NeVStereo: A NeRF-Driven NVS-Stereo Architecture for High-Fidelity 3D Tasks

Supplementary Material

7. Experimental Setup

7.1. Hardware

All experiments were conducted on a high-performance computing server. The system specifications are as follows:

- **CPU:** Dual Intel(R) Xeon(R) Platinum 8468 CPUs (2 Sockets), providing a total of 192 logical threads.
- **RAM:** 1.0 TiB of system memory.
- **GPU:** 8 × NVIDIA H100 NVL GPUs, each equipped with approximately 94 GB of VRAM.

Unless otherwise specified, the training and inference for each scene were executed on a single GPU.

7.2. Setups

NeRF Backbone Configuration. We adopt ZipNeRF as our neural radiance field backbone. The specific training configurations are:

- **Grid Representation:** We use a multi-resolution hash grid with $L = 10$ levels and a feature dimension of $F = 4$ per level. The hash map size is $T = 2^{19}$, with resolutions spanning from $N_{\min} = 16$ to $N_{\max} = 8192$.
- **Sampling:** A hierarchical sampling strategy is employed with two proposal MLP heads (64 samples each) and one final NerfMLP head (32 samples), totaling 160 samples per ray.
- **Training Parameters:** The model is trained for 25,000 iterations. We use a large batch size of 65,536 rays. The learning rate follows a cosine decay schedule, initializing at 8×10^{-3} and decaying to 1×10^{-3}. The optimizer is Adam with $\beta_1 = 0.9, \beta_2 = 0.99, \epsilon = 10^{-15}$.

Stereo Depth Estimation. For the stereo matching component, we utilize FoundationStereo with the following settings:

- **Model Architecture:** We employ the model with a ViT-Large backbone.
- **Inference Settings:** Input images are processed at their original scale ($scale = 1.0$) and padded to be divisible by 32. We set the number of GRU convex upsampling iterations to $N_{\text{iters}} = 64$ (increased from the default 32) and enable mixed-precision (FP16) computation.

Optimization Hyperparameters. Our Multi-View Confidence-Guided (Mv-CG) framework uses the following weights and scheduling strategies:

- **Initialization & Filtering:** We initialize the depth supervision using the coarse COLMAP poses. For multi-view consistency voting, we use a neighborhood size of $N = 10$ views. To filter out unreliable geometry, only pixels with a confidence score (Eq. 6) greater than **0.2** (implying support from at least $m_{\text{vote}} = 2$ neighboring views) are retained.
- **Loss Weights:** The confidence voting loss weight is set to $\lambda_{\text{vote}} = 0.1$, and the NeRF photometric regularization weight is $\lambda_{\text{nerf}} = 0.01$.
- **NeRF Coupling Schedule:** Camera poses are initialized using COLMAP. To ensure a stable geometric initialization before pose refinement, we perform a warm-up phase of 500 steps where poses remain fixed. Following this warm-up, the NeRF-derived RGB loss is coupled into the dense BA loop every 100 steps to jointly refine camera poses.
- **Annealing Schedule:** We apply a coarse-to-fine annealing strategy for consistency thresholds. The depth consistency threshold τ_{depth} starts at 5% and decays to 1%. The reprojection error threshold τ_{reproj} starts at 2.0 pixels and decays to 1.0 pixel.

Final NeRF Refinement. After the iterative pose and depth optimization, we perform a final refinement of the radiance field. To ensure consistency, the network architecture and general training hyperparameters (e.g., batch size, learning rate schedule, optimizer, and hash grid configuration) remain identical to the initial ZipNeRF backbone described above. The key difference lies in the ray sampling strategy. Instead of relying solely on the proposal networks, we employ our Depth-Guided Gaussian Sampling to concentrate samples around the reconstructed surface. For a ray with fused depth $\hat{d}$, sample points t are drawn from a truncated Gaussian distribution $\mathcal{N}(\hat{d}, \sigma^2)$, where the mean is the depth value $\hat{d}$ and the standard deviation is set to 5% of the depth. ($\sigma = 0.05 \cdot \hat{d}$). We summarize the configuration in Table 7

7.3. Runtime Analysis

We provide a runtime breakdown on a sequence of **200 images** (640×480). All neural steps run on a single H100 GPU. As shown in Table 8, the total time is ∼90 min. While much slower than feed-forward methods, this cost ensures high-fidelity geometry and poses.

8. Limitations and Failure Cases

While NeVStereo demonstrates state-of-the-art performance on standard benchmarks, it relies on certain geometric and photometric assumptions. We identify three primary scenarios where our method may degrade or fail:

Table 7. **Detailed Configuration of NeVStereo.**

Module	Parameter	Value
ZipNeRF	Batch Size	65,536
	Hash Grid Levels	10
	Feature Dim per Level	4
	Max Resolution	8192
	Samples per Ray	$64 + 64 + 32$
	Learning Rate	$0.008 \rightarrow 0.001$
	Distortion Loss Weight	0.005
	Anti-interlevel Loss Weight	0.01
FoundationStereo	Backbone	ViT-Large
	GRU Iterations	64
	Input Scale	1.0
Ours (Mv-CG)	Neighborhood Size (N)	10
	Confidence Threshold	> 0.2
	λ_{vote}	0.1
	λ_{nerf}	0.01
	NeRF Warm-up	500 steps
	Coupling Interval	100 steps
	τ_{depth} (Annealing)	$5\% \rightarrow 1\%$
	τ_{reproj} (Annealing)	$2.0\text{px} \rightarrow 1.0\text{px}$
Final Refinement (Gaussian Sampling)	Base Configuration	Same as ZipNeRF
	Distribution Mean (μ)	Fused Depth $\hat{d}$
	Standard Deviation (σ)	$0.05 \cdot \hat{d}$

Table 8. **Runtime Breakdown.** Measured on a single H100 (200 frames). I/O excluded.

Stage	Detail	Time
1. Initialization		
SfM (COLMAP)	Sparse Init.	4 min
Coarse ZipNeRF	Train (25k iters)	25 min
2. Iterative Optimization		
Rendering (R1)	400 views ($\sim$0.8 FPS)	6 min
Stereo Inf. (R1)	200 pairs ($\sim$1.0 FPS)	4 min
Mv-CG Opt.	Pose & Depth Opt.	15 min
Depth Comp.	Fusion & Hole Fill	1 min
3. Final Refinement		
Refined ZipNeRF	Depth-guided Train	25 min
Rendering (R2)	400 views	6 min
Stereo Inf. (R2)	200 pairs	4 min
TSDF Fusion	Mesh Extraction	1 min
Total	**End-to-End**	**$\sim$90 min**

1. Dependency on Explicit Initialization. Our pipeline currently requires an initial set of camera poses provided by SfM to train the coarse NeRF. Consequently, in scenarios where SfM fails to register images completely, such as scenes with extreme motion blur, low visual overlap, or repetitive structures that cause catastrophic SfM failure, our system cannot bootstrap the optimization process. Unlike some recent end-to-end pose estimators that operate without explicit initialization, our method is strictly tied to the success of the initial SfM stage.

2. Sensitivity to Extreme View Sparsity. As a NeRF-driven architecture, the quality of our depth depends on the fidelity of the synthesized stereo pairs. In extremely sparse view settings (e.g., fewer than 10 views for a room), the NeRF backbone is prone to overfitting, producing remarkable "floating" artifacts or incorrect geometry in every novel view. These rendering failures will disrupt the downstream stereo matching step, creating an erroneous geometry.

3. Large Textureless Regions. Although we utilize a strong stereo depth estimation model, the core principle of stereo matching relies on identifying correspondences via texture features. In scenes dominated by large, featureless surfaces (e.g., white walls), the matcher lacks the necessary cues to infer accurate geometry. As illustrated in Fig. 9, this ambiguity results in severe geometric distortions. While the textured foreground objects are reconstructed faithfully, the textureless wall behind them exhibits significant warping and bending, which is clearly visible in the side and top views. Our Mv-CG mechanism can filter out inconsistent outliers, but it cannot correct large-scale shape distortions where photometric cues are entirely absent.

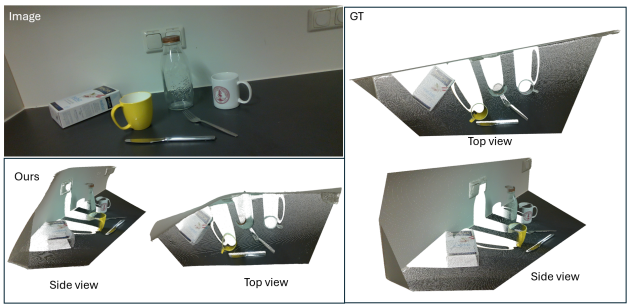

Figure 9. **Failure Case Analysis.** We visualize a scenario with a large textureless white wall, the image is from [17]. **Left** Input RGB image. **Right** The GT depth maps from different views. **Bottom)** Our reconstructed point cloud that demonstrates a strong shape distortion

9. Preliminary Study

9.1. Dataset Splits

To ensure a rigorous evaluation and prevent overfitting to specific scenes, we explicitly divided the datasets into validation and testing sets. The validation sets were used strictly for hyperparameter tuning (e.g., determining confidence thresholds), while all quantitative results reported in the main paper are derived from the held-out test sets.

- **NVIDIA-HOPE:** This dataset consists of 10 sequences indexed from `0000` to `0009`. We selected sequences `0003` and `0006` as the **validation set**. The remaining

NVIDIA-HOPE Dataset					
Metric	**Method**	**Similarity to COLMAP Baseline**			
		KS Stat↓	**Wasserstein**↓	**Median Diff**	**Std Dev** (Method / Ref.)
AbsRel	3DGS based	**0.234**	0.0069	**-0.0022**	**0.0192** / 0.0192
	Nerfacto based	0.339	**0.0040**	-0.0037	0.0105 / 0.0192
$\delta < 1.05$	3DGS based	**0.356**	**0.0183**	**0.0166**	0.0492 / 0.0302
	Nerfacto based	0.458	0.0209	0.0242	0.0403 / 0.0302

Table 9. Statistical analysis on NVIDIA-HOPE shows that 3DGS retains a distribution profile closer to COLMAP compared to Nerfacto, evidenced by lower KS statistics and identical error variance.

8 sequences (`0000`, `0001`, `0002`, `0004`, `0005`, `0007`, `0008`, and `0009`) constitute the **test set**.

- **Redwood RGB-D:** This dataset comprises five distinct indoor environments: *apartment*, *bedroom*, *boardroom*, *loft*, and *lobby*. We utilized the *lobby* scene as the **validation set**. The comparative analysis was performed on the remaining four **test scenes**: *apartment*, *bedroom*, *boardroom*, and *loft*.

9.2. More Statistics

1. NVIDIA-HOPE dataset. Table 9 presents the comparison of the depth distribution characteristics against the COLMAP baseline on the NVIDIA-HOPE dataset. We observe that the 3DGS-based method exhibits consistently higher fidelity to the original SfM distribution across both absrel and $\delta < 5\%$ compared to the Nerfacto-based approach. Specifically, 3DGS achieves significantly lower Kolmogorov-Smirnov (KS) statistics for both absrel (0.234 vs. 0.339) and the accuracy threshold $\delta < 5\%$ (0.356 vs. 0.458), indicating that 3DGS-based NVS-stereo's depth aligns more closely with COLMAP. Most notably, 3DGS perfectly replicates the error dispersion of the baseline, matching COLMAP's standard deviation exactly (0.0192), in contrast to the significantly lower deviation observed in Nerfacto (0.0105). This alignment is further corroborated by smaller median differences in 3DGS for both metrics (-0.0022 for error and 0.0166 for accuracy), confirming that 3DGS serves as a more faithful proxy for the inherent statistical properties of the sparse SfM initialization. The distributions are demonstrated in Fig. 10

However, it is worth noting that while 3DGS is statistically closer, Nerfacto still exhibits moderate similarity rather than complete dissociation. This is likely attributed to the high intrinsic quality of the COLMAP baseline. In such a high-accuracy initialization, the valid depth improvement space is tightly constrained, preventing Nerfacto from diverging drastically from the COLMAP distribution while maintaining depth accuracy.

2. Redwood RGB-D dataset. Table 10 reveals a distinct divergence in method behavior on the Redwood RGB-D dataset, where the initial COLMAP reconstruction is unreliable. Unlike the stable scenario in NVIDIA-HOPE, where both methods maintained reasonable similarity to the baseline, here 3DGS and Nerfacto exhibit opposing statistical characteristics.

The 3DGS-based NVS-stereo still demonstrates a statistical adherence to the noisy COLMAP prior. It retains a remarkably low KS statistic for absrel (0.177) and a near-zero median difference (-0.0028). This indicates that the core distribution of 3DGS output remains "locked" to the erroneous input. However, the massive explosion in standard deviation (0.2310 vs. COLMAP's 0.0261) suggests that while the majority of points mimic the COLMAP distribution, the method suffers from severe outliers, failing to recover correct geometry.

In contrast, the Nerfacto-based NVS-stereo shows a high degree of dissociation from the baseline, with a high KS statistic (0.518) and a significant shift in median error (-0.0240). This "dissimilarity" is positive in this context: it signifies that Nerfacto is actively correcting the geometry rather than overfitting to the sparse SfM points. By diverging from the COLMAP distribution, Nerfacto successfully smooths the noise (lowering Std Dev to 0.0158) and improves accuracy, proving it's not limited by the weak initialization. The distributions are demonstrated in Fig. 11

9.3. Visualization & Analysis

We analyze how rendering artifacts affect downstream stereo matching. Fig. 12 illustrates the significant differences between the two backbones.

In the error heatmaps, **darker colors indicate lower error**, while brighter colors represent higher error. The 3DGS-based view (Top Row) contains high-frequency "floating" artifacts. These structural inconsistencies mislead the stereo matcher. Consequently, they cause large, blocky depth errors, which appear as widespread bright areas in the heatmap.

In contrast, the NeRF-based view (Bottom Row) maintains better geometric continuity. NeRF is not artifact-free; the bright spots in the heatmap confirm that rendering imperfections still lead to depth errors. However, these errors

Redwood RGB-D Dataset (Weak SfM Initialization)					
Metric	**Method**	**Similarity to COLMAP Baseline**			
		KS Stat↓	**Wasserstein**↓	**Median Diff**	**Std Dev** (Method / Ref.)
AbsRel	3DGS based	**0.177**	0.0331	**-0.0028**	0.2310 / 0.0261
	Nerfacto based	0.518	**0.0244**	-0.0240	**0.0158** / 0.0261
$\delta < 1.05$	3DGS based	**0.149**	**0.0530**	**-0.0095**	0.1945 / 0.1501
	Nerfacto based	0.437	0.1123	0.1148	**0.0923** / 0.1501

Table 10. Distribution similarity analysis on Redwood RGB-D. Under weak initialization, 3DGS remains statistically tethered to the COLMAP distribution (lowest KS and Median Diff), implying a failure to correct erroneous priors. In contrast, Nerfacto diverges significantly (high KS), indicating robust geometry correction.

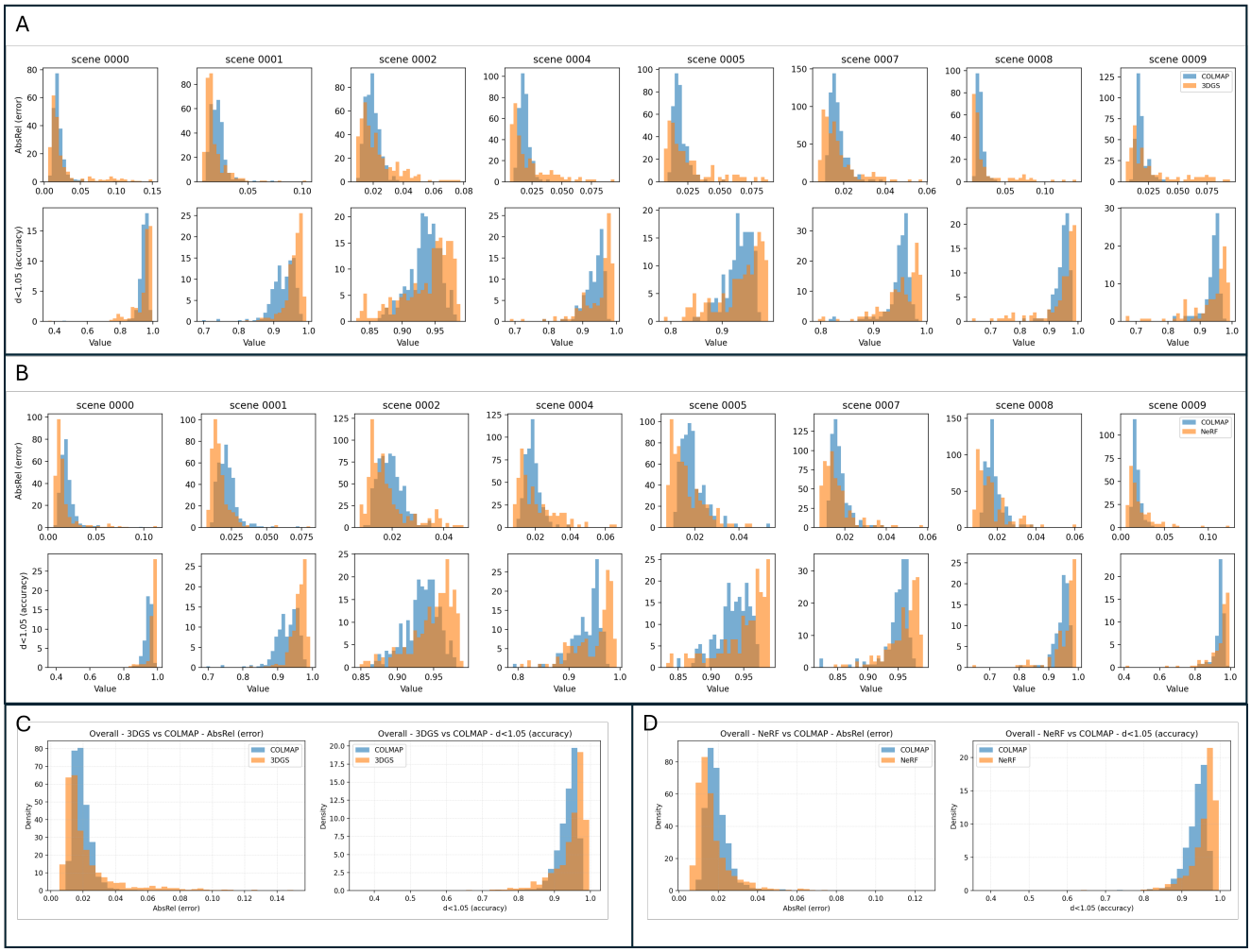

Figure 10. **Statistical comparison of depth distributions on the NVIDIA-HOPE dataset.** We visualize the histograms of AbsRel and accuracy ($\delta < 1.05$) for 3DGS-based and NeRF-based NVS-Stereo against the COLMAP baseline (blue). **(A)** and **(B)** show the per-scene distributions for 3DGS and NeRF, respectively. **(C)** and **(D)** present the aggregated distributions across all test scenes. Observe that the 3DGS distribution (orange in A, C) closely overlaps with the COLMAP baseline, indicating it faithfully replicates the statistical properties of the sparse SfM initialization. In contrast, the NeRF distribution (orange in B, D) deviates more significantly, shifting towards lower error and higher accuracy, demonstrating its ability to regularize geometry beyond the initial sparse points.

are spatially localized and much smaller in area compared to the extensive failures in 3DGS. This proves that modern

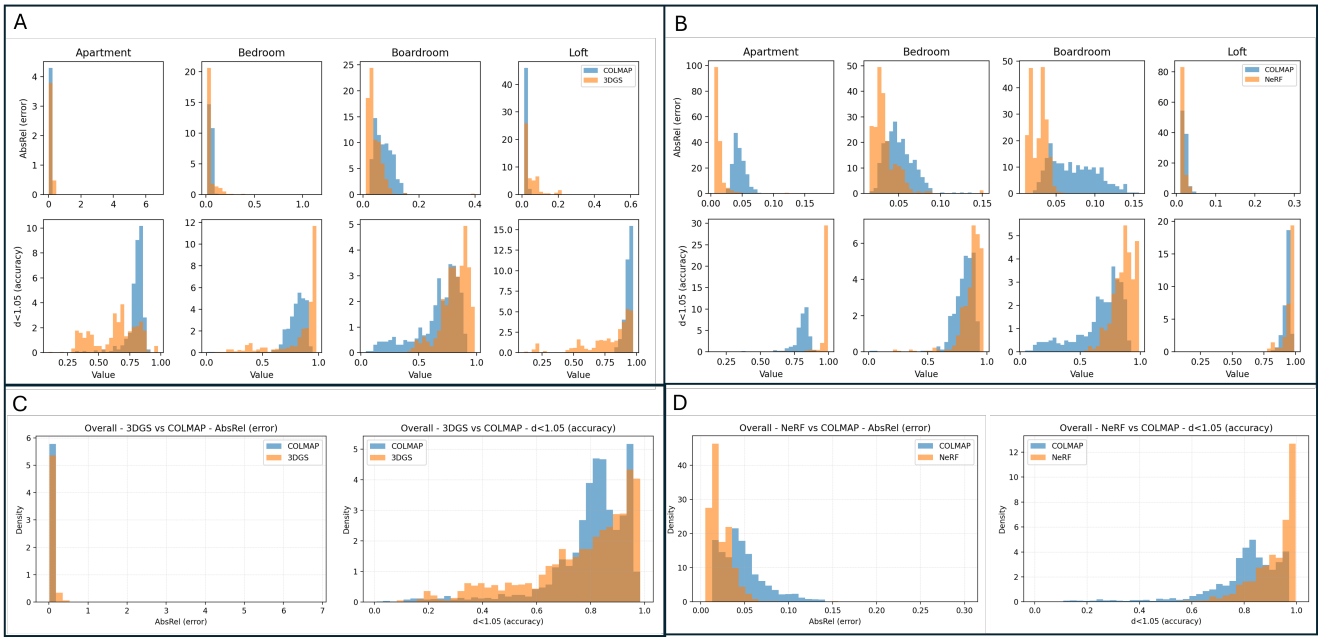

Figure 11. **Statistical comparison of depth distributions on the Redwood RGB-D dataset.** Here, the initial COLMAP reconstruction (blue) contains significant noise and errors. **(A)** and **(C)** illustrate that 3DGS-based NVS-Stereo (orange) statistically adheres to this erroneous prior, exhibiting overlapping distributions with COLMAP. **(B)** and **(D)** demonstrate that NeRF-based NVS-Stereo (orange) successfully dissociates from the noisy baseline. The distribution clearly shifts towards lower error and higher accuracy, proving that NeRF acts as a robust geometric regularizer capable of correcting poor SfM initializations.

stereo estimators are far more robust to NeRF's minor imperfections than to the structural floaters of 3DGS.

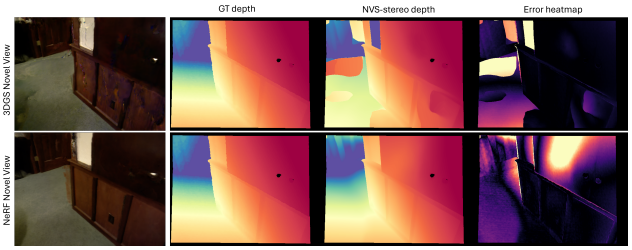

Figure 12. **Visual analysis of rendering artifacts and depth estimation. 3DGS, Top:** The rendered view contains floaters that disrupt stereo correspondence, leading to large-scale depth errors (bright areas in the error heatmap). **NeRF (Nerfacto), Bottom:** The NeRF rendering is geometrically cleaner. Although rendering artifacts may exist, they do not manifest as blocky artifacts. Consequently, the stereo estimator proves robust to NeRF's output, yielding a depth map that closely matches the Ground Truth.

10. Additional Method Info

10.1. TSDF Fusion, Scale Alignment, and Gaussian Ray Sampling.

To mitigate stereo flickering and stabilize loop-closure distillation, we first fuse the final filtered stereo depth D_i^{st} with the final camera poses $\{\mathbf{T}_i\}$ into a truncated signed distance function (TSDF) volume $\mathcal{V}$. The volumetric fusion is defined as

$$\mathcal{V}(\mathbf{x}) = \frac{\sum_i w_i(\mathbf{x})\, \psi\big(D_i^{st}(\pi_i(\mathbf{x})) - z_i(\mathbf{x})\big)}{\sum_i w_i(\mathbf{x})}, \tag{19}$$

where $\psi(\cdot)$ denotes the truncated signed distance, $\pi_i(\cdot)$ projects a 3D point $\mathbf{x}$ to image i, and $z_i(\mathbf{x})$ is its depth in camera coordinates. The fused TSDF is then reprojected to each camera to produce back-projected depths

$$\tilde{D}_i(u,v) = \Pi_z\big(\mathcal{V}^{-1}(\pi_i^{-1}(u,v))\big), \tag{20}$$

where $\Pi_z(\cdot)$ extracts the depth component of the TSDF zero-level surface.

To align scales between the fused geometry and the *initial* ZipNeRF+Stereo depths D_i^{zip}, we compute a per-frame affine rescaling:

$$D_i^{zip}(u,v) \leftarrow \alpha_i\, D_i^{zip}(u,v) + \beta_i, \tag{21}$$

where parameters (α_i, β_i) are estimated via robust regression (e.g., median or Huber) minimizing

$$\min_{\alpha_i,\beta_i} \sum_{(u,v)\in\Omega_i} \rho\Big(\tilde{D}_i(u,v) - (\alpha_i D_i^{zip}(u,v) + \beta_i)\Big)\,.$$

For each pixel (u, v), we define the selected supervisory depth as

$$\hat{d}(u,v) = \begin{cases} \tilde{D}_i(u,v), & \text{if TSDF back-projection is valid,} \\ D_i^{zip}(u,v), & \text{otherwise.} \end{cases} \tag{22}$$

Sampling along each ray follows the truncated Gaussian in Eq. 17, with sample locations generated by Eq. 18. The standard deviation is determined by the validity of the TSDF support:

$$\sigma_d(u,v) = \begin{cases} 0.05\,\hat{d}(u,v), & \text{if } \tilde{D}_i(u,v) \text{ is valid,} \\ 0.25\,\hat{d}(u,v), & \text{otherwise (fallback to } D_i^{zip}\text{).} \end{cases} \tag{23}$$

This adaptive sampling density focuses rays around reliable TSDF geometry while maintaining continuity and differentiability, effectively suppressing stereo-induced flicker and promoting cross-view consistency.

11. Additional Qualitative Results

This section provides additional qualitative comparisons that could not be included in the main paper due to space limitations. We present (1) per-view point cloud visualizations highlighting geometric fidelity across diverse indoor and tabletop scenes, and (2) mesh reconstructions on all remaining MobileBrick subsets to supplement the examples shown in the main paper.

Per-view point cloud comparisons. As a supplement to Fig. 1 in the main paper, Fig. 13 shows single-view point clouds reconstructed by our method and four representative baselines: π^3 [43], VGGT [40], MapAnything [19], and DUSt3R [42]—on the Redwood [27], SCANNET++ [48], and HOPE [37] datasets. Across office, bedroom, loft, and tabletop scenes, our approach produces cleaner geometry, sharper structural boundaries, and noticeably fewer floating points or fragmented artifacts. Compared to the baselines, our reconstructions also exhibit higher geometric accuracy and improved surface consistency.

MobileBrick mesh reconstructions. Figure 14 provides mesh reconstructions for 14 additional MobileBrick [22] subsets not included in the main paper. These visualizations consistently show accurate object geometries and fine structural details across diverse toys, vehicles, aircrafts, and architectural models, serving as an extended qualitative supplement to Fig. 6 in the main paper.

NVS Rendering Results. As a supplement for Fig. 8, Fig. 15 compare our final reconstructed renderings with ZipNeRF [2] on several ScanNet++ scenes. Although ZipNeRF already produces renderings that are visually close to the ground truth, the zoomed-in regions reveal that our method preserves sharper details, cleaner edges, and fewer local artifacts. Red boxes indicate the zoom-in regions from our method, while blue boxes show the corresponding zoom-ins from ZipNeRF.

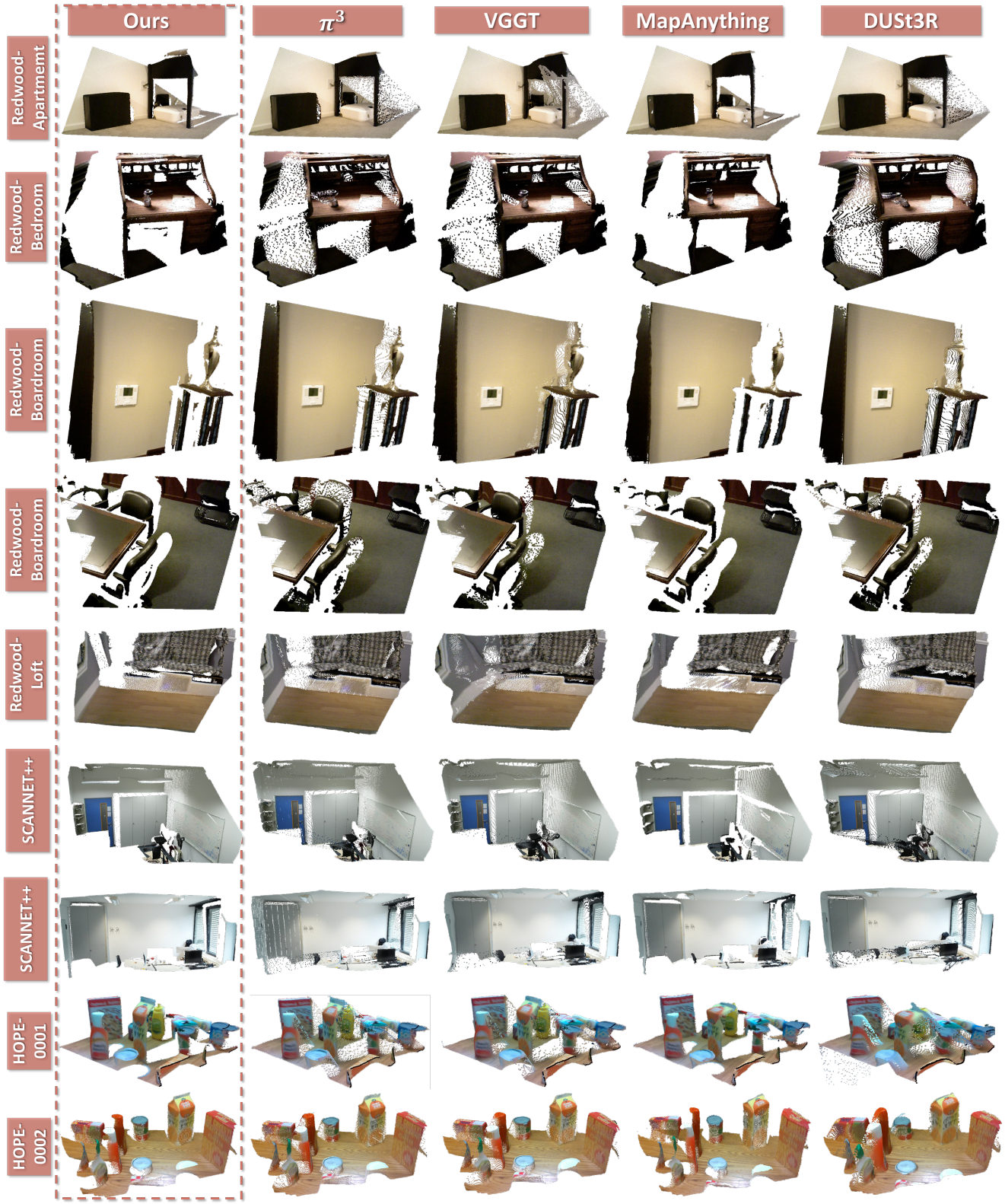

Figure 13. **Additional single-view point cloud comparisons.** Per-view point clouds reconstructed by our method and four baselines (π^3, VGGT, MapAnything, DUSt3R) on the Redwood, SCANNET++, and HOPE datasets. Our approach produces geometry with higher accuracy, fewer floating points, and significantly reduced artifacts across indoor and tabletop scenes, complementing the results of Fig. 1 in the main paper.

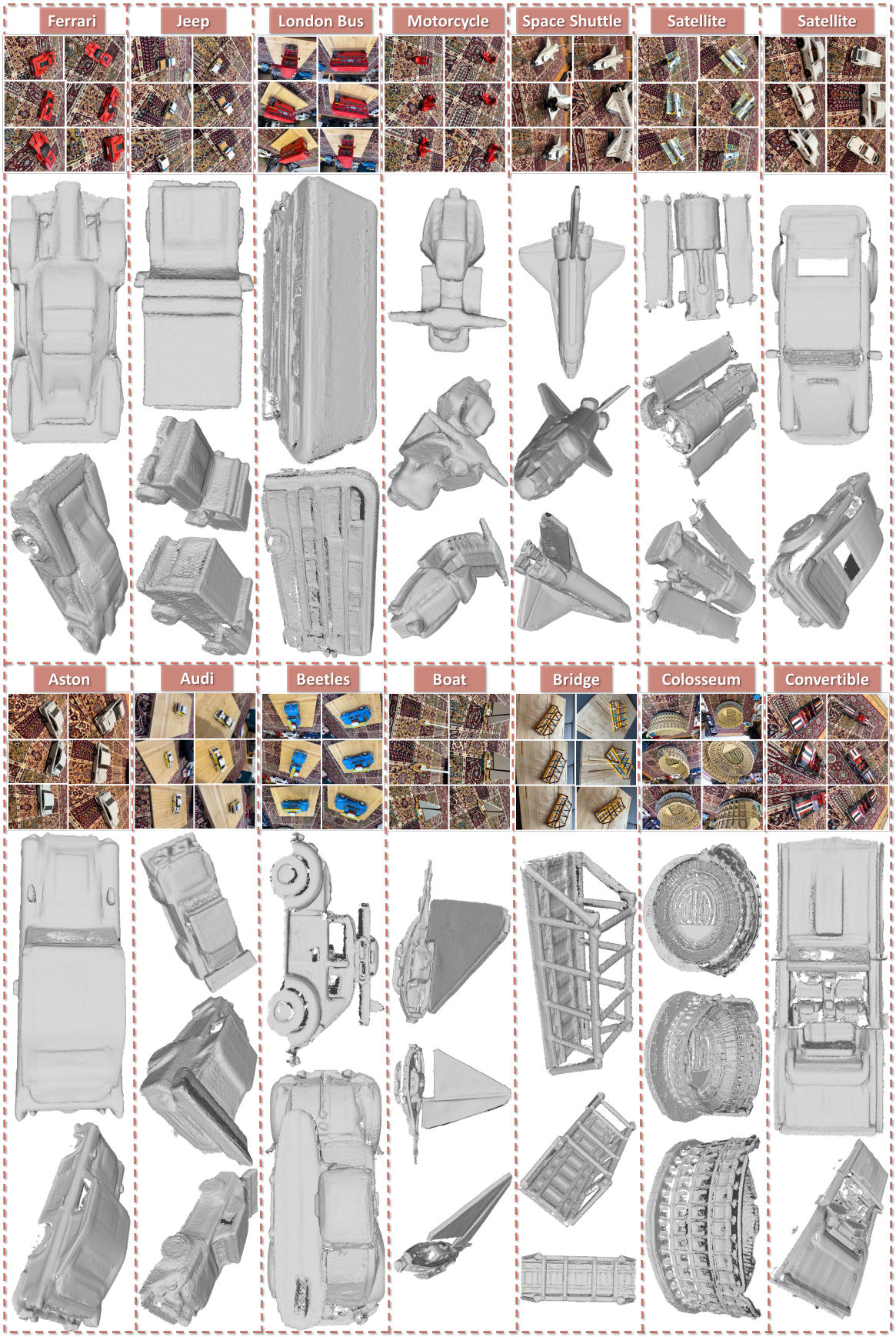

Figure 14. **Additional mesh reconstructions on the MobileBrick dataset.** Mesh visualizations for 14 additional subsets not shown in the main paper. Across vehicles, aircrafts, boats, and architectural models, our method recovers clean shapes and fine geometric structures, providing extended qualitative evidence beyond Fig. 6 in the main paper.

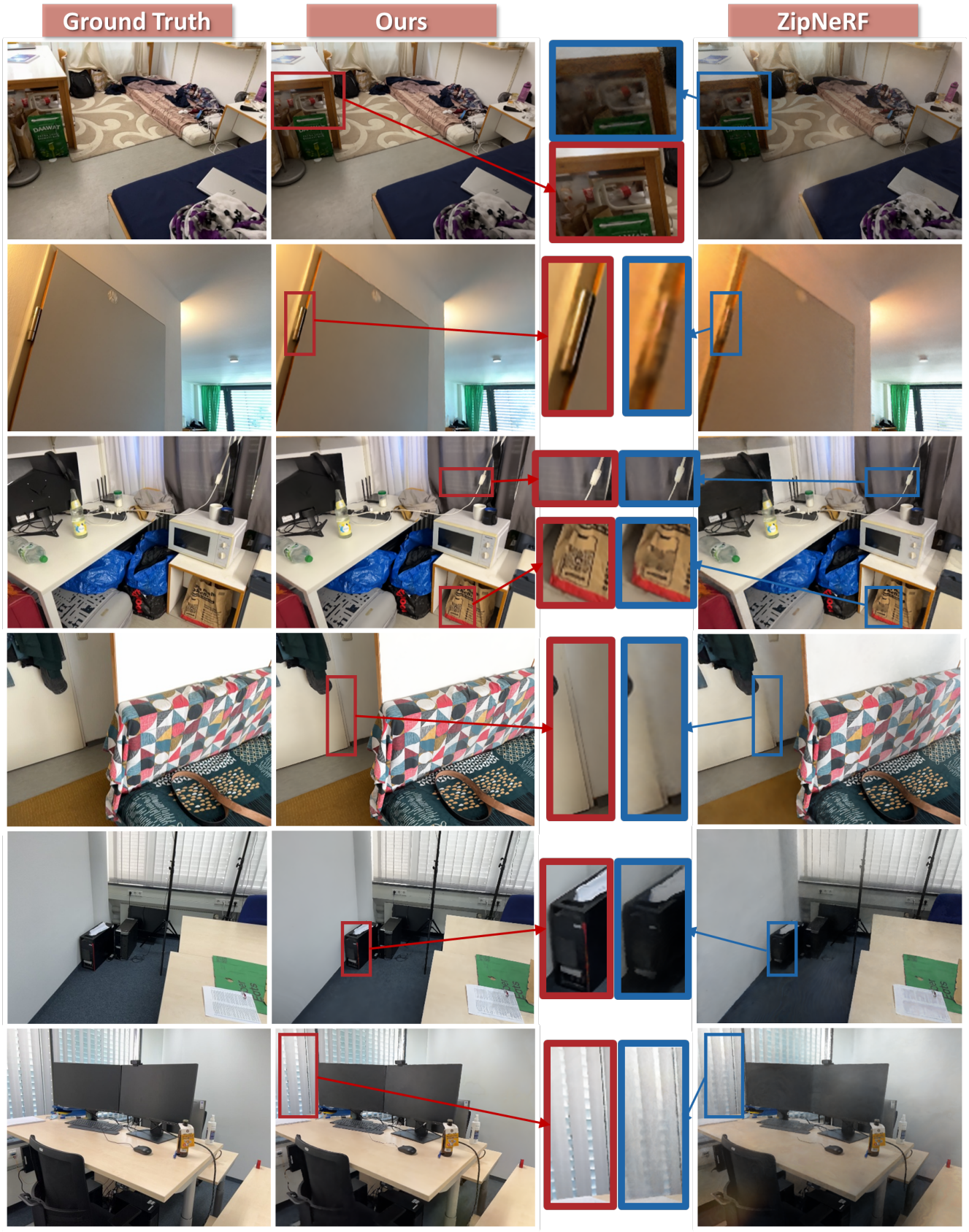

Figure 15. **Additional rendering comparisons on the ScanNet++ dataset.** Despite ZipNeRF producing renderings that are already very close to the ground-truth images, the zoom-in views highlight that our method achieves sharper details and fewer artifacts. Red boxes show zoom-ins from our method, and blue boxes show the corresponding zoom-ins from ZipNeRF.