A. Summary

We provide additional details about the comparative baselines, against which we benchmark, in Appendix B. Further exposition about our disparity smoothing technique (see § 4.1) and edge island filtering method (see § 4.2.3) is given in Appendix C and Appendix D, respectively. Additional visualizations are shown in Appendix E, including methodological illustrations (§ E.1) and qualitative examples (§ E.2 and § E.3). Technical implementation details, such as hyper-parameter values, are discussed in Appendix F. Finally, further explanation about our choice of evaluation metrics is given in Appendix G. Please also view our supplementary website for additional visualizations, including videos.

B. Baseline Details

B.1. Masked-NeRF + DreamFusion

For the *Masked-NeRF + DreamFusion* baseline, we use the same per-scene text prompts we used to generate our reference views, to guide the generation of the masked region using the score distillation sampling (SDS) [49] loss. We found that gradually and uniformly decreasing the maximum noise steps, t_{max}, during fitting, until it equals the minimum noise steps, t_{min}, at the last iteration, improves quality. We suggest this is because, at first, higher noise levels are effective in the generation of global scene structure, and later, lower noise-levels enable fixing details. Due to the unavailability of DreamFusion's code and their underlying diffusion model, Imagen [57], we used stable-dreamfusion [63], with Stable-Diffusion [55] as the underlying diffusion model.

B.2. NeRF-In

As in prior work [42], we used our own implementation of NeRF-In [34], due to the unavailability of official code. Besides the primary distinctions with our method, such as the pixelwise loss, the remaining architecture (e.g., the use of NGP [44]) is identical to our method. Note that this induces minor implementation differences from the concurrent technical report of NeRF-In, such as the choice of pretrained 2D inpainting model.

Since NeRF-In considers the effect of varying numbers of reference images, we considered two variants of NeRF-In: using multiple reference images (i.e., inpainting all images, as in SPIn-NeRF [42] and using a single one. By default, we utilize the latter method, as it obtains better overall performance (in both our experiments and those of NeRF-In itself), but report the performance of both models in Table 1.

B.3. Object-NeRF

Following the Object-NeRF [78] model, we can remove objects by simply ignoring the contribution of masked 3D points in the volume rendering process (equivalent to setting $\sigma_i = 0$ in masked regions). This is possible here due to the assumed availability of a 3D mask. Note that we are only utilizing this particular approach to object removal, not the entire Object-NeRF algorithm (i.e., the construction of the NeRF itself is identical to our method).

C. Disparity Smoothing Details

After performing the initial depth alignment (as discussed in § 4.1), we further reduce the misalignments around the edges of the reference mask, M_r, via smoothing the aligned reference disparity, D_r. More specifically, to improve the visual continuity of the reference-view boundary between the aligned masked disparity, $D_r \odot M_r$, and the unmasked rendered NeRF disparity, $\widehat{D}_r \odot (1 - M_r)$, we smooth D_r to get the edge-smoothed disparity, D_r^{smooth}:

$$D_r^{\text{smooth}} = D_r + D^{\text{correction}}, \tag{11}$$

where $D^{\text{correction}}$ is the smoothed disparity correction obtained by minimizing the following objective:

$$\begin{aligned} &\left\| (\widehat{D}_r - D_r^{\text{smooth}}) \odot (1 - M_r) \right\|_2^2 \\ &+\gamma_{\text{smooth}} \sum_{p \in I_r} \sum_{p' \in \mathcal{N}(p)} \left(D^{\text{correction}}(p) - D^{\text{correction}}(p') \right)^2, \end{aligned} \tag{12}$$

where for a pixel, p, $\mathcal{N}(p)$ is the set of four neighbouring pixels, and γ_{smooth} is the weight of the smoothness loss. The first term in Eq. 12 fits the unmasked pixels of $D^{\text{correction}}$ to the difference of the rendered disparity, $\widehat{D}_r$, and the aligned disparity, D_r. The second term is the smoothness penalty, to smoothly propagate the values of $D^{\text{correction}}$ from outside the mask to inside.

D. Edge Island Filtering Details

When propagating appearance information into the masked area, in order to construct view-dependent effects for supervision in non-reference views, recall that the bilateral solver is sometimes unable to provide sensible colour values in some areas of the masked region, due to the presence of "edge islands" (see § 4.2.3). Such areas are isolated patches in bilateral space, for which the bilateral solver cannot effectively produce colour values (see Fig. 11 for instances of this). In this section, we provide additional details on our filtering algorithm for removing these invalid values, so that they are not used for supervision.

First, we dilate the mask, M_r, with kernel size 5 to get the dilated mask, M_r^{dilated}. Then, for each target view, t, we find the maximum absolute value of the residual inside M_r^{dilated} and outside M_r:

$$\text{res}_t^{\max} = \max \left(\text{abs}(\text{res}_t) \odot \left(M_r^{\text{dilated}} \cap (1 - M_r) \right) \right), \tag{13}$$

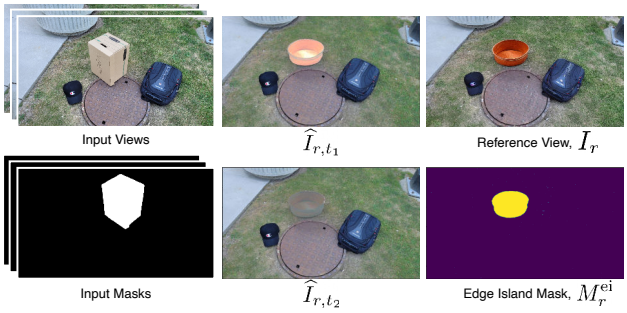

Figure 11: Examples of our "edge island" detection method, designed to filter out erroneous outputs from the bilateral filter (detailed in Appendix D). Left column: input views and masks for the scene. Middle column: view-substituted renders after bilateral inpainting (see also § 4.2), which has produced poor quality colours in the edge island formed by the washtub. Right column: (top) the reference view and (bottom) the detected mask, used to filter out rays that would potentially damage the output.

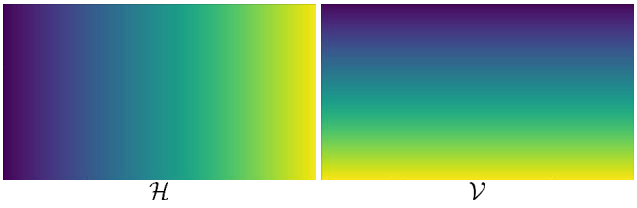

Figure 12: The additional matrices used for tighter alignment around the edges when aligning disparities (see § 4.1). In our experiments, scale and offset were insufficient to have the depths completely aligned around the boundaries of the mask. These two matrices allow the predicted depth to be tilted along the x and y axes.

where abs$(\cdot)$ is the element-wise absolute value. We denote the mask for the pixels in $res_t \odot M_r$ with absolute values higher than $\text{res}_t^{\max} \times c_{\text{ei}}$ as $M_{r,t}^{ei}$, where $c_{\text{ei}} \geq 1$ is the filtering threshold. The mask of the edge island is then obtained as the union of the mask of all of the out-of-distribution values among all of the target views:

$$M_r^{\text{ei}} = \bigcup_t M_{r,t}^{ei}. \tag{14}$$

Fig. 11 shows an example of the effects of an edge island inside the masked region (the orange pan) on the target colours of two example target views, $\widehat{I}_{r,t_1}$ and $\widehat{I}_{r,t_2}$. As shown in the figure, the bilateral solver has failed to predict correct view-dependent colours for the pan, resulting in extreme behaviour inside the pan. Our proposed edge island filtering successfully detects and removes the outlier values via the edge island mask, M_r^{ei}.

[3]IBRNet images in Fig. 13,14 by Wang et al. available in IBRNet [72] under a CC BY 3.0 License.

E. Additional Visualizations

E.1. Methodological Illustrations

Depth Alignment Tilt Matrices. In Fig. 12, we visualize the matrices utilized for tighter depth alignment (see § 4.1). These matrices allow the optimization to *tilt* the depths, in addition to scaling and shifting them.

Overview. We provide an expanded methodological illustration in Fig. 13, covering our approach to providing geometric supervision in the masked region (§ 4.1) and handling the construction of view-dependent effects in non-reference views (§ 4.2); see also Figs. 3, 4, and 5.

View-Substituted Images. We also provide some examples of view-substituted images (see § 4.2.1) in Fig. 14. Notice that the view-substituted images have identical camera viewpoint (and thus image structure) as the reference image, but different colours, corresponding to the view-dependent visual differences across the non-reference images.

E.2. Additional Ablation Examples

Masked Depth and Disocclusion. We show an additional experimental ablation example in Fig. 15, removing masked depth supervision and disocclusion handling (as in Fig. 8). Removing the former causes significantly damaged geometry (and thus considerable visual artifacts as well), while ablating the latter increases blurriness in the disoccluded region (i.e., around newly unveiled details near the occlusion boundary).

Disparity Smoothing. In Fig. 16, we consider the effect of ablating our disparity smoothing approach (see § 4.1 and Appendix C), utilized for obtaining depth in the masked area and matching it to the surrounding scene geometry. Particularly close to the mask boundary, we see that the *un*smoothed geometry has a much more jarring transition between the masked and unmasked areas.

E.3. Qualitative Results

Comparisons. Additional comparisons to SPIn-NeRF, NeRF-In, and DreamFusion are shown for novel view synthesis in Fig. 17. Notice that utilizing the DreamFusion [49] loss along with the Masked-NeRF (see § 5 and Appendix B) can result in unrealistic colours (first row) and sometimes a failure to converge (second row), though the quality improves over Masked-NeRF alone (see Table 1). NeRF-In [34] is blurry in masked areas, as the textures do not match well in a pixelwise manner. SPIn-NeRF [42] reduces this blurriness considerably, but still incurs some level of blur, especially in the presence of more complex textures (e.g., second row). In contrast, our method provides sharp details for all cases.

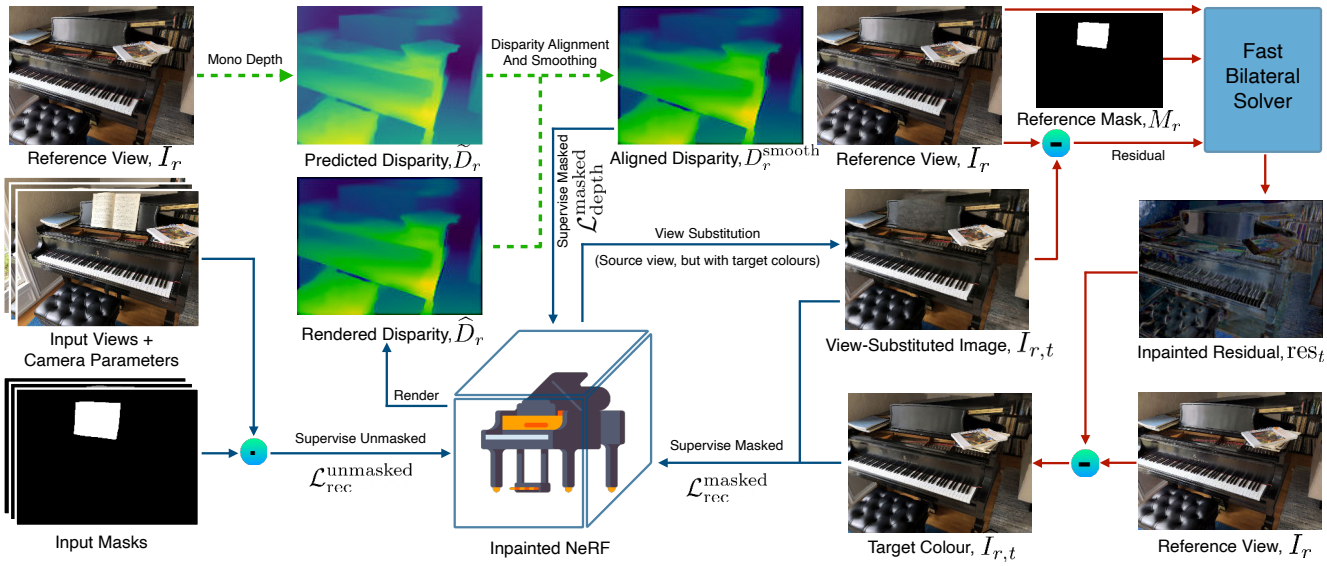

Figure 13: Schematic overview of our NeRF fitting algorithm for 3D inpainting. The inputs to the method are a single inpainted reference view, I_r, and a set of posed images with associated inpainting masks (leftmost column). We begin the fitting process with standard NeRF supervision on the *un*masked areas of the images, after which we can render a disparity map, $\widehat{D}_r$, with reasonable quality outside the mask (lower-left insets). We then use a monocular depth estimator to obtain the predicted disparity, $\widetilde{D}_r$, and apply a novel alignment procedure (§ 4.1) to obtain an aligned disparity map, D^{smooth_r}, which can be used to supervise the depth *under* the mask via loss $\mathcal{L}_{\text{depth}}^{\text{masked}}$ (upper middle inset). Finally, to obtain view-dependent effects in unseen views (§ 4.2), we utilize our new *view-substitution* technique (§ 4.2.1) to render an image, $I_{r,t}$, via the reference camera, but with the colours of a non-reference (target) view, I_t (centre-right inset). The view-substituted image, $I_{r,t}$, is subtracted from the reference view, I_r, to obtain a residual image, $\Delta_t = I_r - I_{r,t}$; we then apply the bilateral solver, $\mathcal{B}$, to refine Δ_t, using the reference mask, M_r, to construct a confidence map (low inside the mask and high outside it), guided by the bilateral affinities of I_r (upper-right insets; see § 4.2.2). This has the effect of "diffusing" the view-dependent effects of the non-reference view from *outside the mask* into the inside of the masked area, obtaining an "inpainted" residual, res_t. Subtracting this from I_r gives our desired colours, $\widehat{I}_{r,t} = I_r - \text{res}_t$, which can be used to supervise the colours *under* the mask (lower-right insets). The resulting combined losses thus supervise the NeRF from non-reference target viewpoints both outside the mask ($\mathcal{L}_{\text{rec}}^{\text{unmasked}}$) and inside the mask ($\mathcal{L}_{\text{depth}}^{\text{masked}}$ and $\mathcal{L}_{\text{rec}}^{\text{masked}}$). See § 4 for details.[3]

Larger Camera Motion. We further provide 3D inpainting results on scenes with larger camera motions in Fig 18. Our model produces view-consistent outputs. For this experiment, we used scenes from IBRNet dataset [72].

Controllability. We also provide more examples of controllable inpainting in Fig. 19. Notice that we can easily control various aspects of the inpainted scene, such as the presence or absence of roots in the tree (upper rows) or the length of the stone bench (lower rows), by simply changing the inpainting of the single reference image. For additional examples of controllable insertion, see also Fig. 10.

F. Implementation Details

In our experiments, both N_{depth} and $N_{\text{bilateral}}$ are set to 2000. We train each scene for 10000 iterations. The disocclusion handling is run every $N_{\text{do}} = 3000$ iterations. The weights $\gamma_{\text{depth}}^{\text{masked}}$, $\gamma_{\text{rec}}^{\text{masked}}$, γ_{do}, η_{do}, and γ_{smooth} are set to 4, 2, 1, 0.25, and 1000, respectively, and c_{ei} is set to 2. We follow [42] and use a combination of [44] and [11] for faster convergence, and dilate all of the masks for 5 iterations with a 5×5 kernel to make sure that the masks cover the whole object, and to mask some of the shadows of the unwanted object. All of the images are downsized four times to reduce memory usage and match the experiments of SPIn-NeRF [42]. We also use the distortion loss proposed by [3] for reducing the floater artifacts. We set the weight of the distortion loss to 0.01. For generating multiple inpainted source views, we leverage the diversity of denoising diffusion models, and use stable-diffusion inpainting v2 [55]. For inpainting the residuals with the bilateral solver, we set the brightness and colour bandwidths to 4, while the spatial bandwidth was set to 128. The strength smoothness and the number of PCG iterations are set to 128 and 25, respectively. For disocclusion handling, we use LaMa [61] as the 2D inpainter and use three target images for T (cor-

Figure 14: Overview of the outputs of our view-substitution method. The input views and masks (top-left) with their corresponding camera parameters, in addition to a single reference view (bottom-left), are the inputs to our multiview inpainting approach. On the right hand side, we show the view-substituted renderings, $\{I_{r,t_1}, \cdots, I_{r,t_4}\}$, for four different target views, $\{t_1, \cdots, t_4\}$. For each view-substituted image, $I_{r,t}$, we also provide the absolute value of the residual, $|I_r - I_{r,t}|$, to illustrate the view-dependent effects provided by our approach. Notice that all of the view-substituted images are looking at the scene from the reference camera, but the rendered colours are from different target cameras.

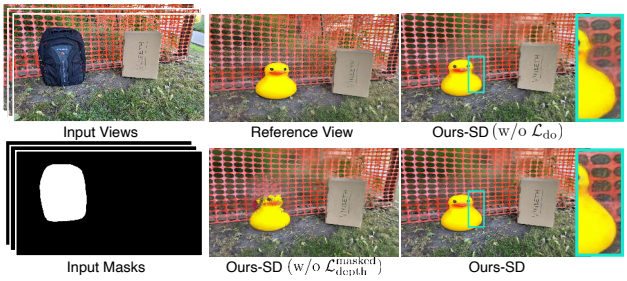

Figure 15: Qualitative example of effects of ablation (see also Fig. 8). Notice the degradation incurred by not using the masked depth supervision (lower-middle inset) and the slightly blurrier outputs in the disoccluded region when not using $\mathcal{L}_{\text{do}}$ (upper-right inset; look closely at the zoomed area, particularly at the background close to the edge of the inserted duck).

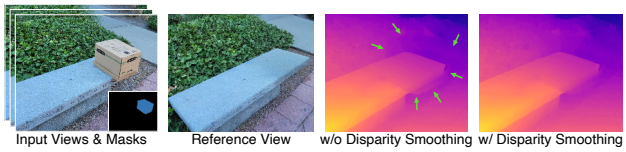

Figure 16: Effect of our disparity smoothing step (see § 4.1 and Appendix C) on the rendered disparities. As illustrated above, the edges of the masked region (around the box) are more blended in with the surrounding after adding the disparity smoothing component.

responding to the cameras furthest leftward, rightward, and upward). A small morphological dilation (four iterations with a 3×3 kernel) is applied to remove noise from the disocclusion masks. The bilateral filter in the disocclusion case uses a spatial bandwidth of only 8. Our implementation is mainly in PyTorch [46]. For generating the inpaintings for *Ours-SD*, we used stable diffusion inpainting v2 [55], and a simple per-scene text prompt describing the inpainted scene. Below are the text prompts used for SPIn-NeRF scenes:

- A stone bench, a bush in the background, the bench is grey with a rectangular shape in perspective, photorealistic 8k
- A wooden tree trunk on dirt, photorealistic 8k
- A red fence, photorealistic 8k
- Stone stairs, photorealistic 8k
- A circular lid made of rusty iron on a grass ground, photorealistic 8k
- A corner of a brick wall, photorealistic 8k
- A wooden bench in front of a white fence, photorealistic 8k
- An image of nature with grass, bushes in the background, photorealistic 8k

Figure 17: Additional qualitative comparisons to baselines with synthesized novel views. The Masked-NeRF+DreamFusion model (second column) does improve quantitatively (see Table 1) over using Masked-NeRF alone or simply removing the object in 3D without inpainting (the "Object-NeRF" baseline), but it does not output sufficiently realistic details to outperform our method: see the oversaturated colours on the fence in first row and the unnatural output in the second row. NeRF-In [34] (third column), here using the "multiple" variant with LaMa [61], is quite blurry, due to disagreements between inpainting details among the input images. SPIn-NeRF [42] (fourth row) improves on this via the use of a perceptual loss [84], but still generates blurry details when significant disagreement among inpaintings are present (semantic differences, as such the presence or absence of the pipe in the second row, and complex textures (e.g., the grassy dirt in row one or the variously coloured bricks in row two) can exacerbate this problem). In contrast, our method is consistently sharp; see also Fig. 7.

Figure 18: Additional results of our model on scenes with larger camera movements.

- A desk in front of a brick wall with an iron pipe, photorealistic 8k
- A brick wall, photorealistic 8k

Note that we did not engineer the prompts to improve the results. We typically selected the first generated inpainting. However, as seen in Fig. 2, sometimes, the stable diffusion

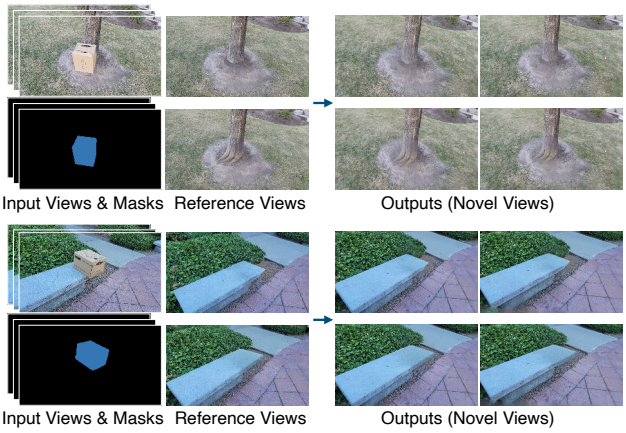

Figure 19: Qualitative illustration of our results on additional scenes from the SPIn-NeRF dataset [42]; see also Fig. 10. For each scene, we use two different reference views to generate corresponding inpainted scenes. For each inpainted scene, we show two novel view renderings. Note the ability to insert novel content into the 3D scene or modify existing scene structure, such as adding the tree roots and controlling the length of the bench. Please see our supplementary website for additional visualizations.

inpainter adds objects in the scene; in those cases, we regenerated the output to get an inpainting without any additional object for a fair comparison to the baselines. For quantitative experiments, we always select the 30-th image among the 60 training views in SPIn-NeRF's dataset [42] as the reference view.

G. Metrics: Additional Details

The ill-posed nature of inpainting means that evaluation is non-trivial: "ground-truth" images are merely one of an infinite number of possible solutions, any plausible member of which should be considered valid. We therefore focus on evaluating perceptual quality and realism, rather than reconstruction, via two types of metrics: full-reference (FR) and no-reference (NR).

In the FR case, we utilize the ground-truth (GT) images of the scene without the object for comparison with LPIPS [84] and Frechet Inception Distance (FID) [16]. LPIPS, a perceptual distance, is far more robust to changes that maintain textural consistency than pixelwise distances. For FID, we compare the distributions of encoded statistics between the inpainted and GT images, which confers high robustness to mismatches in local details, focusing instead on agreement in high-level visual appearance. Both of these metrics were used previously for 3D inpainting evaluation [42]. For both LPIPS and FID, we only perform the comparison inside the bounding boxes of the objects. We expand the bounding boxes by 10% to match SPIn-NeRF's [42] setup.

However, FR metrics are not completely robust to the choice of reference image, preferring solutions more similar to the GT over others that are equally perceptually realistic. This is exacerbated if an inpainting model inserts new semantic content into a scene, as recent diffusion-based approaches are apt to do (e.g., [55, 50]), whether it is perceptually realistic or not. Thus, we consider two NR metrics, where image quality is assessed in a stand-alone manner. The first measure is simply the variance of the image Laplacian, a classical measure of sharpness (e.g., [48]), which has been previously used to evaluate 2D generative image models [66, 17]. The second is MUSIQ [25, 7], a transformer-based model for NR image quality assessment, meant to reproduce human perceptual judgments.

Note that our metrics in the FR case are computed against bounding boxes (containing the object mask) in *held-out* views, while our NR sharpness metrics are computed across 120 renders from a camera in a spiralling pattern (in video form). In this way, we assess inpainting quality in its full 3D context; i.e., we ensure that the inpainting quality generalizes to novel views.