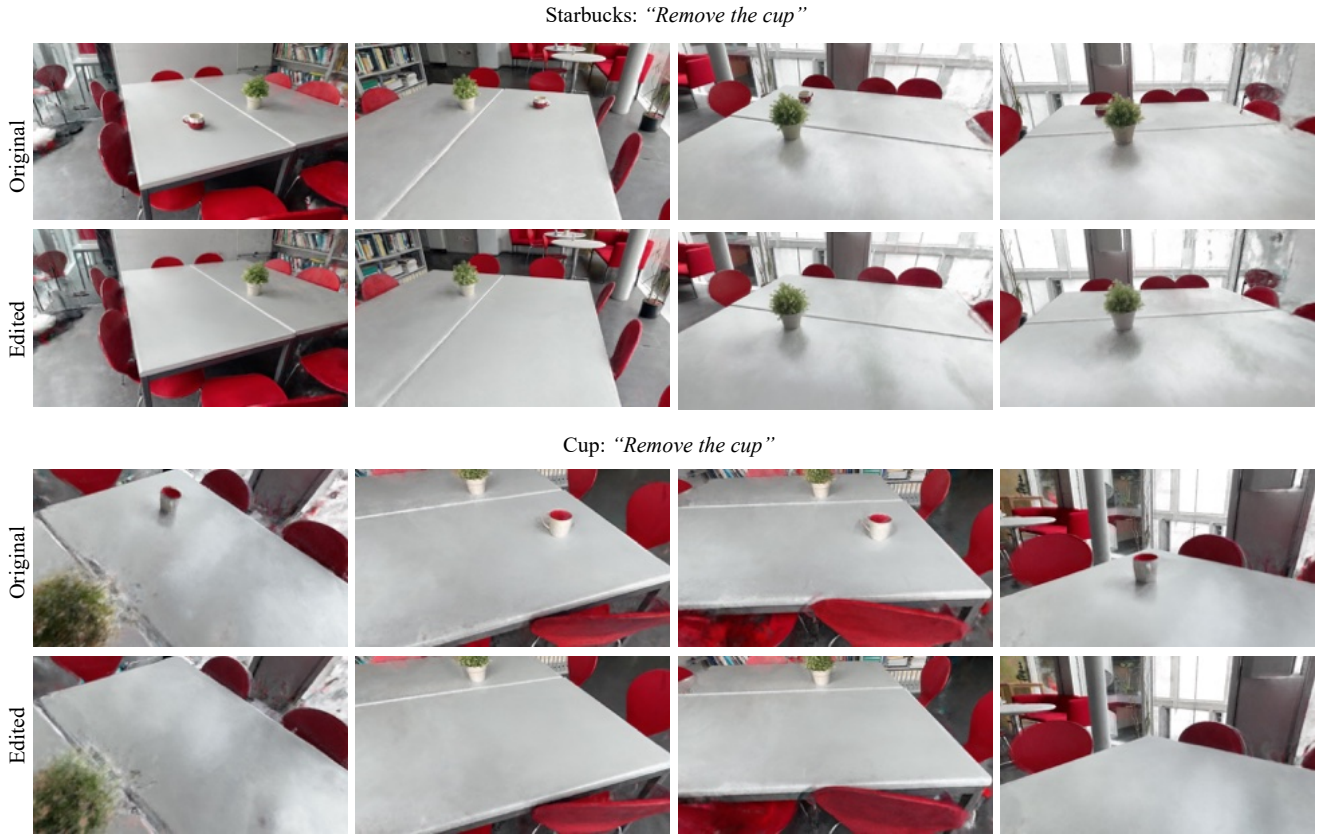

Figure 10. **Quantitative results on our captured datasets.** InNeRF360 generates consistent inpainting region on the occluded sections between the plant and the cup.

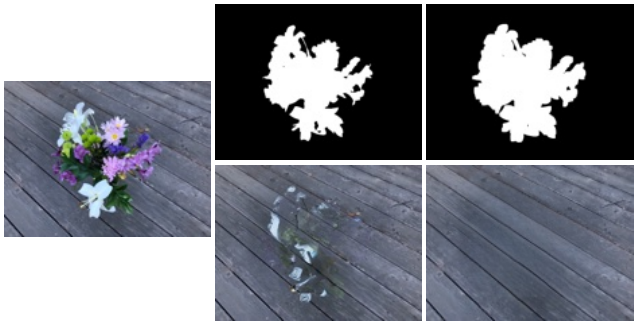

Figure 11. Contexualized segmentation. (a) Original image; (b) top: segmentation masks *without* dilation; bottom: resulting inpainted image; (c) top: segmentation masks *with* dilation; bottom: resulting inpainted image. Dilating the mask provides contextual pixel information and improves inpainting quality.

ically for quantitative evaluation purposes. We select 360° scenes containing different challenging aspects for the segmentation and inpainting tasks. *Bear* contains a large section to remove and complex background texture; *Vasedeck* contains a transparent object to select; *Room* contains multiple objects in different places in the scene; *Bulldozer* contains multiple instances of occlusion between objects to remove and other objects in the scene; *Floating Tree* contains object that does not locate on a flat surface.

Our scenes are captured using a smartphone, and the camera poses are extracted using PolyCam [32]. It's important to note that the estimated camera poses are object-centric but may contain noise, which can result in blurriness outside of the object region. Consequently, for our quantitative evaluation, we specifically focus on assessing the inpainting performance within the bounding box regions of the objects. The number of images included in each scene can be found in Tab. 5. The camera poses are sampled in the open area above and around the object(s) to be inpainted.

C. User Studies on View Consistency

We evaluate the user studies as shown in Tab. 2 with 49 users. For each scene, we present each participant with two 95-frame video clips: one rendered from our inpainted NeRF scene and the other from per-frame inpainted renderings of the original NeRF scene. Participants are asked to indicate which video appears more visually consistent and perceptually plausible. We calculate the percentage pref-

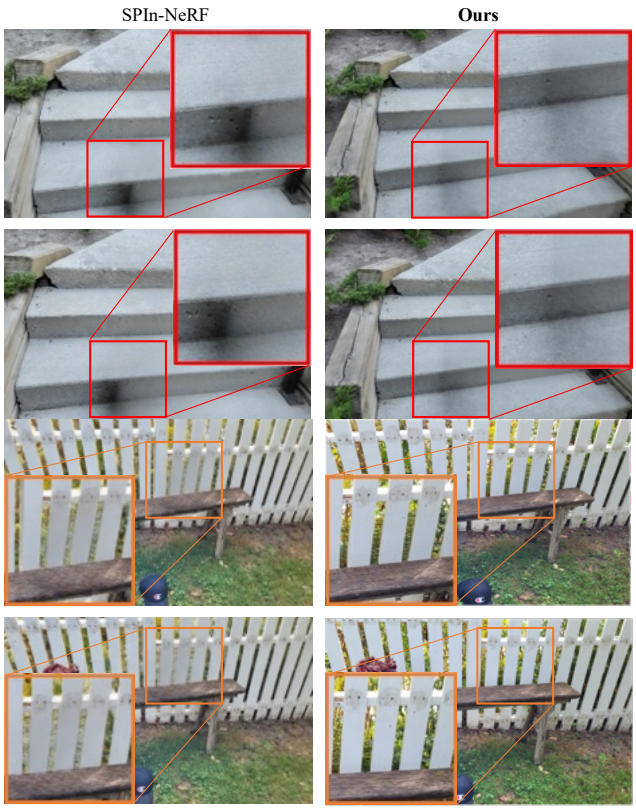

Figure 12. Qualitative comparison with SPIn-NeRF on front-facing datasets. InNeRF360 generates 3D-consistent inpaintings that contain fewer visual artifacts and are more aligned with surrounding regions.

360°	Size	Captured	Size	Frontal	Sizes
Vasedeck	116	Cup	117	[26]-(10)	60
Garden	185	Starbucks	199	Book	60
Room	311	4-Objects	206	Sink	60
Bulldozer	359	Glass Cat	136	Stairs	60
Bear	96			[43]-(001)	260
Floating Tree	96				

Table 5. Number of images in each dataset.

erence for each option by dividing the number of votes by the total number of participants. Fig. 14 provides a set of selected examples that we provide to the users.

This experiment aims to demonstrate that InNeRF360 offers **superior view-consistency** across frames compared to per-frame inpaintings. InNeRF360 achieves such performance due to our geometric and appearance refinement to the initialized NeRF from 2D inpainting. Our approach includes a geometric prior that works in 3D to remove density artifacts, and a masked LPIPS loss that ensures the inpainted region blends perceptually consistently with surrounding areas during training. Additionally, a trained

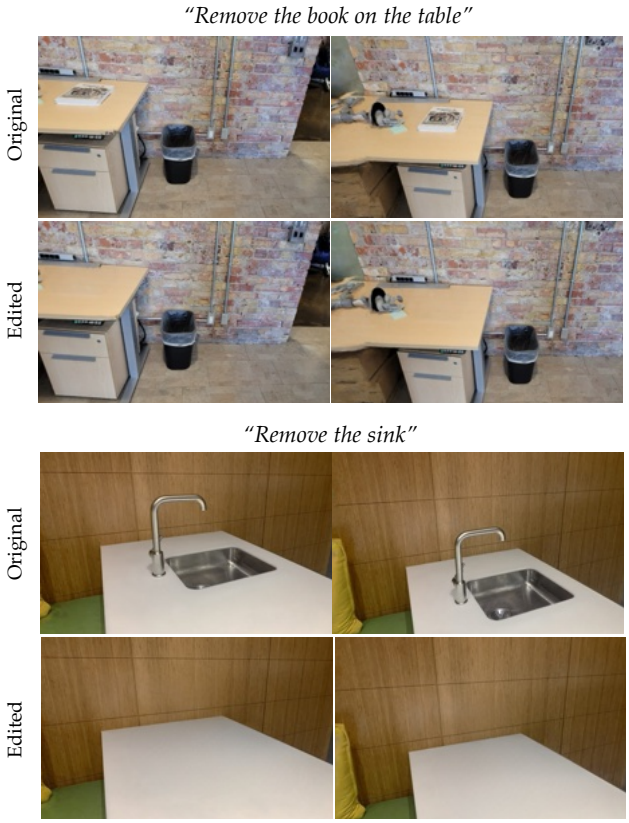

Figure 13. Qualitative results of InNeRF360 on frontal dataset scenes. More results can be found in our supplementary video.

NeRF scene inherently maintains 3D consistency. In contrast, per-frame editing relies solely on the local information of a single 2D viewpoint, lacking 3D consistency across different views. This can result in visual artifacts, which InNeRF360 effectively addresses and resolves.

D. Editing Accuracy

With a simple modification of our method by replacing the image inpainter with a mask-conditioned image editor [8], our method can produce view-consistent editing on specific objects instructed by text, as shown in Fig. 15. In this example, we do not use our $\mathcal{L}_{\text{geom}}$. Our baseline Instruct-NeRF2NeRF (In2n) [9] is also capable of stylizing NeRF scenes, but it cannot pinpoint a particular object for either removal or editing. In2n relies on InstructPix2Pix [3] and operates solely in latent space for image editing. It applies stylization to a large undesired area of the NeRF scene, as illustrated in Fig. 9. In comparison, the modified version of InNeRF360 works with object-level modification, and delivers accurate editing results that accurately address the requested object. Specifically, we utilize our 3D consistent segmentation module to output masks for the dataset images. Then our method can be connected with

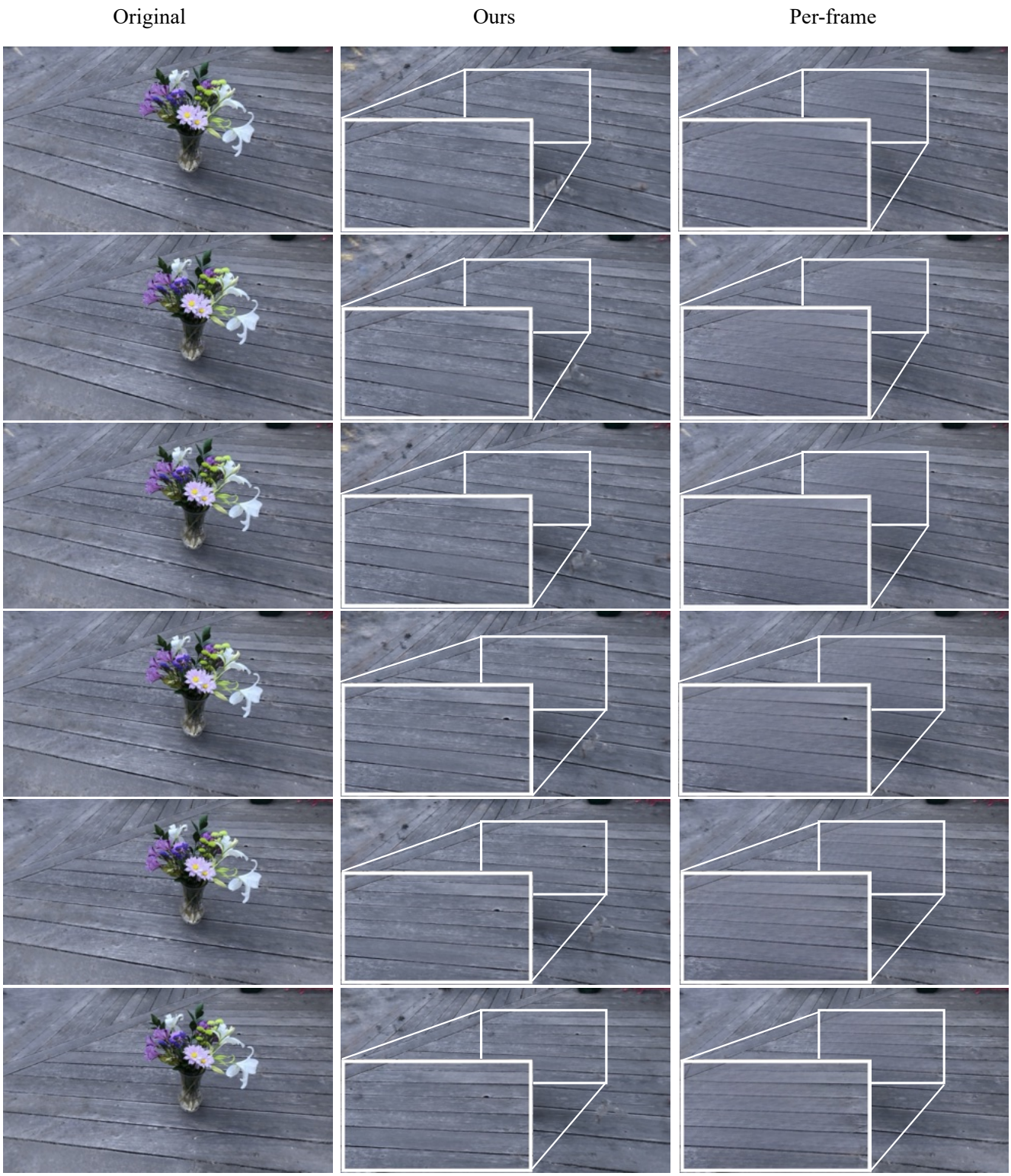

Figure 14. Qualitative comparison on view consistency across different camera poses between InNeRF360 and per-frame inpainting. The zoomed-in region in each frame shows the inpainted quality of our method and per-frame inpainting. Our method contains higher consistency across frames, while the per-frame inpainting tends to be inconsistent and blurry in the inpainted region.

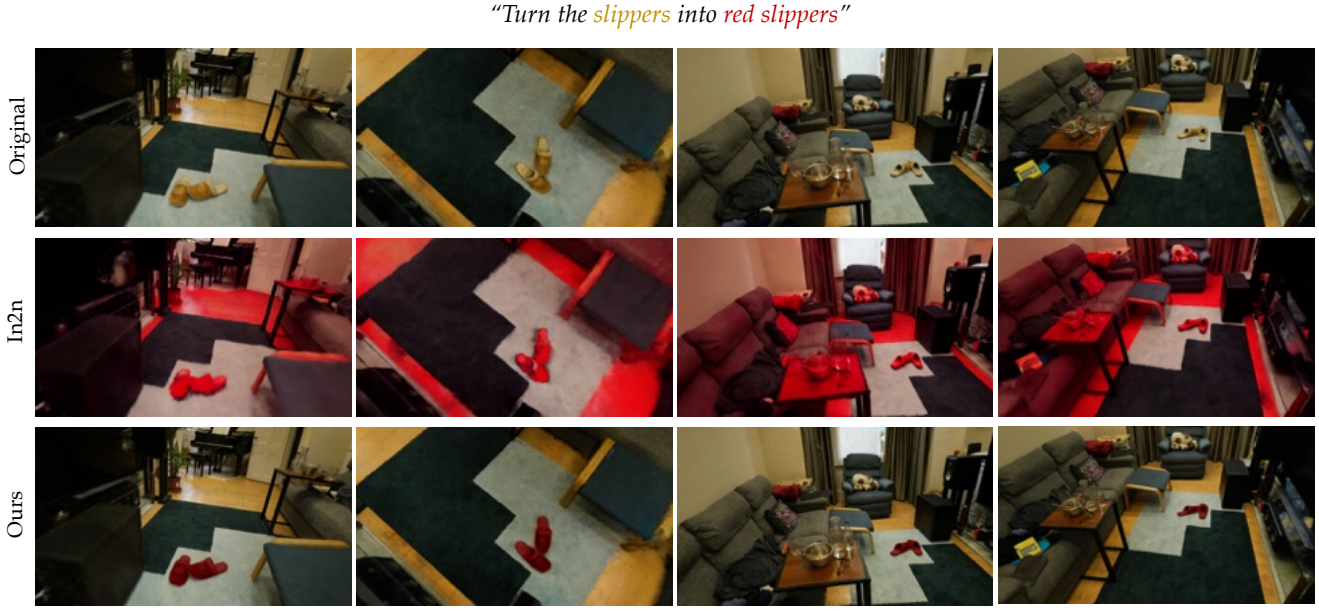

Figure 15. Comparison on In2n with a modified version of InNeRF360 on object-level stylization.

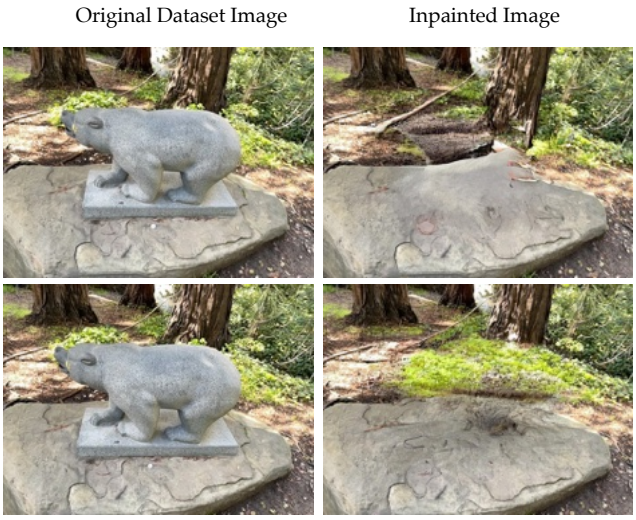

Figure 16. Examples of varying inpainted regions on the bear dataset images. We use the 2D image inpainter [36]. We can see the inconsistent inpaintings for two dataset images with similar camera origins, which can lead to concentrated artifacts near the inpainted region if trained into NeRF directly.

a mask-conditioned image editing method for object-level editing. Here we use [8], a zero-shot image editing method given a text-to-image denoising diffusion model.

Note that, however, the focus of InNeRF360 is to **remove** text-instructed object from the 360° NeRF scene. We provide such examples only to demonstrate the easily achievable extension to object-level editable NeRF with our proposed segmentation method, by making use of powerful 2D image processing tools to address challenging 3D problems.

E. 2D Inpainting Examples

In the scene of *Bear*, the object of interest takes up a large section of the image, and the background around the bear has non-uniform and highly varying patterns. In these cases, diffusion models tend to generate significant variations in pixel content for the inpainted region across different viewing perspectives, as indicated in Ln 385-386 of the main paper on examples of noisy 2D inpainted images. An example illustrating this limitation can be observed in Fig.16 (b), where the inpainted regions exhibit significant discrepancies between two adjacent viewpoints as shown in (a). There are also some results on per-frame inconsistency in our supplementary video.

F. Failure Cases

In *Vasedeck*: While the stain on the table is visually plausible and view-consistent, it is in fact the shadow of the object that was removed. Such shadow has soft edges and can easily be mistaken as part of the floor texture, and thus is overlooked by the image inpainter. Similarly in the staircase scene in frontal datasets, our method cannot completely remove soft shadow, although we produce more plausible results than the baseline method.

Figure 17. Additional segmentation results on *Bear*, *Garden* and *Vasedeck*.

G. Design Justification on Geometry Guidance

In InNeRF360, we opt not to use the inpainted depth images as inpainting priors on geometry like our prior works [25, 26, 43]. Inpainting on 2D depth maps creates inconsistency in geometry supervision between different views, similar to inpainting on RGB images in 2D. Instead, we utilize a trained 3D diffusion model as priors to super-

vise the removal of floaters, and therefore operate directly in 3D space to avoid inconsistency in geometry supervision.

On the other hand, we observe that inpainting depth maps enforce the accumulated floaters in the inpainted region to be "scattered" onto surrounding background environments, causing a blurry background similar to training with pixel-wise L1 in the inpainted region. As shown in Fig. 6, the background water pipe has been made very blurry in the output of SPN-360 which inpaints on both depth maps and RGB images. By not modifying 2D depth maps but operating on 3D space to remove density artifacts directly, we propose a method that is more effective than scattering such artifacts around into unconcentrated regions. The artifacts in the latter scenario are harder to resolve.

H. Implementation Details.

H.1 Training geometric priors

During the training on shapenet, we use voxelized cubes of $m^3 = 32 \times 32 \times 32$. We clamp the density values in voxel grids into $[0, 1]$. During inference time, the geometric prior performs one forward pass without backpropagating through the diffusion model. We set the threshold for determining whether a voxel is empty to be $\rho = 0.01$, and for floater detection and removal, we set $w = 0.02$. Looking at Eq. (6), we experiment with the value of w, which is a hyperparameter for the amount of density to be increased in the occupied voxels, and realize that it trades off with inpainting quality. Specifically, as we decrease w, more densities are guided by the diffusion priors to be removed, and thus occasionally we observe see-through surfaces in the trained scene. As we increase w, floaters artifact may not be completely removed from the scene. The learning rate t is chosen between 10 and 50. We empirically find out that for objects that take up small regions in the scene such as *Bulldozer*, the sampled cube size should be upper bounded around 5% of the scene. A more automatic way of determining the sampled cube size using the prompt and the Shapenet dataset will be interesting to explore in future work.

H.2 Depth warping refinement

For each training view, we select 8 other views with 20 points per view for depth-warping, for a balanced choice over efficiency on the iteration of dataset images and refined segmentation quality.

H.3 Training Inpainted NeRF

For the 2D image inpainter, we adapt the open-sourced code from the Latent Diffusion Model [36]. We use the 'Nerfacto' model from NeRFStudio [39] as our underlying backbone and adapt the implementation of diffusion priors training from Nerfbusters [42]. $\lambda_{\text{geom}} = 0.1$ and $\lambda_{\text{in}} = 0.1$. During training, we train for 1500 iterations without $\mathcal{L}_{\text{geom}}$ for initialization, and another 2500 iterations sampling 30 cubes per iteration.

H.4 Implementation details on SPN-360

We proposed a baseline SPN-360 that is stronger than directly adapting SPIn-NeRF [26] on 360° data, as its released implementation does not support 360-degree NeRF. The implementation of SPN-360 is based on NeRFacto. We obtain masks for SPN-360 through initialization from Dino and Semantic NeRF refinement, and inpaint with LaMa [38] on both RGB and depth map. Its segmentation masks and inpainting results are both generated by the method used in SPIn-NeRF so that we can compare them with our In-NeRF360 results

I. Additional results on segmentation.

Fig. 17 shows additional segmentation results on selected datasets with our segmentation module.

J. Acknowledgements

The authors thank Michele Vidulis and Desmond (Zhenyuan) Liu for their time spent proofreading and kind suggestions during the paper writing. The authors also thank the generous and insightful comments from all the reviewers, without whom this work will not be able to come to its current shape.