Supplementary Material
Inpaint360GS: Efficient Object-Aware 3D Inpainting via Gaussian Splatting for 360° Scenes

Shaoxiang Wang[1,2] Shihong Zhang[3] Christen Millerdurai[1]
Rüdiger Westermann[3] Didier Stricker[1,2] Alain Pagani[1]

[1]German Research Center for Artificial Intelligence [2]RPTU [3]Technical University of Munich

Abstract

In the supplemental material, we provide additional details about the following:

- *Dataset Details. (Section 1)*
- *Implementation Details. (Section 2)*
- *Additional Ablation Study and Experiment Analysis. (Section 3)*
- *Per-Scene Breakdown of the Results. (Section 4)*

1. Dataset Details

We provide a comprehensive analysis of datasets employed in our study, highlighting the limitations of existing datasets and motivating the introduction of a novel dataset specifically designed for 3D 360° inpainting evaluation.

Mip-NeRF 360 [1]. This dataset comprises professionally captured 360° imagery obtained with high-end cameras. It features exceptional image quality, carefully curated scenes, and precisely calibrated camera parameters. However, Mip-NeRF 360 lacks ground truth for after object removal scenarios, thereby precluding its use for quantitative assessment of 3D inpainting performance.

Instruction-NeRF2NeRF [4]. This dataset provides complete 360° views and encompasses a wide variety of scenes. Nonetheless, similar to Mip-NeRF 360, it does not include ground truth for post-removal conditions. In addition, its relatively lower image quality and limited resolution, while sufficient for current methodologies, may not meet the demands of future advances in 3D inpainting.

AuraFusion 360 [20]. This dataset includes only a single object per scene and lacks challenging multi-object, complex environments. Our dataset addresses this limitation by incorporating scenes with multiple occluded objects. In Gaussian-based inpainting, where accurate point cloud initialization is critical, we ensure fair evaluation by excluding any inpainting-view-specific points. In contrast, AuraFusion360 suffers from data leakage, as its sparse point cloud includes points visible only in inpainting views. Moreover, frames extracted from video often lack sufficient quality.

IMFine [16]. Similar to AuraFusion 360, the IMFine dataset also suffers from data leakage from test set. In addition, it does not provide masks for regions that become visible only after object removal. This lack of ground-truth masking makes it impossible to distinguish between masked and background areas, thereby preventing meaningful quantitative evaluation such as FID calculation on the inpainted regions. The frames are extracted from the video as well.

SPIn-NeRF [12]. SPIn-NeRF provides ground truth for inpainting following object removal, addressing a crucial limitation of Mip-NeRF360. However, its scope is restricted to front-facing views, limiting its applicability to full 360° inpainting tasks. Additionally, the dataset primarily consists of small, enclosed environments, thereby constraining its utility to a narrow range of inpainting applications. Furthermore, inconsistent camera parameters (such as ISO, exposure, and white balance) between the original and post-removal captures introduce unintended variations in scene appearance. This discrepancy compromises the reliability of the ground-truth data, rendering quantitative evaluation meaningless, as the observed differences may stem from photometric inconsistencies rather than actual inpainting errors.

Our Dataset. To overcome the aforementioned limitations, we introduce a new high-quality dataset specifically designed for 360° inpainting with quantitative evaluation. This dataset is acquired using diverse imaging devices across scenes of varying scales and incorporates multiple difficulty levels within the same scene to better accommodate future developments in 3D inpainting.

To ensure diversity in scene scales and realistic application scenarios, we employ a combination of DSLR cameras, and drones for data collection. Large-scale

	InNeRF360	Mip-NeRF 360	Instruct-NeRF2NeRF	AuraFusion 360	IMFine	SPIn-NeRF	Ours
Device	--	Sony NEX C-3 & Fujifilm X100V	Smartphone & Mirrorless Camera	--	DJI Pocket 3	Samsung Galaxy S20 FE	Canon EOS 5D
Example Data	--						
Camera Views	360°	360°	360°	360°	360°	180°	360°
Image Size	--	3115 × 2078	985 × 729	960 × 540	1920 × 1080	4032 × 2268	2950 × 1909
Inpainting GT after object removal	--	✗	✗	✓	✓	✓	✓
No data leakage for inpainting area	--	--	--	✗	✗	✗	✓
Mask of unseen after object removal	--	✗	✗	✓	✗	✓	✓
Multi-scale Scenarios	--	✓	✓	✗	✗	✗	✓
Fixed Camera Settings	--	✓	--	Image extracted from video	Image extracted from video	✗	✓
Varying Complextity	--	✓	✗	✗	✓	✗	✓

Figure 1. **Dataset Comparison.** We compare our new dataset with existing datasets commonly used for inpainting tasks, including unpublished InNeRF360 [18], Mip-NeRF 360 [1], AuraFusion 360 [20], IMFine [16], SPIn-NeRF [12], and Instruction-NeRF2NeRF [4]. Our dataset is designed for well-structured 360° inpainting scenarios, including challenging multiple occluded objects, no data leakage in the inpainting regions of the point cloud, and consistent camera settings within each scene.

outdoor scenes are captured using a DJI Mini 2 drone, which is equipped with a 24 mm f/2.8 lens with a fixed focus range. For smaller outdoor scenes, we utilize a Canon 5D with an 24-105 mm zoom lens, fixed at its widest focal length (24 mm). This choice minimizes focal length variations, thereby reducing geometric distortions, perspective inconsistencies, and optical aberrations, facilitating subsequent processing.

For each scene, we manually configure white balance, ISO, shutter speed, aperture, and focus based on a reference image, and keep these settings fixed throughout the capture process to ensure photometric consistency across frames. For indoor scenes, we utilize large diffuse light sources and LED spotlights to mitigate strong cast shadows. In outdoor environments, we capture scenes under overcast conditions. Overcast conditions produce soft shadows that minimally affect scene illumination.

Each scene consists of 100-200 images, during which target objects are manually moved to facilitate dynamic scene acquisition. The dataset consists of two main parts. The first part includes all objects in the scene. The second part serves as the ground truth for inpainting, where targeted objects are removed to introduce novel viewpoints, enabling quantitative evaluation of inpainting performance. To ensure that both the training and test inpainting datasets share a consistent coordinate system, we process them jointly using the publicly available COLMAP [15] software to obtain camera poses and a sparse point cloud. Within each scene, cameras share a single set of intrinsic parameters, and we adopt a pinhole camera model for undistortion. Importantly, to prevent data leakage, we remove point cloud regions corresponding to the test-time occluded region, a crucial step that has often been overlooked in prior works [16, 20].

Regarding the mask of the object, we use SAM [13] method and our proposed mask association to connect with each other to get unified object ID. With the selected object ID, we can get the object mask per image. In addition, in order to evaluate the unseen area after object removal and background respectively. We also prepared the mask of the unseen region after object removal for this dataset.

2. Implementation Details

Gaussian Field Initialization. We initialize our scene using the default settings from the original 3D Gaussian Splatting framework. Notably, we operate in evaluation mode, where only $7/8$ of the training data is used for training, while the remaining $1/8$ interval-sampled data is reserved for evaluation.

Mask Association. To obtain raw 2D segmentation masks, we employ the 2D segmentation foundation model HQSAM [13]. The model is used with their default parameter configurations. During the association stage, we set a predefined GS-IoU threshold $\sigma = 0.2$ for matching objects in the Key Object Database. To improve association accuracy per view, each image is divided into 16×16 patches, and mask matching is performed at the patch level. The maximum number of object categories allowed in the classification process is 256.

Object Feature Distillation. To distill object features from the 2D associated object masks into the 3D Gaussian Field, we randomly initialize each Gaussian with a 16-dimensional feature vector f_i to represent its identity. For neighbor aggregation, we apply a k-nearest neighbor (KNN) strategy with $k = 5$. Additionally, a linear transformation $\Phi(\cdot)$ projects the feature dimension to Q, where Q represents the quantity of object categories obtained during mask association, with a maximum of 256. In the overall loss function, we set the weighting factor $\lambda = 0.0005$. The optimization process is conducted over 2000 iteration steps.

Virtual Camera Views. For the virtual camera views $\mathcal{V} = (I_j, D_j, M_j)_{j=1}^{L}$, we utilize 90% of the known training camera poses and the object center in world coordinates to initialize the virtual camera centers. These centers are distributed along a circular trajectory whose radius is adaptively determined based on the area of the NBS region mask. Notably, a smaller camera path radius brings the virtual camera closer to the object, which typically results in a larger NBS region mask. A too-large inpainting region may lead to failure cases for the 2D inpainter. Specifically, we empirically the mask area to lie within 1% to 50% of the full image area to ensure effective inpainting.

Object Removal and 2D Inpainting. For object removal, we leverage SAM-Tracking [3] to enable both prompt-based and click-based interactive segmentation. Once an object is identified, all Gaussian points corresponding to the object, including those computed using the Delaunay convex hull, are removed from the scene.

During the 2D inpainting stage, the input is the rendered scene where removed objects create empty regions. We use SAM-Tracking to generate the corresponding inpainting masks. They are from virtual camera views $\mathcal{V}$. The rendered image after removal object, corresponding mask and last inpainted image are fed into the LaMa inpainting model [17] to reconstruct missing regions. The encoder and decoder of LaMa are frozen, while latent representation (ℓ_t, ℓ_{t+1}) is trainable here. A similar approach is applied for depth inpainting, ensuring structural consistency across views.

During the 2D inpainting stage, the input consists of rendered images with missing regions caused by object removal. Inpainting masks are generated using SAM-Tracking. The masked image, corresponding inpainting mask, and the previously inpainted image are fed into the LaMa inpainting model [17] to reconstruct the missing content. While the encoder and decoder of LaMa are frozen, the latent representations (ℓ_t, ℓ_{t+1}) extracted from rendered images remain trainable. Optimization steps we set 10 here. A similar procedure is applied for depth inpainting to ensure structural consistency across views. The above inpainting process is executed on virtual camera views $\mathcal{V}$.

3D Inpainting. We initialize the NBS region of the Gaussian field using depth-color fusion from the first inpainted color and depth images of the virtual camera view. During the 3D inpainting stage, we set the loss weights to $\lambda_1 = 0.2$ and $\lambda_2 = 0.005$, and perform optimization for 2000 iterations.

Algorithm 1 *Inpaint360GS*

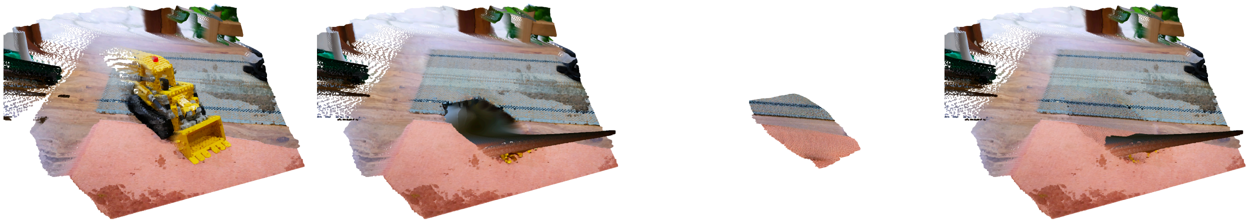
```
RGB images                                                         ▷ Input
$p \leftarrow$ SfM Points                     ▷ Sparse point position and camera pose in 3D
$p, s, \alpha, c \leftarrow$ OptimizedAttributes()        ▷ Position, covariances, opacities, colors through 3DGS [6]
$m = (m_1, m_2, \dots, m_K) \leftarrow$ Zero-shot 2D Segmentation  ▷ SAM's masks at Various $K$ Views
$(O_1, O_2, \dots, O_K) \leftarrow$ Mask association through Key Object Management  ▷ Multi-view consistent associated masks in 3D
$f \leftarrow$ identity vector                ▷ Initialize identity vector for each Gaussian
$(p, s, \alpha, c, f) \leftarrow$ FreezeParam()   ▷ Freeze all parameters except identity vector $f$
while not converged do
    $V, C, O \leftarrow$ SampleTrainingView()            ▷ Camera view, image and mask
    $\hat{C}, \hat{D}, \hat{O} \leftarrow$ Rasterize($p, s, \alpha, c, f, V$)   ▷ Rendered image, rendered depth and identity mask
    $\mathcal{L}_{Dis} \leftarrow \mathcal{L}_{\text{obj}}(O, \hat{O}) + \lambda \mathcal{L}_{\text{space}}(f, f^1, f^2, \dots, f^k)$   ▷ Distillation Loss function
    $f \leftarrow$ Adam($\nabla \mathcal{L}_{Dis}$)                      ▷ Backprop & Step
end while
$\mathcal{V} = \{(C_i, D_i, M_i)\}_{i=1}^{L}$             ▷ Virtual camera view after object removal
$C_{inp}, D_{inp} \leftarrow$ ConditionLaMa($\mathcal{V}$)          ▷ Inpainted color and depth
$\mathcal{R}_{inp} \leftarrow C_{inp}, D_{inp}$        ▷ Initialize Gaussian field $\mathcal{R}_{inp}$ for NBS region
while not converged do
    $\mathcal{L}_{\text{3DInp}} \leftarrow (1 - \lambda_1)\mathcal{L}_1(C_{\text{inp}}, \hat{C}, M) + \lambda_1 \mathcal{L}_{\text{D-SSIM}}(C_{\text{inp}}, \hat{C})$
    $+\lambda_2 \mathcal{L}_{\text{LPIPS}}(C_{\text{inp}}, \hat{C}, M)$            ▷ 3D inpainting loss function
    $\mathcal{R}_{inp} \leftarrow$ Adam($\nabla \mathcal{L}_{3DInp}$)                ▷ Backprop & Step
end while
```

2.1. Proof of the Validity of Depth Definition

A typical neural point-based approach (*e.g.*, [9]) computes the color C of a pixel by blending $\mathcal{N}$ ordered points overlapping the pixel:

$$C = \sum_{i \in \mathcal{N}} c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j) = \sum_{i \in \mathcal{N}} c_i \alpha_i T_i = \sum_{i \in \mathcal{N}} c_i w_i, \quad (1)$$

where $\mathbf{c}_i$ is the color of each point and α_i is given by evaluating a 2D Gaussian with covariance Σ [22] multiplied with a learned per-point opacity. $T_i = \prod_{j=1}^{i-1}(1 - \alpha_j)$ is transmittance after passing i gaussian point.



(a) 3D Gaussian Field

(b) 3D Gaussian Field after Object Removal

(c) Inpainted Mask Area

(d) Visualization of (b) + (c)

Figure 2. **Validity of Depth Definition.** While (a) and (b) represent the point clouds generated from the Gaussian field under the given camera pose before and after object removal, respectively, (c) is constructed via color-depth fusion between the inpainted image and the depth defined in Eq. (2). The point cloud in (c) can be effectively used as initialization for the 3D inpainting stage.

Similarly, depth is defined as

$$D = \sum_{i \in \mathcal{N}} z_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j) = \sum_{i \in \mathcal{N}} z_i w_i \qquad (2)$$

where z_i is z-coordinate in the camera coordinate system.

Our goal is to prove the weight along a current sampling ray r as:

$$w_i = 1 - T_i = w_{i-1} + T_{i-1}\alpha_i$$

$$\begin{aligned} w_i &= w_{i-1} + T_{i-1}\alpha_i \\ &= w_{i-2} + T_{i-2}\alpha_{i-1} + T_{i-1}\alpha_i \\ &\dots \\ &= T_0\alpha_1 + T_0\alpha_1 + \dots + T_{n-1}\alpha_n \\ &= (T_0 - T_1) + (T_1 - T_2) + \dots + (T_{n-1} - T_n) \\ &= T_0 - T_n = 1 - T_n \end{aligned}$$

To validate the effectiveness of our depth definition, we present visualizations in Fig. 2. Subfigures (a) and (b) show point clouds rendered from the 3D Gaussian field before and after object removal, respectively. In (c), we visualize the fused point cloud generated by combining the inpainted RGB image and the estimated depth defined in Eq. (2). Notably, unlike (a) and (b), which are derived directly from the Gaussian field, (c) is obtained through depth-color fusion. When using (c) as the initialization for the 3D inpainting stage on (b), the resulting reconstruction (d) demonstrates strong geometric consistency, validating our initialization strategy. This approach avoids the depth alignment issues present in [10, 19, 20].

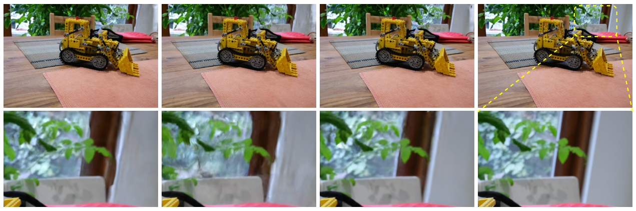
3. Additional Ablation Study and Experiment Analysis

Detailed Time Analysis of pipeline: We report the runtime breakdown of different stages in our pipeline for the `bear` and `kitchen` scenes, corresponding to Tab. 2 in the main paper. The "Pure 3DGS" time refers to the training time required to learn the Gaussian field without any editing components. Adding the time for Mask Association and Distillation yields the total time for the "Vanilla Gaussian" baseline in Tab. 2. The "Inpainting" time includes both 2D and 3D inpainting steps.

Tab. 3	Pure 3DGS	Mask Association	Distillation	Inpainting	Total
`bear`	17 mins	2.5 mins	2 mins	2.5 mins	24 mins
`kitchen`	8 mins	3 mins	1 mins	3 mins	15 mins

Table 1. **Detailed Runtime and Model Size Comparison.**

Analysis of the Effectiveness of Consistent Object ID Mask on Rendering. Compare our two-stage method with the one-stage semantic Gaussian method, GauGroup [21]. Our approach achieves superior global consistency, not only for foreground objects but also for the background. Figure 3 compares the rendering quality of our method with that of GauGroup [21]. Our approach achieves superior rendering quality due to more consistent multi-view segmentation masks and a training strategy that independently optimizes the 3D Gaussian Splatting (3DGS) and the integration of semantic masks. Due to the incorporation of object masks, the background geometry is further refined compared to vanilla 3DGS [6].



3DGS, PSNR=31.46 GauGroup, PSNR=30.72 Ours, PSNR=31.34 GT Test-Set

Figure 3. **RGB Rendering Comparison.** While GauGroup[21] sacrifices rendering quality in color fidelity to incorporate object IDs, our method achieves comparable PSNR[dB ↑] to the naive 3DGS[6] method. Please zoom in for details.

Analysis of Object Removal Accuracy. In Fig. 4, we compare the performance of our method in target object removal. The results demonstrate that our approach achieves more precise object removal and produces a more accurate inpainting-ready base. This indicates that our method can more effectively assign consistent spatial Gaussian representations, leading to better convergence without misclassified surrounding artifacts. To quantitatively assess the accuracy of object mask identification, we introduce the Average Mask Coverage Ratio (AMCR), defined as:

$$\text{AMCR} = \frac{1}{N}\sum_{k=1}^{N}\left(\frac{\|M_k\|_1}{H \times W} \times 100\%\right) \tag{3}$$

It quantifies the proportion of empty regions in the image after object removal, averaged over the N training images. For each image, the binary mask $M_k \in [0,1]^{H\times W}$ indicates pixels to be inpainted, with 1 denoting removed regions. A lower AMCR value implies more accurate object segmentation and less redundant inpainting area, which typically leads to better reconstruction performance.

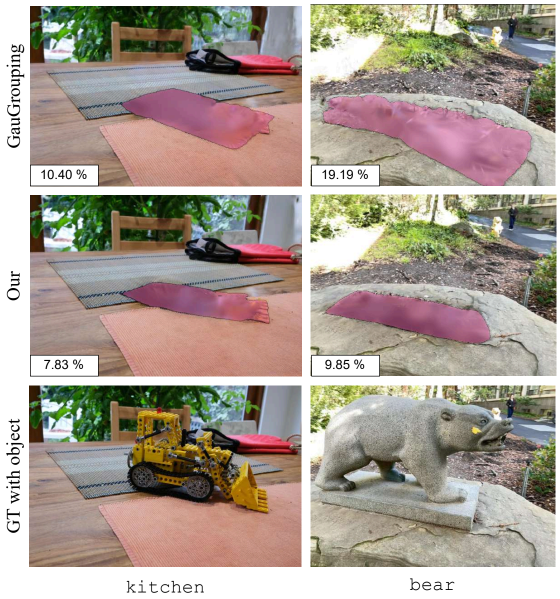


Figure 4. **Object Removal Comparison.** Our method accurately removes the target object, demonstrating superior 3D segmentation compared to GauGroup [21]. A more precise removal leads to better inpainting results. We report the Average Mask Coverage Ratio(AMCR) [% ↓], indicating the proportion of empty regions in the image, lower values reflect better segmentation effectiveness.

Ablation on Loss Term. In Fig. 5, we validate the effectiveness of the spatial similarity loss function described for object ID distillation. The results demonstrate that incorporating this loss significantly improves artifact removal and preserves complex object boundaries during object removal.

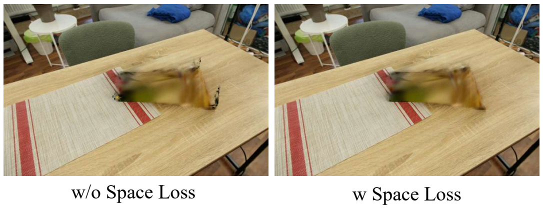


Figure 5. **Ablation on Spatial Similarity Loss.** Without the spatial similarity loss, object removal on complex structures leaves significant artifacts.

Ablation on Depth-guided Inpainting. In Fig. 6, we demonstrate that incorporating a depth prior dramatically accelerates convergence, achieving a reasonably good result within only 200 steps.

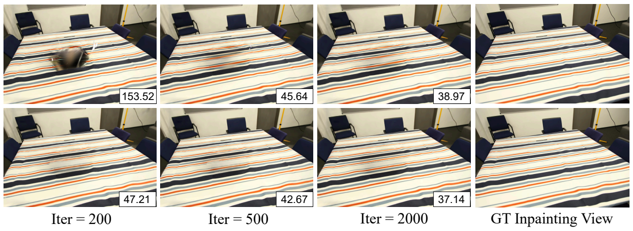


Figure 6. **Ablation on Depth-guided Inpainting.** We report the FID score [↓] here. With depth-guided inpainting we can achieve faster convergence and better quality.

Ablation on 2D Segmentation Foundation Model Selection. As shown in Fig. 7, while Gaga [11] adopts SAM [8] and utilizes 20% of the Gaussians within the mapped region to distinguish foreground and background, our method employs HQSAM [13] combined with K-means clustering for this task. Driven by a more compact loss function, our approach achieves a 5× speed-up in overall efficiency, enabling the potential for interactive applications.

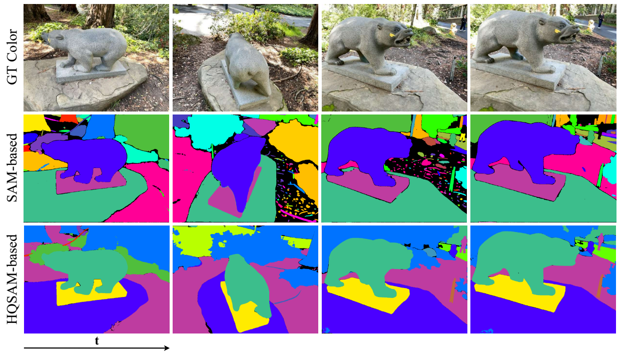


Figure 7. **Ablation on 2D Segmentation Foundation Models between SAM [8] and HQSAM [13] on Instruct-NeRF2NeRF [4] dataset.**

Analysis of Mask Association on Corner Case (Validation of K-Means $K = 2$ stability). In Fig. 9, we visualize the rendered objects after distillation on the LERF [7] dataset. This particular scene is challenging due to its high object density and the presence of extreme bird's-eye view angles. Such conditions pose significant difficulties for foreground-background separation using our K-means-based binary clustering. As shown, the DEVA-based GauGroup [21] produces noisy and inconsistent reconstructions under these settings. Such as `red chair` on the table, its thin leg can not be segmented correctly. In contrast, our method exhibits robust performance across different viewpoints. Effective multi-view scene segmentation in such cases is crucial for accurate object removal in subsequent stages.

Nevertheless, the method also struggles when applying K-Means clustering with $K = 2$. For certain objects, such as the `old camera` and the `gray pumpkin`, the algorithm incorrectly assigns them to the same object ID while attempting to separate foreground from background. Empirically, inspired by the strategy adopted in Gaga [11], we add an additional parameter that retains only 50% of the points in the foreground for this specific scene. This adjustment enables correct segmentation, suggesting that more effective methods or parameter choices remain to be explored.

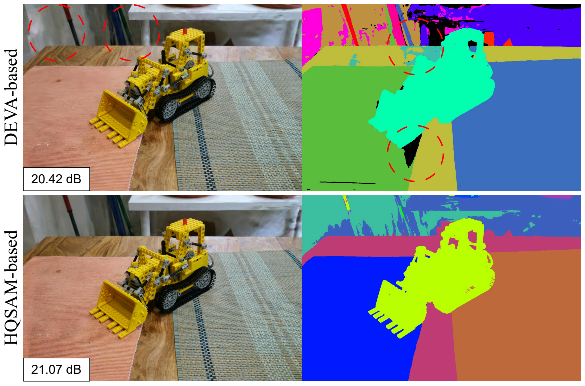


Figure 8. **Performance on Sparse View Inputs.** Our two-stage method can achieve a constantly better rendering quality(*e.g.*, background) and segmentation result.

Analysis of Mask Association on Corner Case (Sparse View). To validate the effectiveness of our mask association under sparse-view settings, we selected 1/8 of the images (35 out of 279) from the `kitchen` scene of MipNeRF360. We compare our method against GauGroup, which is based on DEVA. The results show that our approach remains robust. We attribute this to the fact that our method performs mask association directly in the 3D point cloud, whereas DEVA treats the problem as a video signal, which introduces significant challenges. As illustrated in Fig. 8, our method achieves superior rendering quality and, moreover, provides more consistent and unified segmentation results.

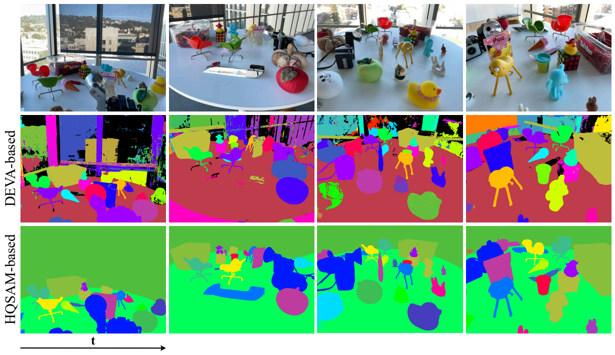


Figure 9. **Performance on Corner Case in the LERF [7] Dataset.**

Ablation on 2D Inpainting Model. Many recent methods introduce diffusion models for 2D inpainting [2, 14]. However, these models often produce visually plausible but semantically uncontrollable textures. Achieving view-consistent textures across multiple perspectives becomes particularly challenging. As a result, approaches like AuraFusion360 [20] and ImFusion [16] require extensive per-scene finetuning to enforce multi-view consistency. In Fig. 10, we compare LaMa [17] and LeftRefill [2]. While diffusion-based methods show high-quality results, our choice of LaMa offers a more efficient alternative, aligning with our emphasis on practical and scalable scene reconstruction. Exploring diffusion models with lightweight finetuning remains a promising future direction.

Ablation on Number of Virtual Camera Views. In Tab. 2, we investigate how the number of virtual camera views affects the inpainting performance in terms of FID. We report results under two settings: (1) using only the constrained training views, and (2) using virtual views without conditional previous-frame guidance. The performance curves show that our model converges when approximately 30 virtual views are used, demonstrating the effectiveness and sufficiency of our view sampling strategy.

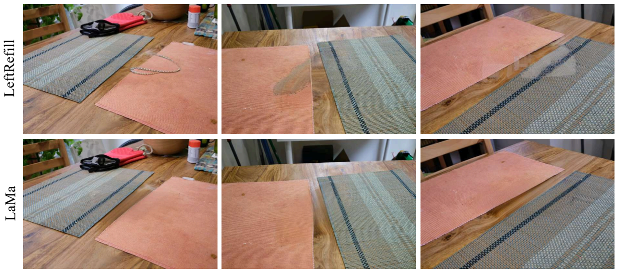
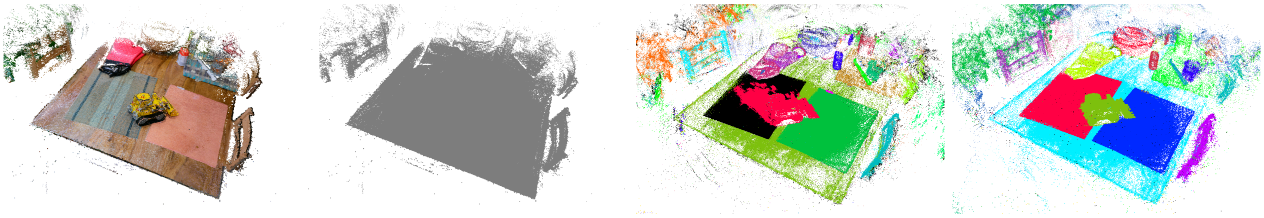


Figure 10. **Ablation on 2D Inpainting Model.**



(a) 3D Gaussian Field for Reference (b) 3D Gaussian Field w/o Object ID (c) 3D Gaussian Field after 2D Object Mask Association (d) 3D Gaussian Field after associated 2D Object Mask Distillation

Figure 11. **Visualization of Point Cloud with/without Object IDs Information** on `kitchen` scene [1]. After obtaining the pure Gaussian field through a standard 3D reconstruction process (a), we leverage mask association to generate (c), a raw and noisy point cloud with initial identity labels. Through identity distillation, we finally obtain (d), where consistent 2D identities are embedded into the 3D Gaussian field.

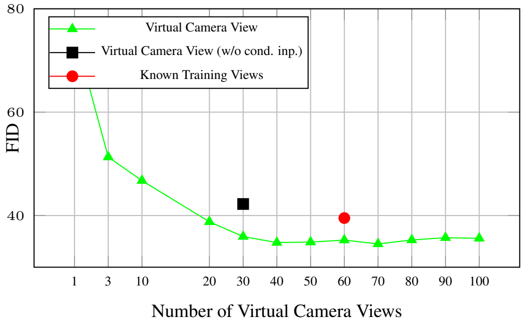


Table 2. Impact of the Number of Virtual Camera Views on FID.

Hyperparameter Selection for the Perceptual Loss. In Tab. 3, we demonstrate the impact of varying LPIPS (perceptual loss) weights on the FID of our reconstructed views.

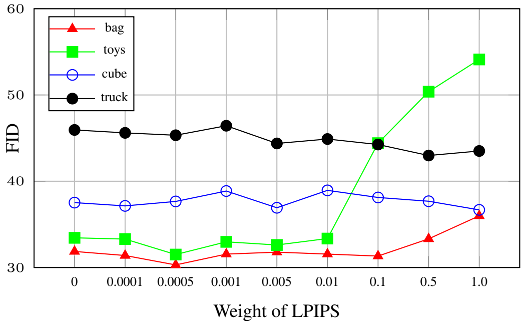


Table 3. Impact of Weight of LPIPS on FID.

Analysis of the Gaussian ID Distillation Process. In Fig. 11, we visualize the process of distilling object IDs into the Gaussian field. Our pipeline begins with a reconstructed pure Gaussian field (a). We first initialize per-Gaussian object ID features to obtain (b). After performing mask association, we obtain (c), a Gaussian field with raw object identity labels. However, the mask association process is primarily used to generate view-consistent segmentation masks across frames. Finally, through our 3D distillation process, the refined and consistent object identities are embedded into the Gaussian field, as shown in (d).

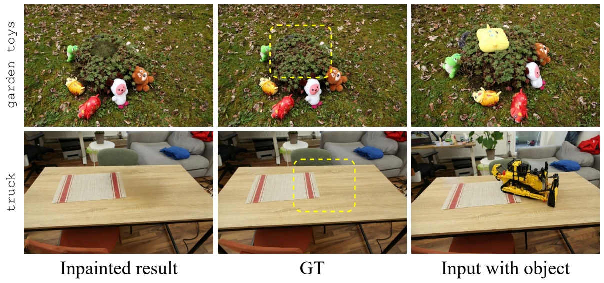


Figure 12. **Inpainting Failure Case.**

Discussion on limitation and feature work. The main limitation of our work lies in the inpainting stage after object removal, as illustrated in Fig. 12. First, our method is not able to properly handle shadows cast by removed objects. Second, to balance computational efficiency with the need for producing reasonable and controllable results, we employ LaMa as the 2D inpainter. However, this choice limits the inpainting quality in scenes with complex textures, where LaMa often fails to reconstruct fine-grained details. Diffusion-based methods,*e.g.*, AuraFusion360 [20], can generate complex textures from a single view but struggle to ensure consistency across multiple views, and their refinement typically requires long inference times.

4. Per-Scene Breakdown of the Results.

In Fig. 13 to Fig. 24, we provide detailed multi-view comparisons across different scenes, and the per-scene quantitative results in Tab. 4 further confirm the robustness and consistency of our method. However, floaters can still be observed in NBS regions from certain viewpoints (see our video), which mainly stem from inconsistencies in the inpainting results across views. This highlights the need for developing more consistent and efficient inpainters in future work.

When comparing to baselines, we observe that GScream [19] achieves stronger performance than SPIn-NeRF [12] on full-image metrics, but performs worse in the masked regions because it cannot reliably remove target objects. In contrast, SPIn-NeRF demonstrates better handling of object removal, which results in improved performance on masked-area evaluations.

In addition, to further validate its applicability in forward-facing scenarios, we compare our approach with GauGroup on SPIn-NeRF [12] dataset. Our method still delivers superior results, highlighting its scalability and generalization ability beyond 360° settings.

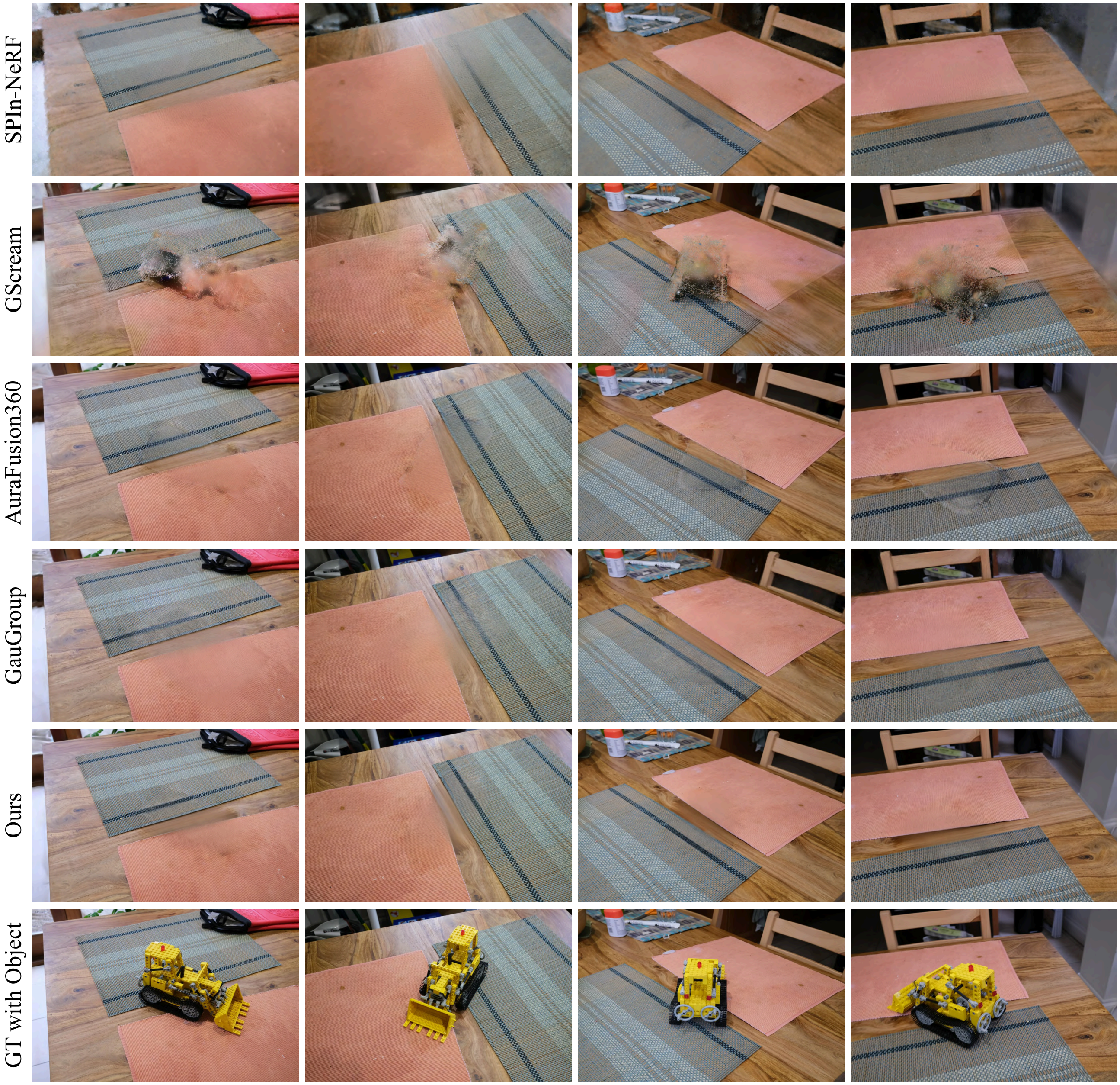


Figure 13. **Multi-view comparison on Mip-NeRF 360 [1]** `kitchen`. We evaluate SPIn-NeRF [12], GScream [19], AuraFusion360 [20], GauGroup [21] and our method, with object-inclusive ground truth images provided for each corresponding view. Each column represents a distinct viewpoint, and four representative angles are selected to comprehensively demonstrate the performance across the full set of views. Our method achieves superior multi-view consistency with detailed texture and smooth boundary compared to the baseline approaches.

Scene	Methods	PSNR ↑	masked PSNR ↑	SSIM ↑	masked SSIM↑	LPIPS ↓	masked LPIPS ↓	FID ↓
fruits	SPIn-NeRF [12]	11.15	34.21	0.4617	0.9963	0.6253	0.0056	367.19
	GScream [19]	23.39	31.03	0.8506	0.9934	0.2559	0.0091	80.17
	AuraFusion [20]	23.93	37.38	0.8617	0.9975	0.2450	0.0042	61.94
	GauGroup [21]	22.55	35.92	0.8485	0.9973	0.2365	0.0043	60.13
	Inpaint360GS (Ours)	27.38	44.54	0.9014	0.9993	0.1657	0.0011	30.33
doppelherz	SPIn-NeRF [12]	21.24	41.66	0.5421	0.9986	0.5227	0.0031	258.82
	GScream [19]	24.56	38.73	0.8108	0.9958	0.1849	0.0030	88.09
	AuraFusion [20]	27.81	44.39	0.8545	0.9989	0.1379	0.0014	32.56
	GauGroup [21]	27.40	43.69	0.8787	0.9991	0.1096	0.0013	44.90
	Inpaint360GS (Ours)	29.2	46	0.9129	0.9994	0.0789	0.0009	20.13
toys	SPIn-NeRF [12]	25.97	39.79	0.6558	0.9919	0.3785	0.0086	119.03
	GScream [19]	25.27	31.68	0.8164	0.9865	0.1860	0.0138	376.61
	AuraFusion [20]	27.05	39.94	0.8011	0.9917	0.1996	0.0073	41.03
	GauGroup [21]	24.08	34.90	0.7683	0.9886	0.1796	0.0065	64.97
	Inpaint360GS (Ours)	28.14	40.58	0.8707	0.9928	0.0995	0.0053	33.29
garden toys	SPIn-NeRF [12]	21.79	33.57	0.5730	0.9855	0.3778	0.0134	116.17
	GScream [19]	21.01	28.60	0.7066	0.9841	0.2358	0.0130	130.18
	AuraFusion [20]	21.34	30.49	0.7147	0.9834	0.2372	0.0134	64.41
	GauGroup [21]	22.41	33.56	0.7585	0.9850	0.1590	0.0103	48.70
	Inpaint360GS (Ours)	23.68	33.71	0.8094	0.9857	0.1228	0.0098	30.58
bag	SPIn-NeRF [12]	23.08	34.39	0.5278	0.9872	0.4728	0.0076	124.15
	GScream [19]	24.84	32.52	0.7913	0.9827	0.2264	0.0124	187.60
	AuraFusion [20]	26.46	34.22	0.8211	0.9861	0.2056	0.011	55.12
	GauGroup [21]	26.28	35.04	0.827	0.9874	0.1586	0.0062	33.74
	Inpaint360GS (Ours)	27.97	37.45	0.8627	0.9887	0.1263	0.0056	31.41
car	SPIn-NeRF [12]	19.15	22.12	0.3901	0.9456	0.5541	0.0485	334.78
	GScream [19]	19.35	23.02	0.7015	0.9474	0.2741	0.0413	324.76
	AuraFusion [20]	21.22	26.01	0.7718	0.9524	0.1769	0.0283	67.82
	GauGroup [21]	18.43	24.65	0.6516	0.9468	0.2609	0.0388	157.23
	Inpaint360GS (Ours)	20.71	27.96	0.7309	0.9475	0.1943	0.0357	88.95
red cone	SPIn-NeRF [12]	18.71	32.04	0.3572	0.9929	0.5177	0.0094	127.88
	GScream [19]	19.31	30.53	0.6970	0.9866	0.2528	0.0121	84.36
	AuraFusion [20]	20.55	36.14	0.7526	0.9927	0.1967	0.0077	31.02
	GauGroup [21]	21.14	37.44	0.7744	0.9914	0.1346	0.0053	19.97
	Inpaint360GS (Ours)	21.45	38.83	0.7973	0.9933	0.1201	0.0051	21.42
yellow cone	SPIn-NeRF [12]	17.92	36.09	0.3130	0.9893	0.6374	0.0087	379.17
	GScream [19]	24.77	33.21	0.8124	0.9880	0.1775	0.0089	140.88
	AuraFusion [20]	25.90	39.06	0.8195	0.9912	0.1590	0.0049	35.78
	GauGroup [21]	26.32	39.99	0.8480	0.9921	0.1171	0.0035	28.78
	Inpaint360GS (Ours)	26.33	42.51	0.8642	0.9926	0.0935	0.0039	21.38
cube	SPIn-NeRF [12]	17.52	27.32	0.6621	0.9708	0.4315	0.0279	351.46
	GScream [19]	15.32	22.09	0.6596	0.9703	0.4321	0.0290	396.07
	AuraFusion [20]	22.48	27.82	0.8645	0.9807	0.1506	0.0118	43.24
	GauGroup [21]	20.10	27.51	0.8127	0.9749	0.2071	0.0197	118.93
	Inpaint360GS (Ours)	22.52	28.58	0.8879	0.9874	0.1079	0.0083	37.14
redbull	SPIn-NeRF [12]	20.98	41.00	0.4699	0.9973	0.4691	0.0052	186.81
	GScream [19]	19.24	26.42	0.6218	0.9923	0.3637	0.0087	286.52
	AuraFusion [20]	23.22	40.80	0.7258	0.9982	0.2178	0.0021	47.57
	GauGroup [21]	23.06	41.36	0.7409	0.9981	0.1870	0.0025	63.98
	Inpaint360GS (Ours)	23.55	42.62	0.7655	0.9988	0.1573	0.0014	34.94
truck	SPIn-NeRF [12]	24.23	30.54.99	0.7604	0.9919	0.3626	0.0101	164.01
	GScream [19]	21.93	27.16	0.8458	0.9813	0.1838	0.0181	173.49
	AuraFusion [20]	25.51	31.43	0.8763	0.9903	0.1675	0.0081	49.30
	GauGroup [21]	23.70	28.39	0.8829	0.9898	0.1465	0.0115	83.21
	Inpaint360GS (Ours)	25.62	33.99	0.9172	0.9923	0.0975	0.0080	45.63
avg.	SPIn-NeRF [12]	19.71	34.53	0.5000	0.9854	0.5002	0.0140	229.95
	GScream [19]	20.95	28.47	0.7380	0.9819	0.2715	0.0161	206.25
	AuraFusion360 [20]	23.15	35.78	0.7923	0.9872	0.1915	0.0097	47.71
	GauGroup [21]	23.20	35.73	0.7928	0.9862	0.1770	0.0102	65.87
	Inpaint360GS (Ours)	24.40	36.29	0.8370	0.9886	0.1300	0.0078	35.93

Table 4. **Per scene quantitative comparison on the Inpaint360GS dataset.**

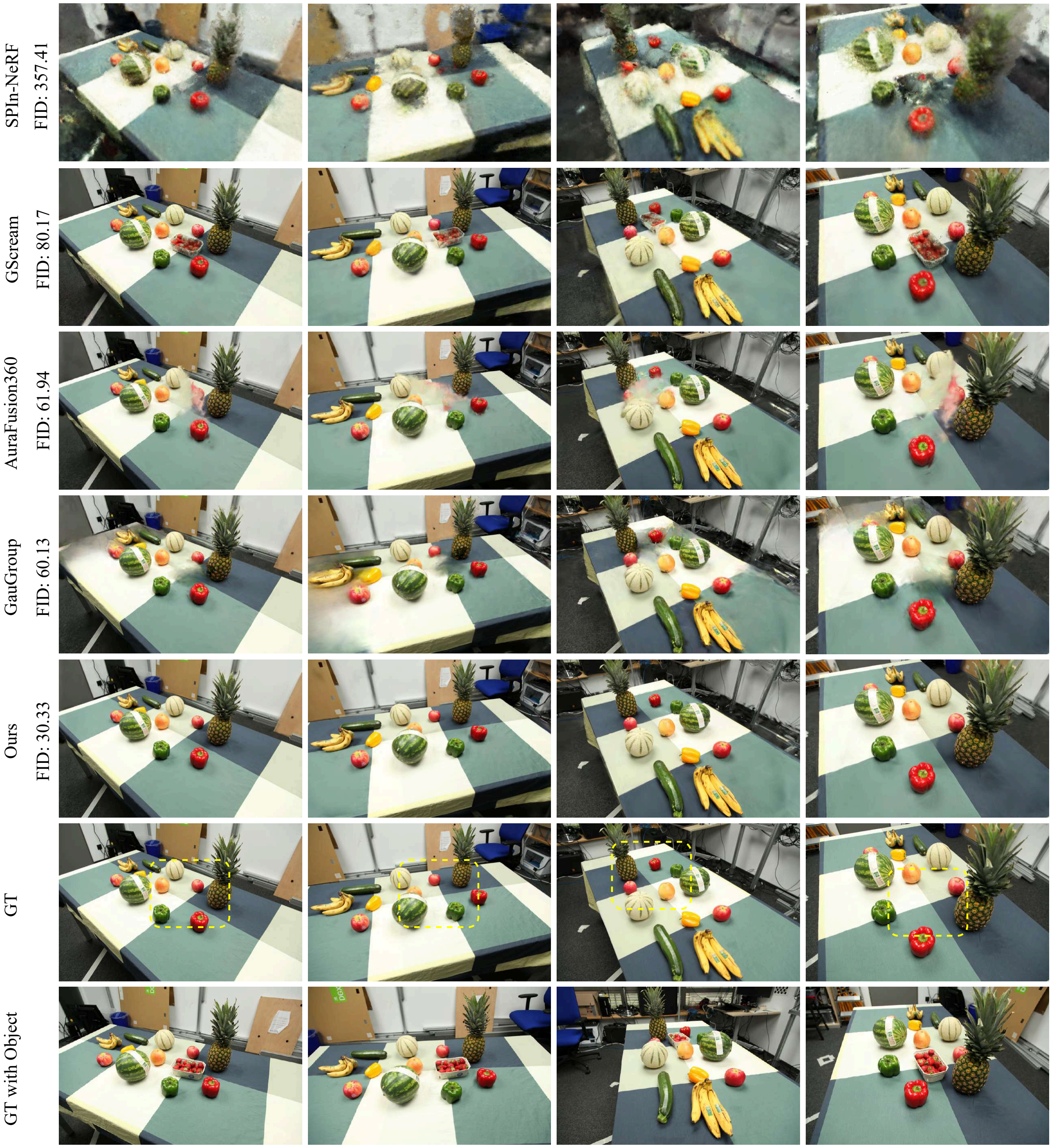


Figure 14. **Multi-view comparison on Inpaint360GS** `fruits`. We evaluate SPIn-NeRF [12], GScream [19], AuraFusion360 [20], GauGroup [21] and our method, with object-inclusive ground truth images provided for each corresponding view. In the scene with multiple objects, our method demonstrates a clear advantage. This can be attributed to our precise object ID assignment within the Gaussian field, which is further integrated into the virtual camera view. As a result, our method is able to identify more accurate never-been-seen (NBS) regions. We attribute the above performance gains to these key design choices.

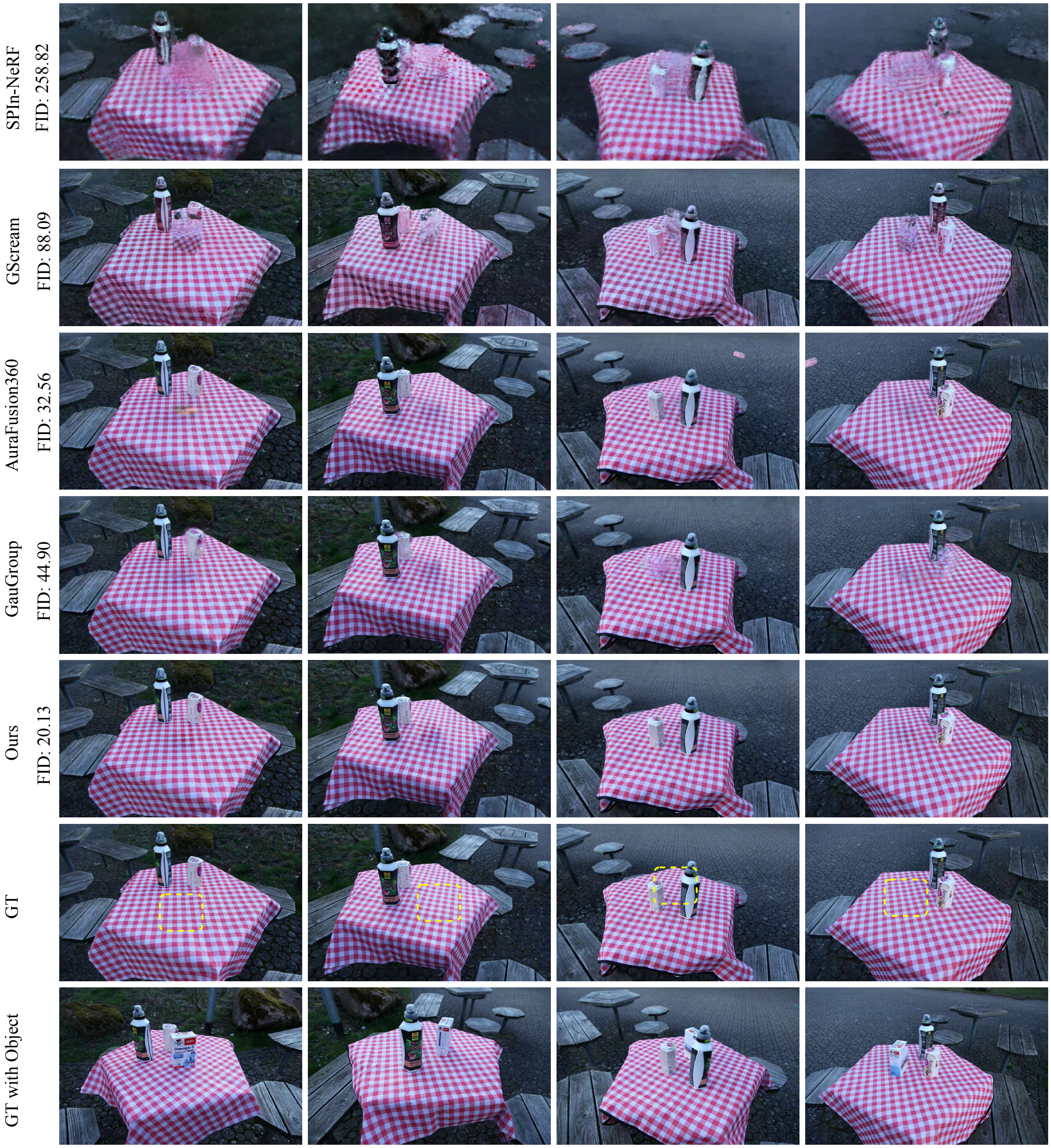


Figure 15. **Multi-view comparison on Inpaint360GS** `doppelherz`. We evaluate SPIn-NeRF [12], GScream [19], AuraFusion360 [20], GauGroup [21] and our method, with object-inclusive ground truth images provided for each corresponding view. The scene poses significant challenges due to distant viewpoints and multiple objects, making NBS region detection unreliable. While AuraFusion360 suffers from floating textures due to poor depth alignment, our method remains robust, benefiting from the structured virtual camera trajectory that facilitates consistent and accurate NBS region identification. Our approach first removes occluding objects and then performs inpainting, enabling efficient utilization of scene information for faithful reconstruction.

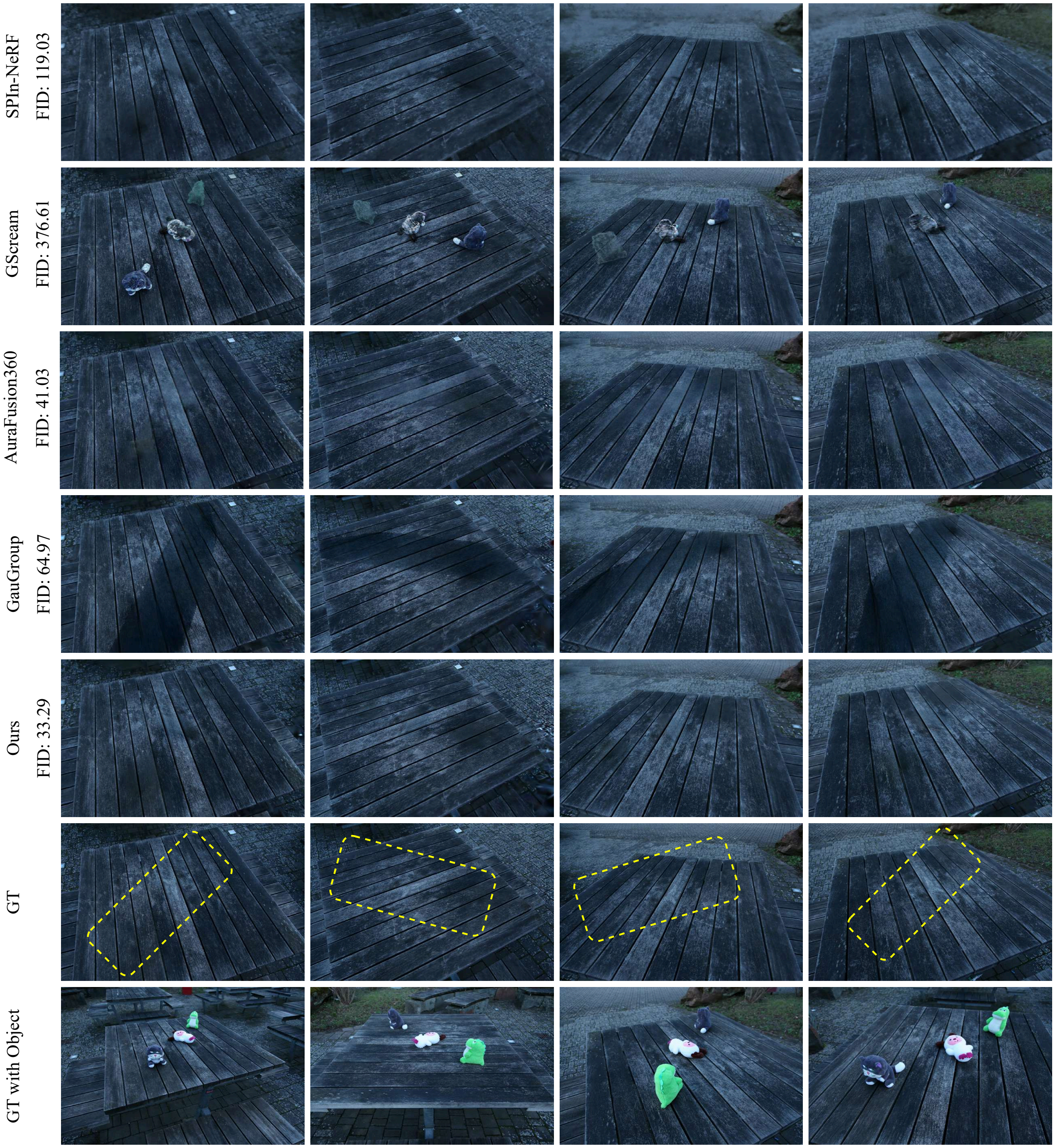


Figure 16. **Multi-view comparison on Inpaint360GS** `toys`. We evaluate SPIn-NeRF [12], GScream [19], AuraFusion360 [20], GauGroup [21] and our method, with object-inclusive ground truth images provided for each corresponding view. This scene, though containing multiple objects, is relatively simple due to the sparse layout and lack of occlusion. Both AuraFusion360 and SPIn-NeRF demonstrate visually pleasing results under this setting. Nonetheless, our method achieves more consistent appearance across views.

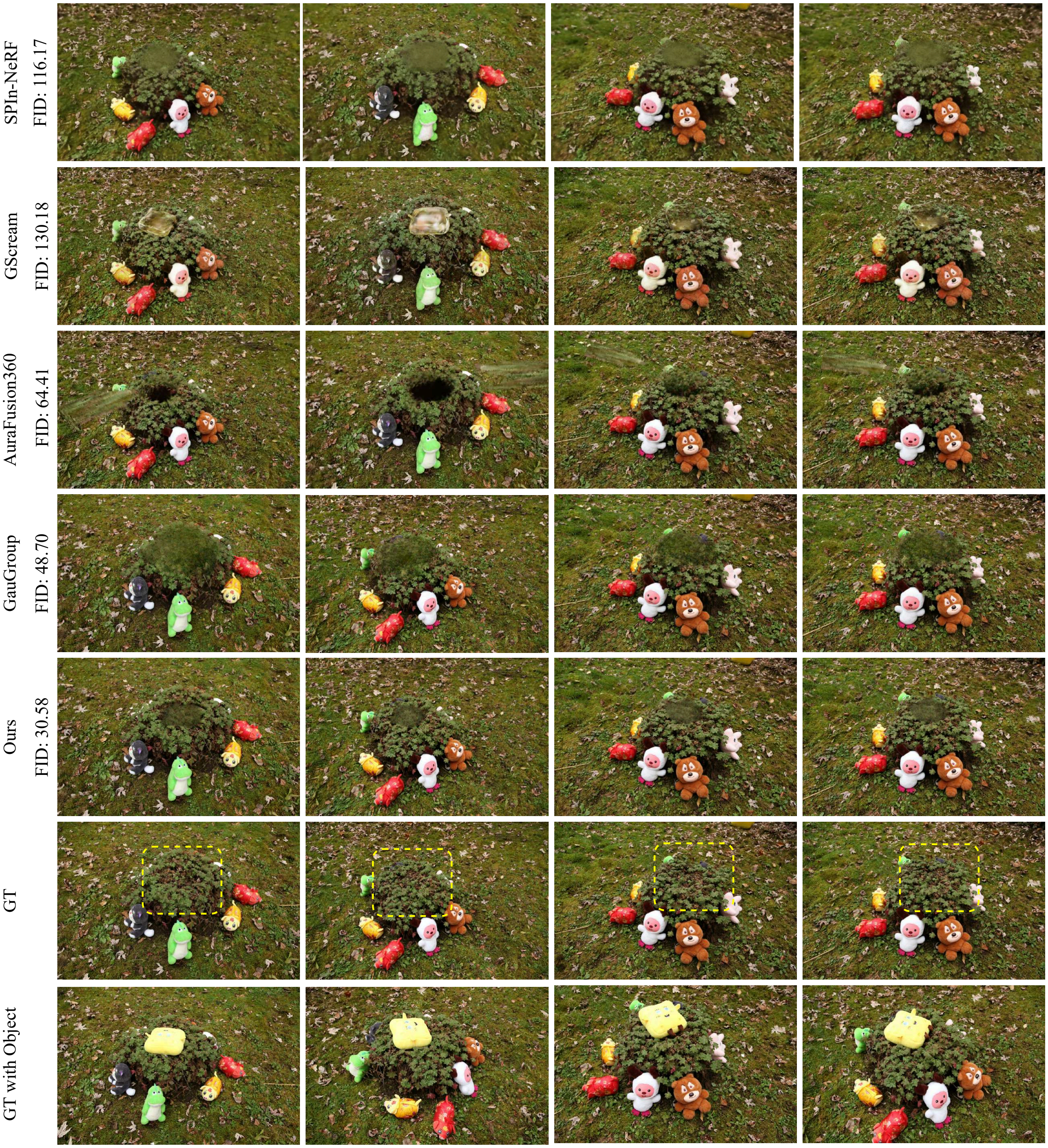


Figure 17. **Multi-view comparison on Inpaint360GS** `garden toys`. We evaluate SPIn-NeRF [12], GScream [19], AuraFusion360 [20], GauGroup [21] and our method, with object-inclusive ground truth images provided for each corresponding view.This scene is particularly challenging due to the unpredictable NBS region and the stochastic nature of the leaf textures. Our chosen 2D inpainting model (LaMa), while efficient, lacks the generative capacity of diffusion-based models to synthesize such fine-grained details. Nevertheless, our method achieves the best overall visual quality among all baselines, despite lacking highly detailed textures.

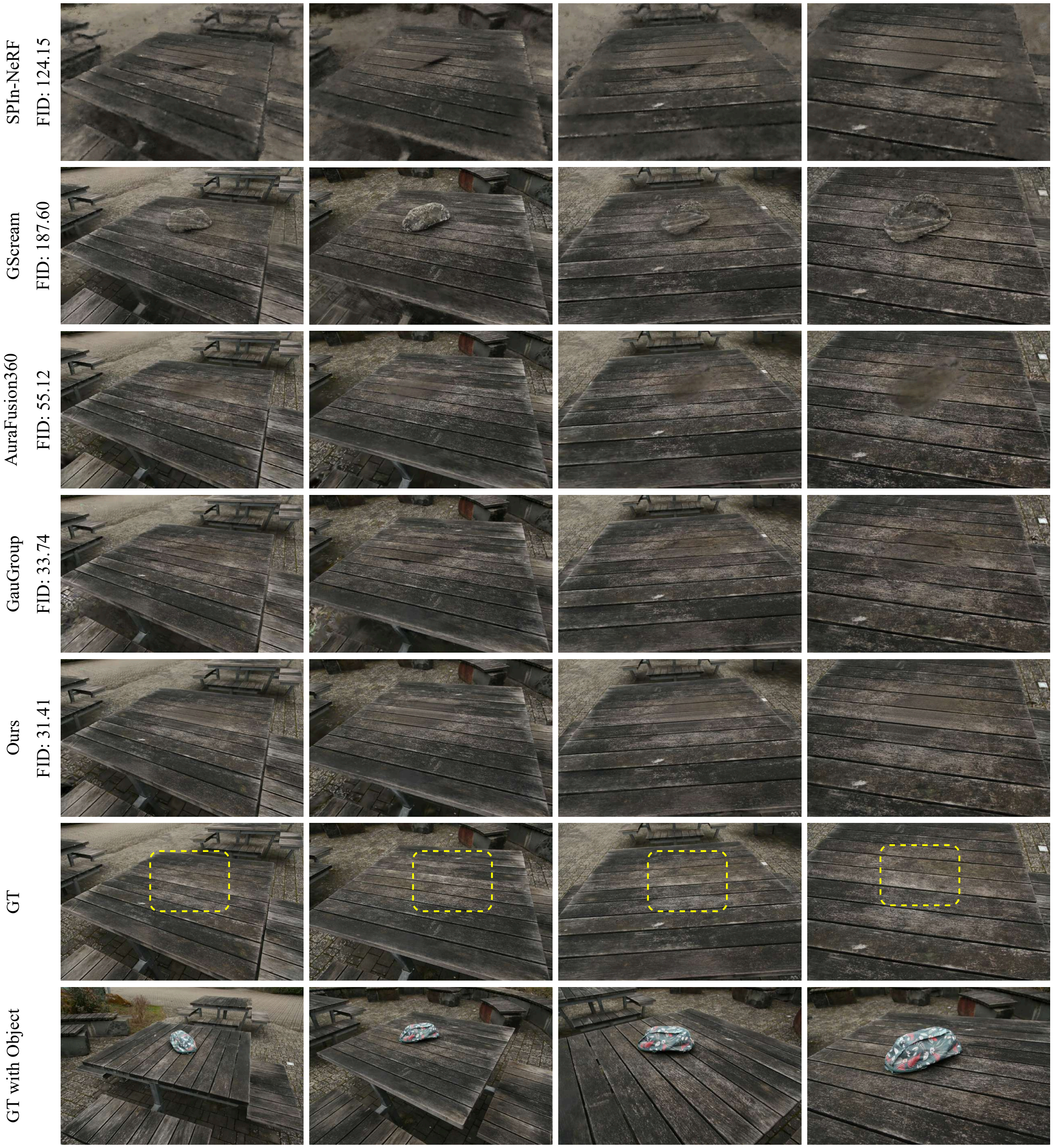


Figure 18. **Multi-view comparison on Inpaint360GS** `bag`. We evaluate SPIn-NeRF [12], GScream [19], AuraFusion360 [20], GauGroup [21] and our method, with object-inclusive ground truth images provided for each corresponding view. Our method achieves the best FID score, produces noticeably smoother edges, and is approximately 5 × faster than the 3D inpainting stage of the second-best method GauGroup [21].

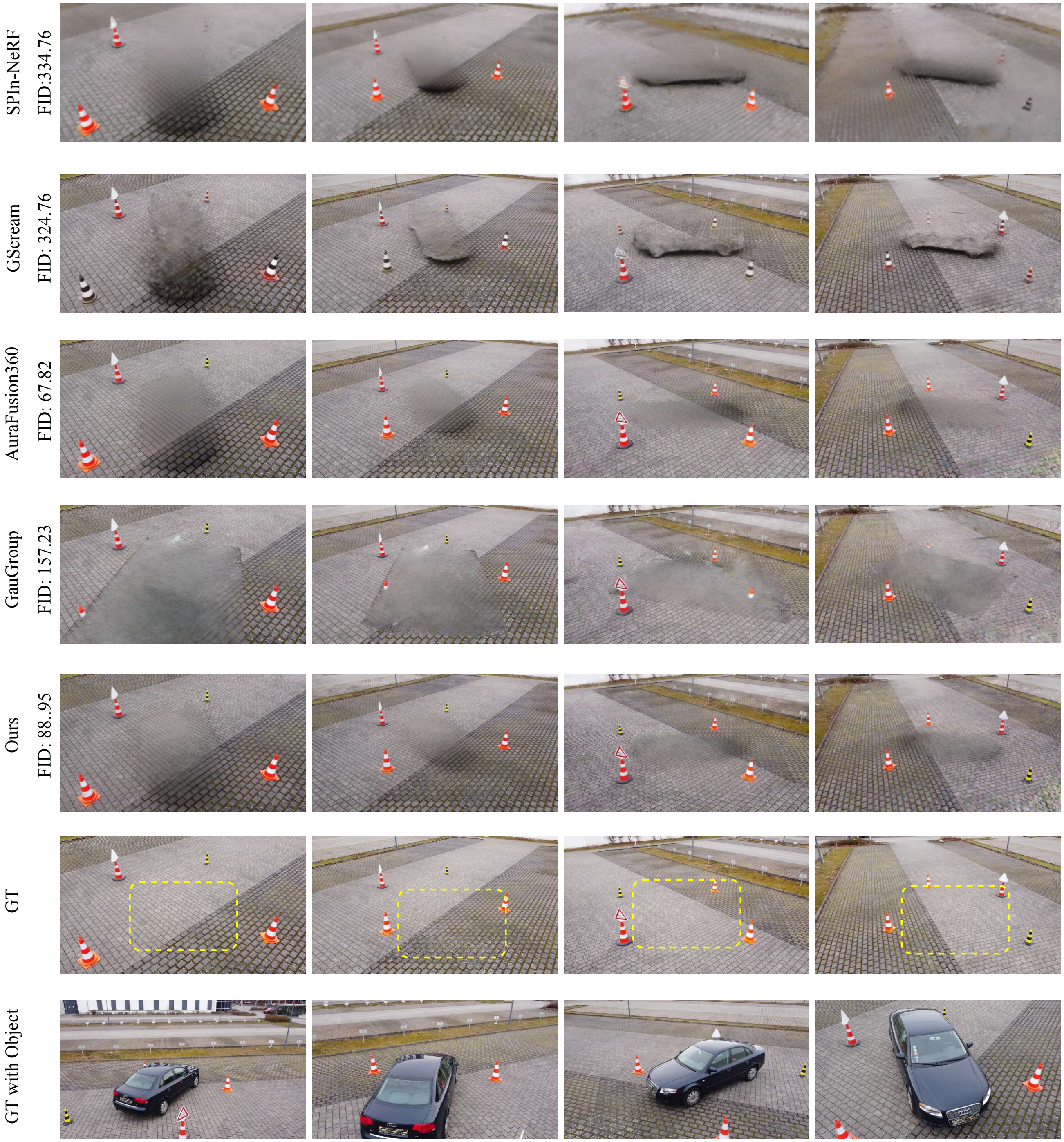


Figure 19. **Multi-view comparison on Inpaint360GS** `car`. We evaluate SPIn-NeRF [12], GScream [19], AuraFusion360 [20], GauGroup [21] and our method, with object-inclusive ground truth images provided for each corresponding view.This scene is particularly challenging due to the complex and texture-less ground surface, which makes it difficult to infer plausible textures. AuraFusion360 achieves strong FID performance due to its single-view guidance combined with extensive post-refinement. However, its optimization time is approximately 20× longer than ours. In contrast, our method achieves competitive results with a significantly more efficient pipeline.

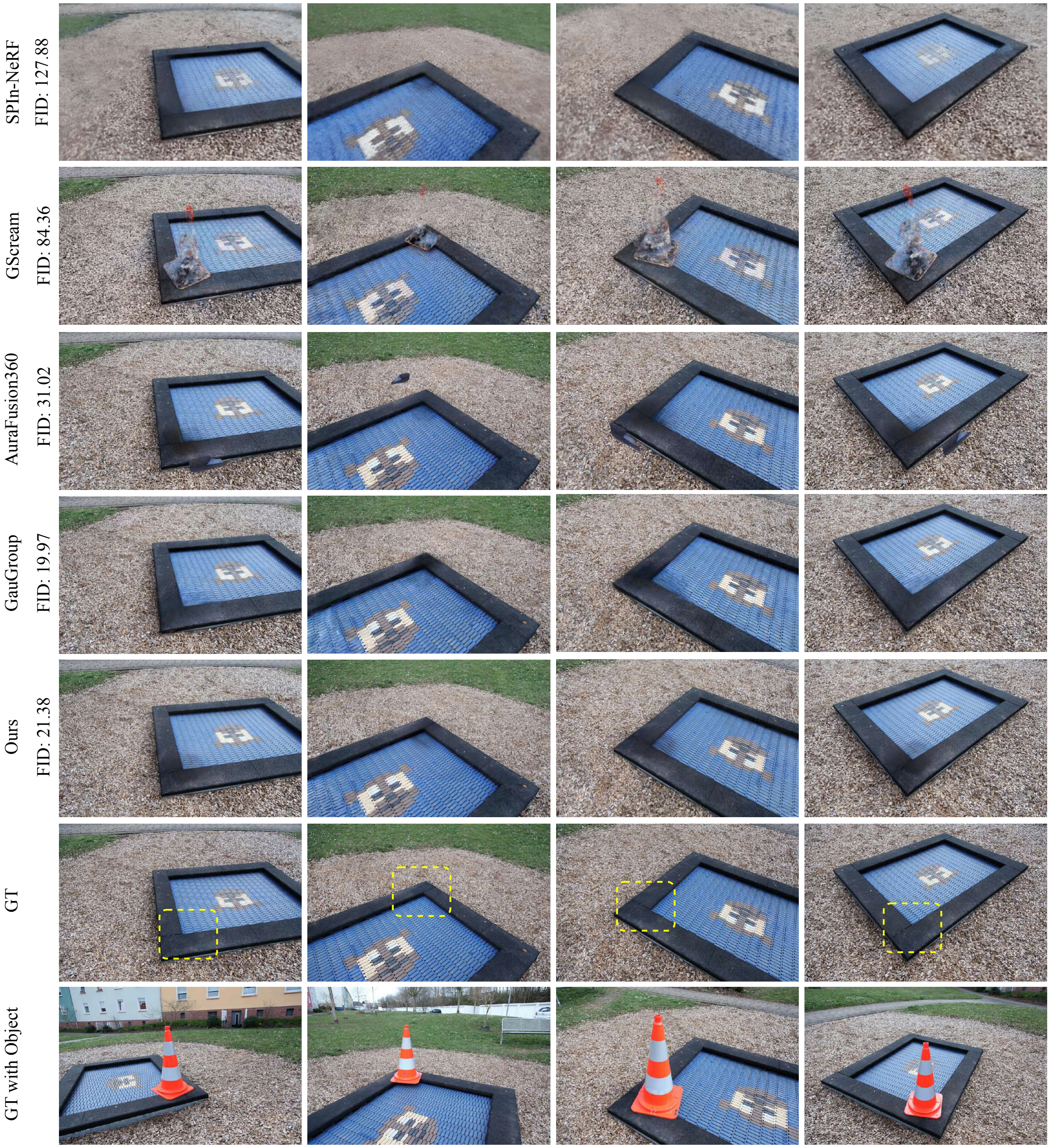


Figure 20. **Multi-view comparison on Inpaint360GS** `red cone`. We evaluate SPIn-NeRF [12], GScream [19], AuraFusion360 [20], GauGroup [21] and our method, with object-inclusive ground truth images provided for each corresponding view. This scene presents a challenging case due to significant depth variations and complex textures, making accurate inpainting difficult. GauGroup achieves the best visual quality, while our method performs comparably, producing plausible results with effective depth reasoning.

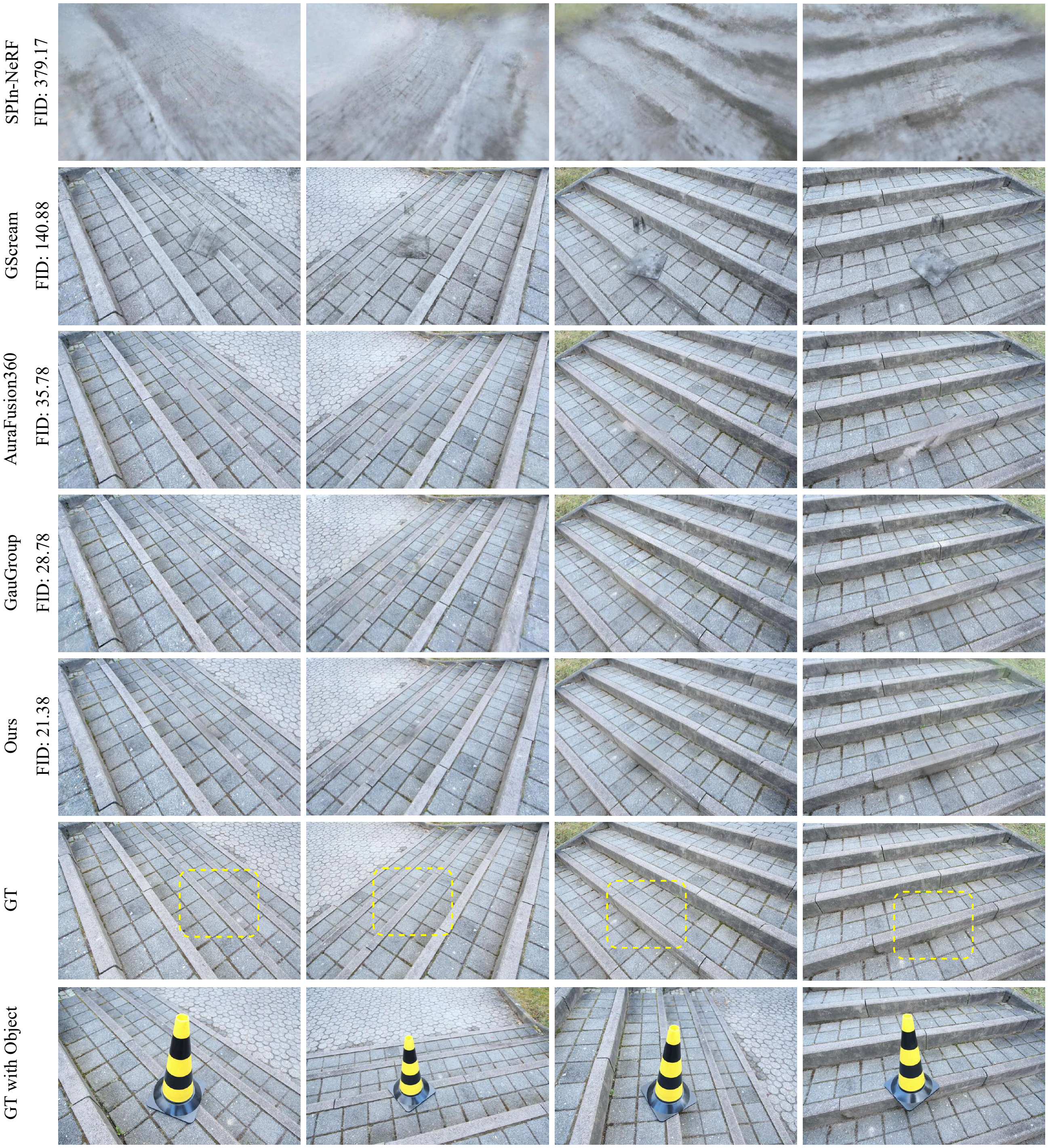


Figure 21. **Multi-view comparison on Inpaint360GS** `yellow cone`. We evaluate SPIn-NeRF [12], GScream [19], AuraFusion360 [20], GauGroup [21] and our method, with object-inclusive ground truth images provided for each corresponding view. This scene includes a staircase, posing a challenge for depth estimation. Our method converges efficiently and maintains strong performance. Notably, GauGroup [21] achieves the second-best results but requires 5× longer optimization time.

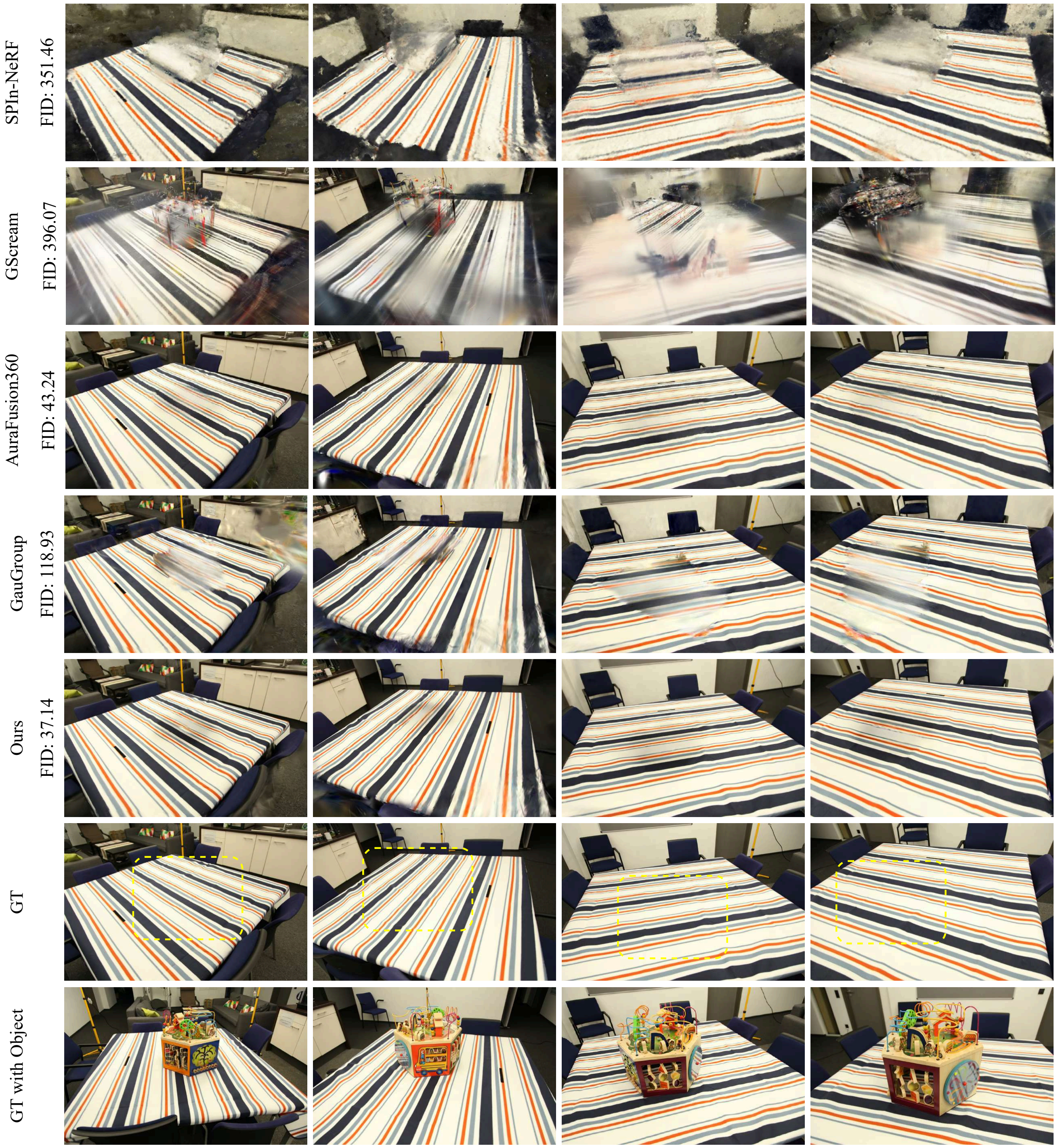


Figure 22. **Multi-view comparison on Inpaint360GS** `cube`. We evaluate SPIn-NeRF [12], GScream [19], AuraFusion360 [20], GauGroup [21] and our method, with object-inclusive ground truth images provided for each corresponding view. In this scene, GScream [19] encounters significant issues due to inconsistencies between the depth provided by Marigold [5] and the depth scale of the COLMAP-initialized point cloud. The failure of depth alignment leads to degraded performance. While AuraFusion360 demonstrates competitive performance, it exhibits noticeable boundary ambiguity in the inpainted regions. In contrast, our method avoids this problem by directly defining depth using intrinsic properties of the Gaussian scene, thereby eliminating the need for external depth alignment. As a result, our pipeline achieves the best performance.

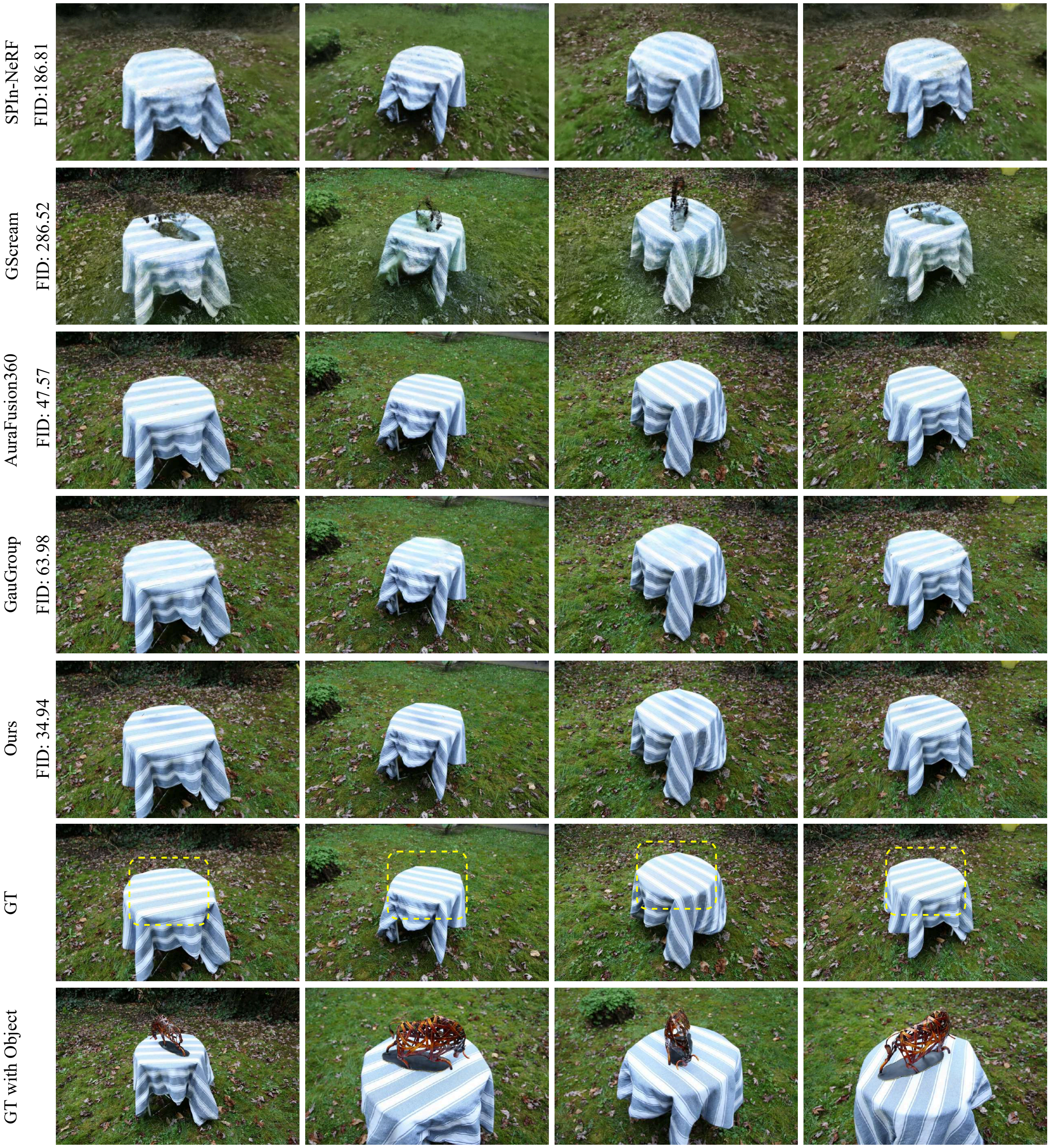


Figure 23. **Multi-view comparison on Inpaint360GS** `redbull`. We evaluate SPIn-NeRF [12], GScream [19], AuraFusion360 [20], GauGroup [21] and our method, with object-inclusive ground truth images provided for each corresponding view. Although this is a single-object scene, the bull model contains fine-grained structures such as horns and a tail, posing challenges for accurate 3D Gaussian identity assignment. All methods except GScream produce visually reasonable results under this setting. Please zoom in for details.

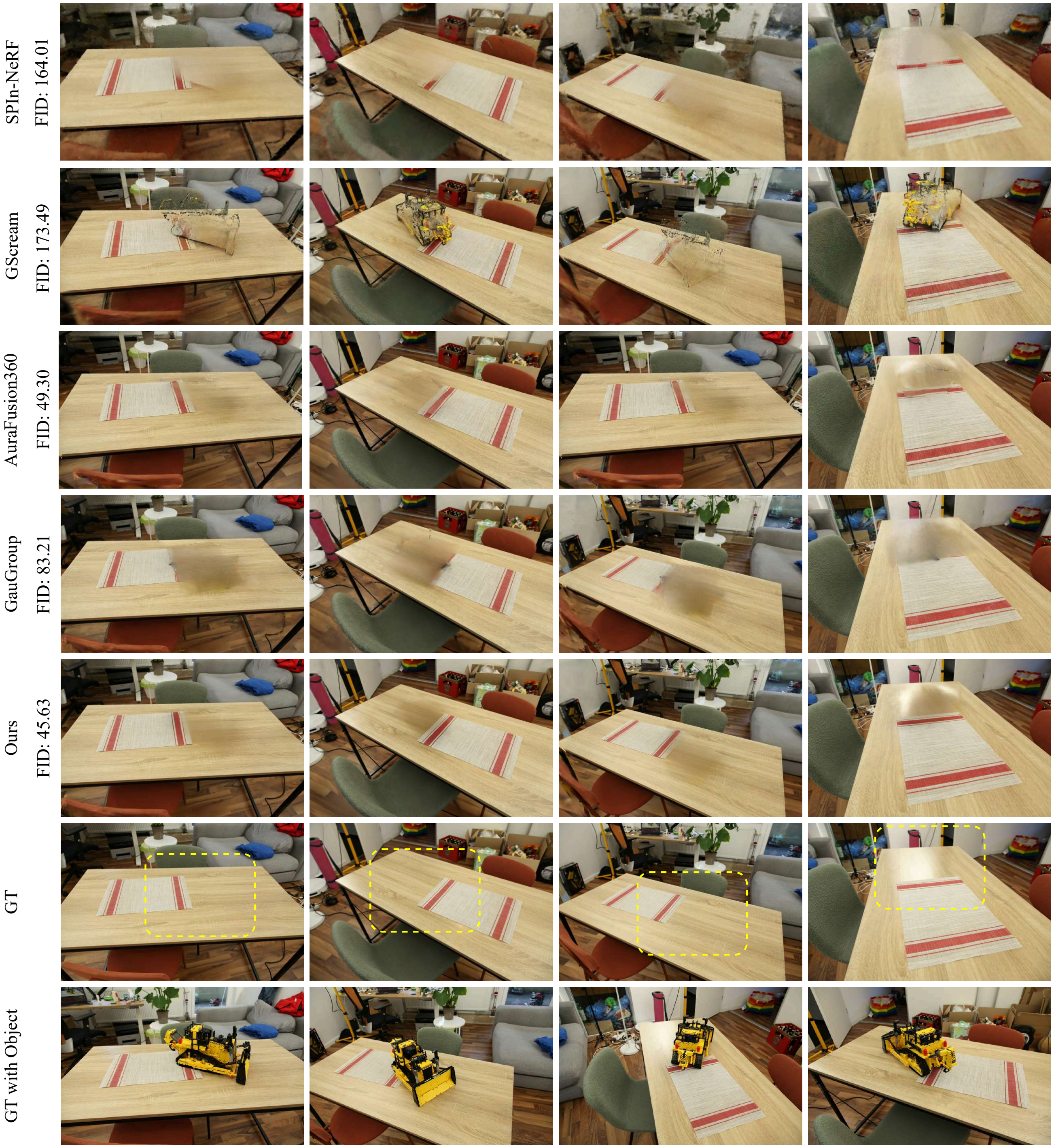


Figure 24. **Multi-view comparison on Inpaint360GS** `truck`. We evaluate SPIn-NeRF [12], GScream [19], AuraFusion360 [20], GauGroup [21] and our method, with object-inclusive ground truth images provided for each corresponding view. Our method achieves the best FID score and is 20 × faster than AuraFusion [20], while requiring no additional parameter tuning. However, none of the evaluated methods, including ours, are yet capable of effectively handling complex lighting and shadow effects present in the scene, which remains an open challenge for future research.

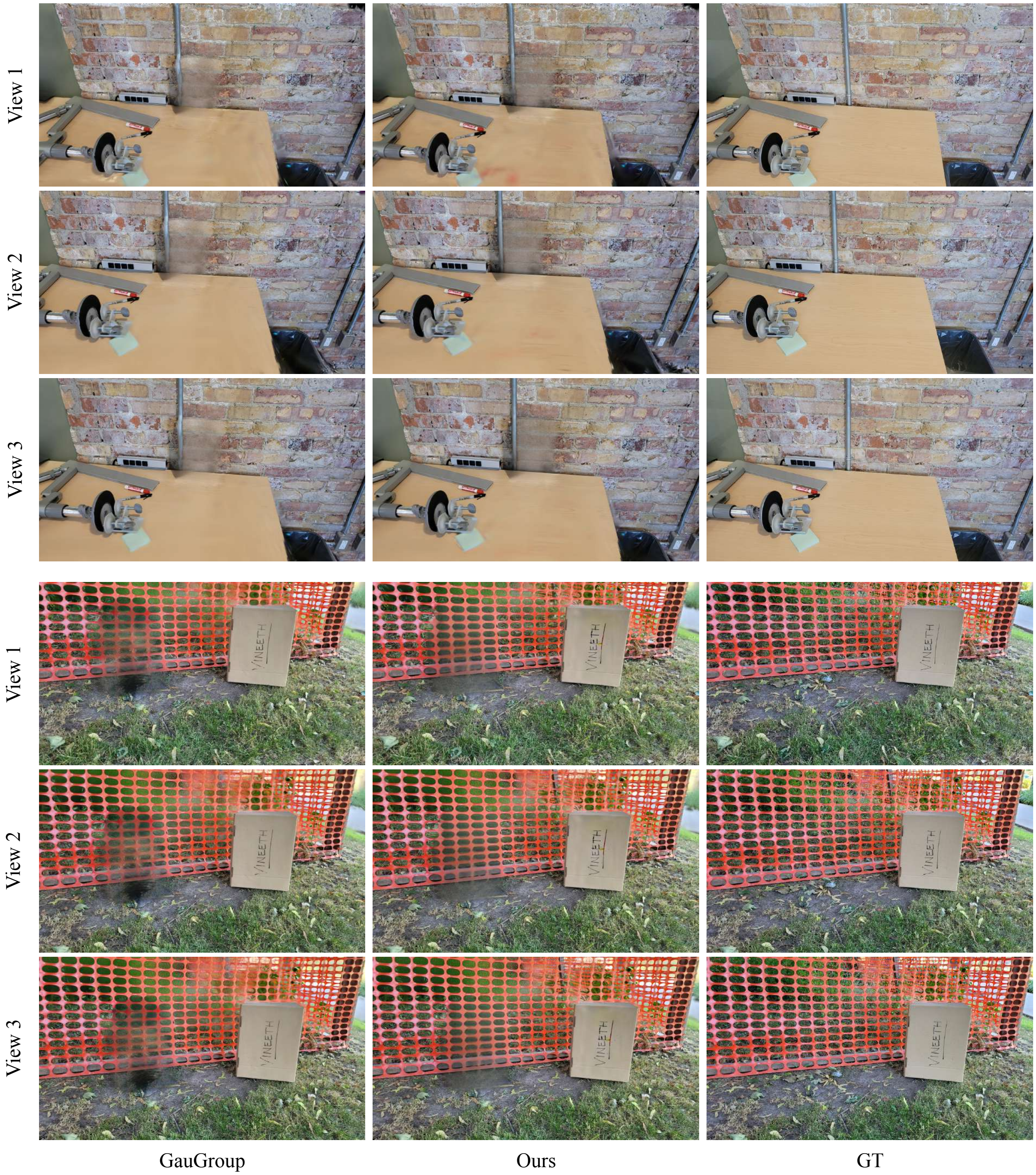


Figure 25. **Performance on SPIn-NeRF [12] Dataset.** We evaluate GauGroup [21] and our method on front facing SPIn-NeRF [12] dataset. Our method remains robust on this dataset and consistently outperforms GauGroup, achieving a 0.6 dB improvement in PSNR and a notable 5 points gain in FID.

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*, pages 5470–5479, 2022. 1, 2, 7, 9

[2] Chenjie Cao, Yunuo Cai, Qiaole Dong, Yikai Wang, and Yanwei Fu. Leftrefill: Filling right canvas based on left reference through generalized text-to-image diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7705–7715, 2024. 6

[3] Yangming Cheng, Liulei Li, Yuanyou Xu, Xiaodi Li, Zongxin Yang, Wenguan Wang, and Yi Yang. Segment and track anything. *arXiv preprint arXiv:2305.06558*, 2023. 3

[4] Ayaan Haque, Matthew Tancik, Alexei A Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19740–19750, 2023. 1, 2, 5

[5] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 19

[6] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. 3, 4

[7] Justin* Kerr, Chung Min* Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *International Conference on Computer Vision (ICCV)*, 2023. 6

[8] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 5

[9] Georgios Kopanas, Thomas Leimkühler, Gilles Rainer, Clément Jambon, and George Drettakis. Neural point catacaustics for novel-view synthesis of reflections. *ACM Transactions on Graphics (TOG)*, 41(6):1–15, 2022. 3

[10] Zhiheng Liu, Hao Ouyang, Qiuyu Wang, Ka Leong Cheng, Jie Xiao, Kai Zhu, Nan Xue, Yu Liu, Yujun Shen, and Yang Cao. Infusion: Inpainting 3d gaussians via learning depth completion from diffusion prior. *arXiv preprint arXiv:2404.11613*, 2024. 4

[11] Weijie Lyu, Xueting Li, Abhijit Kundu, Yi-Hsuan Tsai, and Ming-Hsuan Yang. Gaga: Group any gaussians via 3d-aware memory bank, 2024. 5, 6

[12] Ashkan Mirzaei, Tristan Aumentado-Armstrong, Konstantinos G Derpanis, Jonathan Kelly, Marcus A Brubaker, Igor Gilitschenski, and Alex Levinshtein. Spin-nerf: Multiview segmentation and perceptual inpainting with neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20669–20679, 2023. 1, 2, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22

[13] Lu Qi, Jason Kuen, Tiancheng Shen, Jiuxiang Gu, Weidong Guo, Jiaya Jia, Zhe Lin, and Ming-Hsuan Yang. High-quality entity segmentation. In *ICCV*, 2023. 2, 3, 5

[14] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 6

[15] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[16] Zhihao Shi, Dong Huo, Yuhongze Zhou, Yan Min, Juwei Lu, and Xinxin Zuo. Imfine: 3d inpainting via geometry-guided multi-view refinement. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26694–26703, 2025. 1, 2, 6

[17] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2149–2159, 2022. 3, 6

[18] Dongqing Wang, Tong Zhang, Alaa Abboud, and Sabine Süsstrunk. Innerf360: Text-guided 3d-consistent object inpainting on 360-degree neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12677–12686, 2024. 2

[19] Yuxin Wang, Qianyi Wu, Guofeng Zhang, and Dan Xu. Learning 3d geometry and feature consistent gaussian splatting for object removal. In *European Conference on Computer Vision*, pages 1–17. Springer, 2024. 4, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21

[20] Chung-Ho Wu, Yang-Jung Chen, Ying-Huan Chen, Jie-Ying Lee, Bo-Hsu Ke, Chun-Wei Tuan Mu, Yi-Chuan Huang, Chin-Yang Lin, Min-Hung Chen, Yen-Yu Lin, et al. Aurafusion360: Augmented unseen region alignment for reference-based 360deg unbounded scene inpainting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16366–16376, 2025. 1, 2, 4, 6, 7, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21

[21] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. In *European Conference on Computer Vision*, pages 162–179. Springer, 2024. 4, 5, 6, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22

[22] Wang Yifan, Felice Serena, Shihao Wu, Cengiz Öztireli, and Olga Sorkine-Hornung. Differentiable surface splatting for point-based geometry processing. *ACM Transactions on Graphics (TOG)*, 38(6):1–14, 2019. 3