EDGS: Eliminating Densification for Efficient Convergence of 3DGS

Supplementary Material

A. Implementation details

Evaluation protocol. Following standard practice in 3DGS-based reconstruction, every 8th camera view is used for testing. For the Mip-NeRF360 dataset, we follow the original 3DGS protocol [30] and downsample outdoor scenes by a factor of four and indoor scenes by a factor of two. Other datasets are used at their original resolution.

Initialization. We initialize the scene using up to 180 reference views. For each reference view I_i, we select its two nearest neighbors based on camera-pose proximity and compute dense correspondences using RoMa [12]. Each forward matching pass takes 0.21 s on an NVIDIA A100 GPU. For every correspondence, we compute the triangulated 3D point and evaluate its reprojection error in both participating views. We keep only matches with confidence above $\tau_{\text{corr}} = 0.05$ and reprojection error below $\tau_{\text{proj}} = 0.01$ (in NDC units). From the resulting geometrically consistent set, we sample 20K correspondences per reference view according to our distribution $\mathbf{p}_i$. For each sampled point, we estimate the spherical harmonics coefficients, initialize the color from the corresponding pixel in I_i, and set its initial scale proportional to its distance from the reference camera.

For the default setting (180 reference views, 2NN, 20K correspondences/view), full-scene initialization takes $\sim$ 120 s end-to-end on an A100. Dense matching dominates this cost: 180×2 forward passes at 0.21 s each account for $\approx$ 76 s. Triangulation and SH estimation are run iteratively per view and take $\sim$ 11 s and $\sim$ 15 s, respectively; the remaining time is spent on reprojection-based filtering, data preparation, and splat instantiation. We use a single CPU core for auxiliary preprocessing, and peak GPU memory during initialization is $\sim$ 15 GB.

Spherical harmonics initialization. As outlined in the main manuscript (Sec. 3.5), we estimate spherical harmonics (SH) coefficients for each sampled Gaussian using its available multi-view color observations. We provide additional implementation details here.

Each Gaussian typically has only two usable observations (one from the reference view and one from its nearest neighbor), making the SH estimation problem underdetermined. Despite the limited observations, we formulate the full SH coefficient matrix $\hat{\mathbf{H}}_k \in \mathbb{R}^{16\times3}$ (degree $l = 3$) and initialize all coefficients except the first component (index 0) using the pseudoinverse solution $\mathbf{Y}_k^{+}\mathbf{O}_k$. The first component is initialized directly from the reference-view color at pixel (u_k^i, v_k^i) in I^i, ensuring that the initial appearance is consistent with the reference image while still providing a stable estimate for higher-order coefficients.

Following the standard 3DGS optimization protocol, we progressively unfreeze the SH coefficients during the optimization process to ensure a fair comparison with prior work. The color component and the first four spherical harmonics coefficients are optimized from initialization, with each subsequent coefficient progressively unfrozen every 1,000 iterations.

Methods	12-view			24-view		
	SSIM ↑	PSNR ↑	LPIPS ↓	SSIM ↑	PSNR ↑	LPIPS ↓
Mip-NeRF 360 [2]	0.432	17.73	0.520	0.530	19.78	0.431
RegNeRF [52]	0.437	18.84	0.544	0.546	20.55	0.398
SparseNeRF [66]	0.395	17.44	0.609	0.600	21.13	0.389
3DGS [30]	0.499	17.49	0.431	0.588	19.93	0.401
SparseGS [73]	<u>0.577</u>	**19.37**	<u>0.398</u>	**0.713**	**23.02**	<u>0.290</u>
EDGS + 3DGS	**0.594**	<u>18.96</u>	**0.388**	<u>0.699</u>	<u>22.25</u>	**0.289**

Table A1. Quantitative results on the Mip-NeRF360 [2] dataset under 12- and 24-view training settings. Although EDGS is not designed for sparse-view reconstruction, it performs on par with specialized baselines and in some cases surpasses SparseGS [73] that leverage diffusion-based score distillation losses.

Optimization. After initialization, we employ the standard 3DGS optimization schedule, disabling densification and omitting gradient aggregation for detecting under-reconstructed regions in order to isolate the effect of initialization. All models are trained for the same number of iterations as competing methods (30000 steps) unless explicitly stopped early at 5K or 10K steps (*EDGS + 3DGS 5K* and *EDGS + 3DGS 10K*). All experiments were conducted on an NVIDIA A100 (80 GB). Our method requires at most 15 GB of GPU memory.

Integration with ADC methods. When combining EDGS with adaptive density-control methods, we enable densification but maintain the final Gaussian count by initializing with fewer points. Specifically, we use 140 reference views, 8.5k sampled correspondences per view, and two nearest neighbors, which yields initialization sizes comparable to those produced by the other ADC strategies.

B. Sparse-view setting

We evaluate our method in sparse-view settings following the protocol established by SparseGS [73]. Experiments are conducted on seven scenes from the Mip-NeRF360 dataset, excluding *flowers* and *treehill*, to ensure a fair comparison with prior work. For each scene, we reserve every eighth image as a test view and uniformly sample either 12 or 24 of the remaining images as the training set.

Training schedules match previous standards: 10k itera-

tions for the 12-view setup and 30k iterations for the 24-view setup. In this experiment, we disable densification for EDGS to focus on the effect of adding our initialization. For each reference view, we sample 50,000 correspondences and treat all training views as reference views, selecting the two nearest neighboring views for correspondence matching.

We compare against established baselines for sparse reconstruction, including RegNeRF [52], SparseNeRF [66], the original 3DGS method [30], and the state-of-the-art sparse 3DGS variant, SparseGS [73].

Even without densification, EDGS delivers performance competitive with methods specifically tailored for sparse-view reconstruction. As demonstrated in Tab. A1, the proposed initialization provides reliable geometric alignment and stable optimization, enabling EDGS to match or exceed methods that rely on additional regularization or learned priors [73] in handling limited supervision. These results indicate that a strong correspondence-based initialization alone can substantially improve the quality of 3DGS reconstruction in sparse-view scenarios. This also suggests that densification is not essential for achieving robust performance when the initialization effectively captures scene geometry given only limited number of views.

C. Nearest Neighbors

Ensuring full scene coverage requires sampling from all regions of each reference view. However, a single neighboring view typically overlaps with only a subset of the reference image, meaning that reliable correspondences exist only for those shared regions. To cover the entire reference view with reliable matches, we therefore aggregate correspondences from multiple nearest neighbors. Using a larger number of reference views further ensures that all parts of the scene receive initial splats. In Fig. A1, we visualize the contribution of individual neighbors. The reference image I^i is shown in the top-left. Rows 2–5 show, for each nearest camera j(sorted by distance to the reference view camera), the corresponding ground-truth view (left) and confidence map $\mathbf{c}^{ij}$ (right), indicating which pixels in I^i were matched to that neighbor. Each neighbor covers only a subset of the reference view, which motivates aggregating confidence values on a per-pixel basis. We therefore define the aggregated confidence map for a reference view I_i as

$$\mathbf{c}^i(u,v) = \max_{j \in \mathbb{I}_i} \mathbf{c}^{ij}(u,v),$$

which assigns each pixel (u, v) in I_i the highest correspondence confidence across all its neighboring views $\mathbb{I}_i$. The final aggregated confidence map $\mathbf{c}^i$ (top-right) is used to sample points. The visualization uses the *treehill* scene from Mip-NeRF360 [2].

Increasing the number of neighbors yields more correspondences, but this approach quickly leads to diminishing returns because different neighbors often match largely overlapping regions. In contrast, initialization time grows roughly linearly with the number of neighbors. We therefore obtain a more efficient trade-off by sampling more reference views while restricting each reference image to its 2 nearest neighbors. This provides broad scene coverage without unnecessary computational overhead.

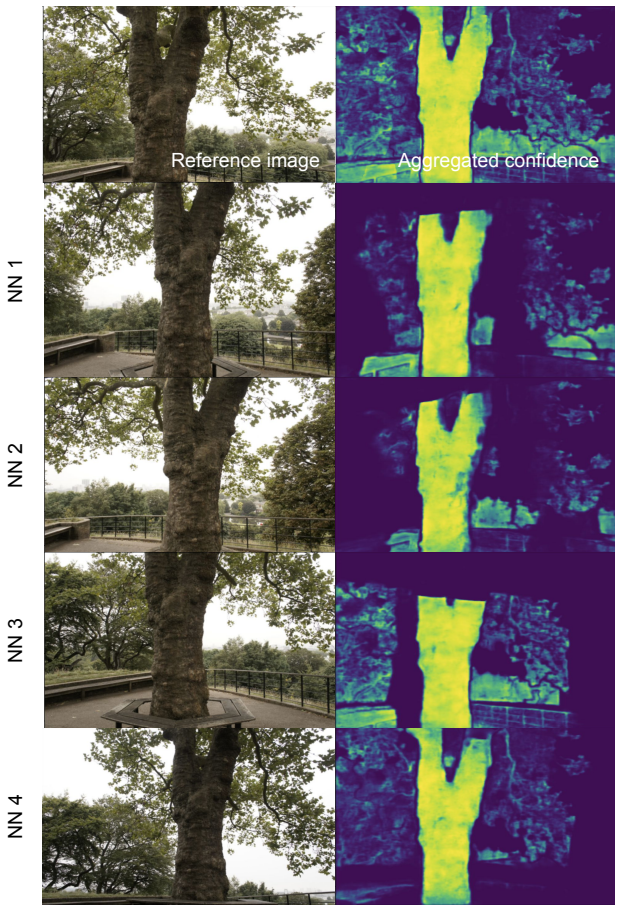

Figure A1. Visualization of correspondence extraction from multiple nearest neighbors for reference image I^i. The top-left picture shows the reference view. Each subsequent row displays a neighboring ground-truth image (left), ordered by camera proximity, and its corresponding matching confidence map $\mathbf{c}^{ij}$ (right). The top-right picture shows the aggregated confidence map $\mathbf{c}^i = \max_{j \in \mathbb{I}_i} \mathbf{c}^{ij}$, formed by combining scores from all neighbors. Aggregation provides denser and more uniform coverage of the reference frame. Example shown for the *treehill* scene from Mip-NeRF360.

D. Gaussians motion through optimization

We provide videos in the supplementary material (folder `sec_D_gaussians_motion/`) that visualize how Gaussian parameters evolve during optimization. This experiment highlights that, thanks to our more accurate initialization, EDGS begins much closer to the final solution, leading to substantially smaller parameter updates and shorter optimiza-

tion trajectories. To visualize splat motion, we record the positions and colors of all Gaussians at every iteration. After training, we identify the final set of Gaussians and reconstruct their trajectories by tracing their position and color histories backward through the optimization. For 3DGS, which performs densification, we additionally follow each split or cloned Gaussian back to its corresponding "parent" in order to maintain consistent trajectories from the first iteration; EDGS does not require this step. We illustrate this analysis on the *flowers* and *stump* scenes from the Mip-NeRF360 dataset, where visualizations confirm that our splats undergo far fewer adjustments and reach high-quality reconstructions much earlier in training.

E. Initialization with denser COLMAP

To further test whether EDGS' improvements could be attributed to simply starting from a *stronger* or *denser* prior, we include additional initialization experiments for completeness. The goal is to evaluate whether "more Gaussians" or alternative priors can match EDGS when densification is removed. In Tab. A2, we evaluate a naïve "denser COLMAP" baseline by duplicating the COLMAP points $10\times$ and $50\times$ and adding noise $\epsilon \sim \mathcal{N}(0, \sigma)$ with $\sigma \in \{10^{-3}, 10^{-1}\}$ to the 3D coordinates to avoid exact overlap. This substantially increases training time but does not improve reconstruction quality, showing that "more Gaussians" alone is insufficient; EDGS' gains come from a geometrically accurate initialization rather than raw point count.

Mip-NeRF 360	Init Duplication	σ	SSIM ↑	PSNR ↑	LPIPS ↓	Train time	#G (10^6)
3DGS [31]	$\times 1$	-	0.816	27.49	0.215	26 m	2.8
	$\times 10$	10^{-1}	0.815	27.25	0.209	31 m	3.0
	$\times 10$	10^{-3}	0.815	27.22	0.211	32 m	3.1
	$\times 50$	10^{-1}	0.812	27.10	0.208	44 m	3.4
	$\times 50$	10^{-3}	0.814	27.07	0.209	47 m	3.9

Table A2. Adding more Gaussians to the COLMAP initialization is not enough.

F. EDGS Initialization vs. 3DGS Convergence

To better isolate the effect of initialization, we compare the spatial splat distributions produced by EDGS and standard 3DGS. Our key finding is that EDGS starts from a configuration that already closely matches the distribution vanilla 3DGS reaches only after optimization has converged. We visualize this effect from real camera viewpoints while reducing splat scales so that individual Gaussians become apparent; see Fig. A2. The comparison shows that EDGS places splats in the regions necessary to represent the final scene structure from the start, whereas standard 3DGS begins from a sparse, coarse scaffold and must iteratively densify it throughout training. This helps explain the stronger convergence behavior of EDGS.

We further analyze this phenomenon at the scene level using a voxelized representation; see Fig. Fig. A5. Specifically, we discretize the scene into a regular voxel grid and compute the splat density in each voxel. The projected voxel distributions show that the final spatial density reached by standard 3DGS is already closely approximated by EDGS at initialization. This indicates that the densification mechanism of standard 3DGS primarily compensates for weak initial support, whereas EDGS starts from a scene-wide support that is already close to the converged solution.

We additionally visualize the average voxel color to make the projected structure easier to interpret for the *bicycle* scene from MipNeRF-360 [2]. The results show that standard 3DGS assigns sufficient splats to prominent nearby structures, such as the bicycle and the bench, but leaves distant regions largely unsupported until the optimization process gradually introduces them. In contrast, EDGS covers both near and far regions from initialization, yielding a scene-wide splat distribution that already resembles the final converged state.

G. Additional visual results

Initializations. In Fig. A6, we compare our initialization against the standard SfM-based initialization used in 3DGS. The latter produces sparse and uneven point clouds, often leaving large background regions underrepresented. In contrast, our approach initializes splats densely across the entire scene. Although the resulting initialization may appear noisy, the optimization quickly suppresses erroneous splats and retains only those consistent with target views. This dense starting point ensures that all regions receive early supervision, avoiding the multiple densification rounds required by the standard 3DGS pipeline to reach a comparable level of coverage and reconstruction quality.

Extreme Viewpoint Rendering. EDGS effectively handles extreme viewpoint variations, outperforming the baseline when rendering from camera angles far outside the training set. As shown in Fig. A3, our dense initialization prevents the need for stretching small Gaussians to compensate for pixel loss at a distance, resulting in a more stable and accurate reconstruction. As visualized for *garden* scene from the Mip-NeRF360 dataset, our method avoids large Gaussians and exhibits less noise compared to the competing approach.

Robustness to noise in initialization. In Fig. A4, we visualize the effect of adding synthetic noise to our initialization. This complements the quantitative robustness analysis in the main manuscript and provides an intuitive sense of how strongly the initialization must be perturbed before reconstruction quality degrades. As the figure shows, only substantial corruption produces visibly degraded initial splat configurations. Even for noticeable noise levels such as $\sigma = 0.05$ (see the visualizations in Fig. A4 and the corre-

Figure A2. **Reduced-scale splat renderings from training views.** We visualize the initialized and converged splats of baseline 3DGS and EDGS using the standard Gaussian renderer while reducing splat scales so that individual Gaussians become visible. EDGS already initializes the scene with the splats required for the final reconstruction, whereas baseline 3DGS starts from a much coarser scaffold and only reaches a similar distribution after optimization and densification. This highlights that EDGS begins near the converged spatial distribution that standard 3DGS must discover iteratively.

Figure A3. Extreme viewpoint rendering. EDGS (right) better preserves details and reduces stretched Gaussians when rendering from viewpoints far outside the training set compared to the 3DGS (left). This results in a more consistent distribution and improved quality, especially in challenging regions like the building and flower pot.

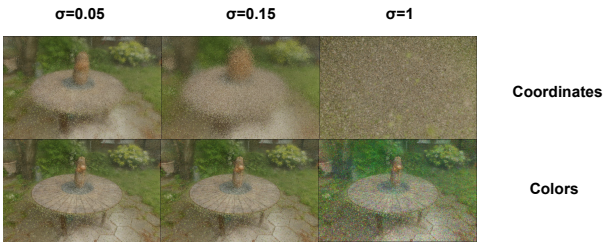

Figure A4. The impact of noise on initialization quality. The first row shows the effect of adding noise to coordinates, while the bottom row demonstrates the effect of adding noise to color values.

sponding scores in Fig. 7), EDGS reliably recovers accurate reconstructions. Only very large perturbations, typically $\sigma > 0.15$, lead to significant degradation in performance.

Qualitative comparison. Full-resolution versions of the renders shown in the main paper are provided in Figs. A7 to A9. For clearer comparison in Fig. A9, we also include renderings for the original 3DGS method.

Video results for front-facing scenes. Our method is also compatible with front-facing scenes. We use sequences with primarily forward-facing camera orientations, with little variation of viewpoint. From each scene, we extract 24 frames and first process them with COLMAP to recover camera intrinsics and extrinsics. We compare the original 3DGS pipeline with EDGS on the same data. For each scene, we visualize three stages of our approach: the initial splat placement, the intermediate fitting stage, and the final rendering from multiple viewpoints. We use 24 reference frames and 10,000 correspondences per reference. The results can be found in the folder `sec_E_front_facing_scenes/` of the supplementary material.

Video results for synthetic data. We also visualize performance on the Synthetic dataset [48]. We compare the original 3DGS pipeline with EDGS on the same data. Both methods are allocated equal runtime budgets, allowing for

a direct comparison at matching optimization time. In both settings, we align the optimization timeline to ensure comparability. EDGS consistently achieves high-quality reconstructions faster than 3DGS, thanks to its rich and dense initialization, which provides the necessary detail from the very beginning. See folder `sec_E_synthetic_scenes/` in the supplementary material.

H. Per-scene results

We provide a more detailed evaluation of *EDGS + 3DGS* from Tab. 1, *EDGS +3DGS 5K* from Tab. 2 and *3DGS MCMC with EDGS init* from Tab. 3. We include per-scene scores for these models in Tabs. A3 to A8. Note that densification is disabled for the first three models; for the final model, we begin with fewer points and apply the adaptive densification strategy from 3DGS-MCMC.

I. Notation

To simplify the understanding of the paper, we include a table of notation Tab. A9. This table provides a concise summary of the key symbols and terms used throughout the paper, along with their definitions.

EDGS + 3DGS	Truck	Train	Dr Johnson	Playroom
SSIM	0.898	0.837	0.900	0.907
PSNR	26.16	22.39	29.50	30.12
LPIPS	0.091	0.172	0.233	0.213
# Gaussians	1.6	1.1	1.5	1.7
Time in minutes	25	21	30	30

Table A3. Per-scene quantitative results on the Tanks & Temples and Deep Blending subsets.

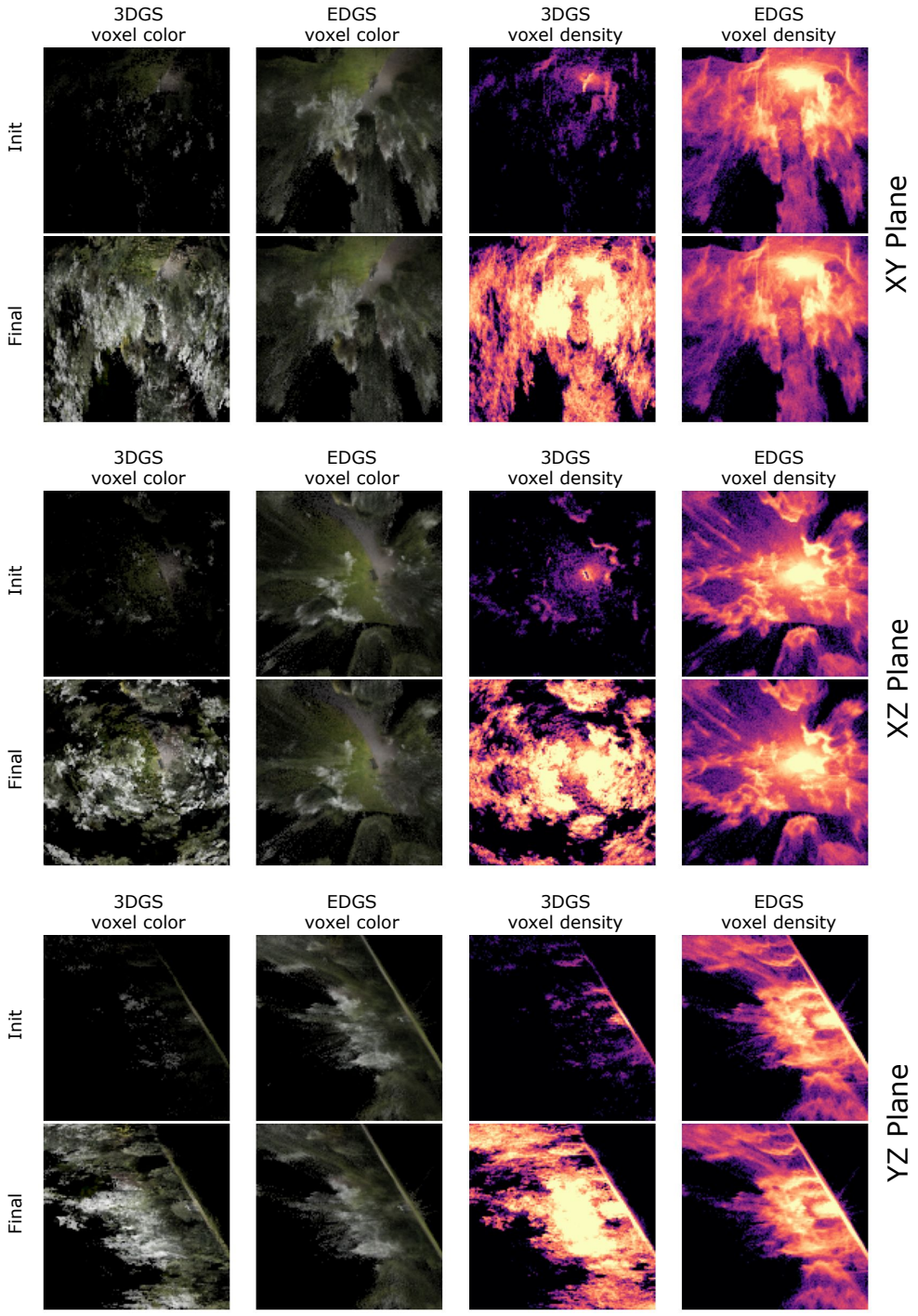

Figure A5. **Projected voxel distributions on the *bicycle* scene from MipNeRF-360 [2].** We voxelize the scene and visualize projected voxel color and projected voxel density for baseline 3DGS and EDGS on the XY, XZ, and YZ planes, at initialization and after convergence. EDGS starts from a scene-wide splat distribution that is already close to the converged distribution of standard 3DGS. In contrast, standard 3DGS initially covers mainly prominent nearby structures, such as the bicycle and bench, and only later expands to distant regions through densification. The average voxel color makes the projected structure easier to relate to the underlying scene content.

Figure A6. Visual comparison of initialization methods on the *stump* scene from the Mip-NeRF360 dataset [2]. The left image represents the ground truth. The middle image shows the traditional 3DGS approach initialization with Structure-from-Motion (SfM) [61]. The right image illustrates initialization with our method using matchings. Despite a noisy appearance at initialization, our model can jointly optimize all the Gaussians and achieve better reconstruction quality.

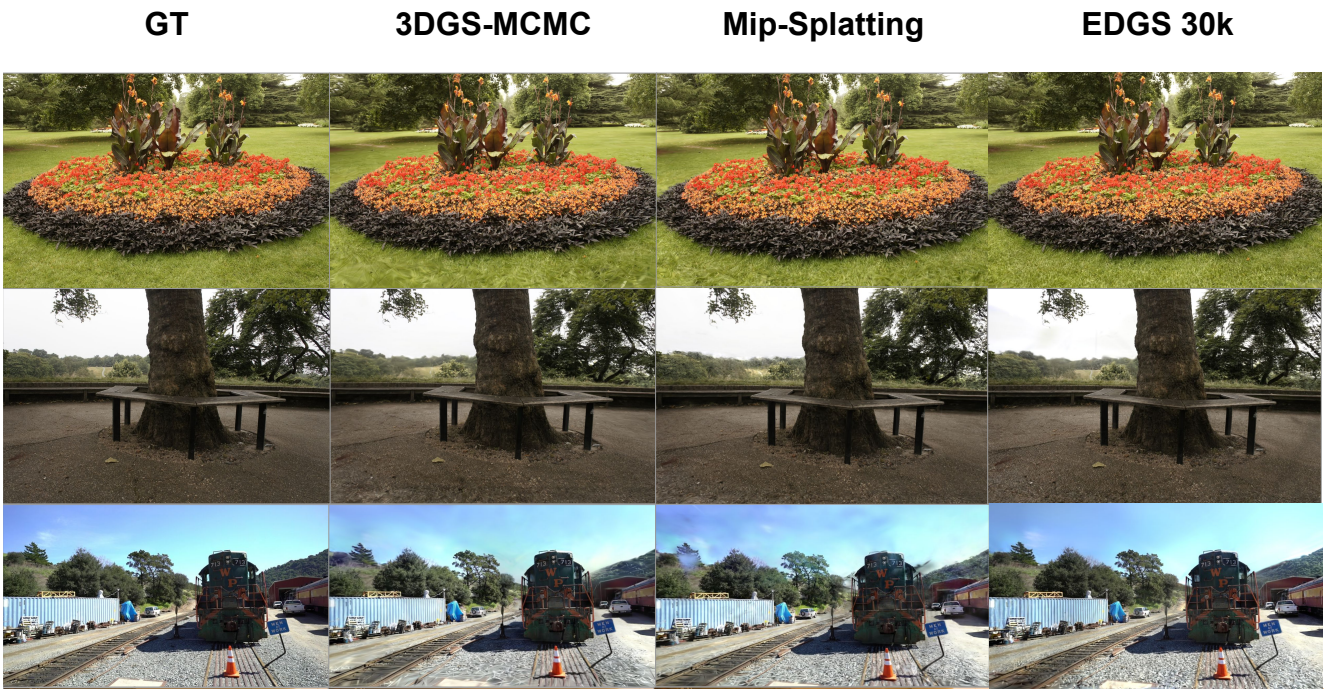

Figure A7. Additional qualitative results are presented for the scenes *treehill*, *flowers* and *train*. For clarity, areas of interest have been zoomed in Fig. 3. These results are best viewed digitally for optimal detail.

	bicycle	flowers	garden	stump	treehill	room	counter	kitchen	bonsai
EDGS + 3DGS 5K	0.760	0.619	0.853	0.789	0.654	0.941	0.912	0.943	0.950
EDGS + 3DGS	0.792	0.641	0.876	0.783	0.655	0.954	0.931	0.955	0.962
3DGS-MCMC + EDGS Init	0.815	0.664	0.891	0.815	0.666	0.941	0.928	0.943	0.959

Table A4. Per-scene quantitative results (SSIM) on the Mip-NeRF360.

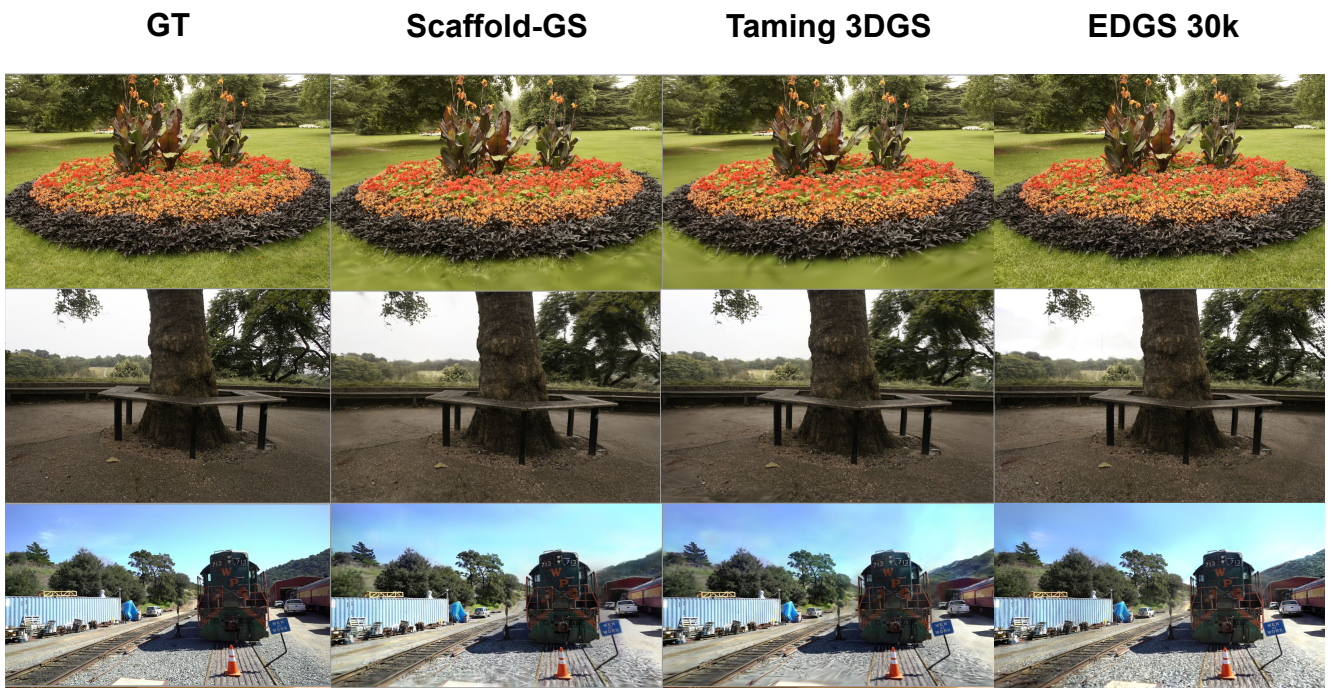

Figure A8. Additional qualitative results are presented for the scenes *treehill*, *flowers* and *train*. For clarity, areas of interest have been zoomed in Fig. 3. These results are best viewed digitally for optimal detail.

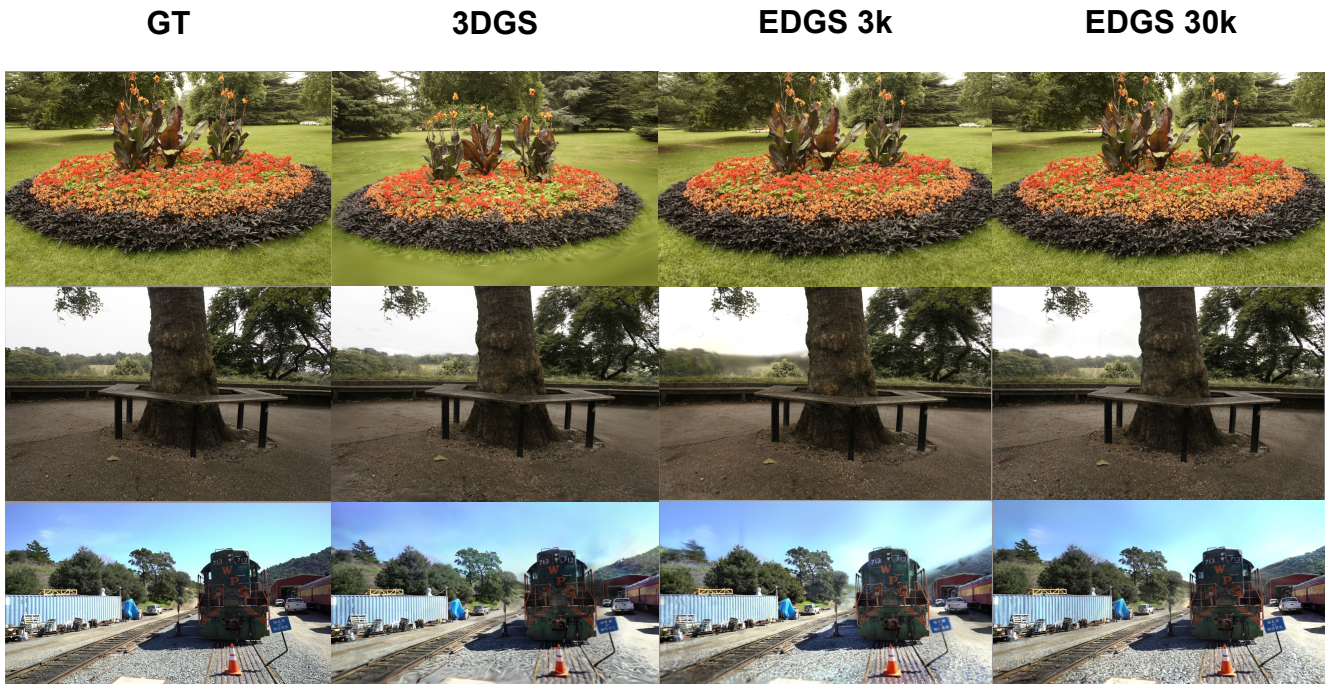

Figure A9. Additional qualitative results are presented for the scenes *treehill*, *flowers* and *train*. For clarity, areas of interest have been zoomed in Fig. 3. These results are best viewed digitally for optimal detail.

	bicycle	flowers	garden	stump	treehill	room	counter	kitchen	bonsai
EDGS + 3DGS 5K	24.58	21.37	26.75	26.78	22.32	30.63	27.91	30.06	30.96
EDGS + 3DGS	25.39	21.57	27.67	26.67	22.47	32.87	29.62	32.99	32.96
3DGS-MCMC + EDGS Init	26.14	21.85	28.39	27.42	22.8	32.47	29.62	32.47	33.41

Table A5. Per-scene quantitative results (PSNR) on the Mip-NeRF360.

	bicycle	flowers	garden	stump	treehill	room	counter	kitchen	bonsai
EDGS + 3DGS 5K	0.203	0.310	0.120	0.203	0.278	0.110	0.113	0.073	0.085
EDGS + 3DGS	0.161	0.267	0.095	0.192	0.252	0.089	0.088	0.059	0.070
3DGS-MCMC + EDGS Init	0.145	0.242	0.084	0.165	0.239	0.165	0.149	0.1	0.144

Table A6. Per-scene quantitative results (LPIPS) on the Mip-NeRF360.

	bicycle	flowers	garden	stump	treehill	room	counter	kitchen	bonsai
EDGS + 3DGS 5K	2.8	2.5	2.8	1.9	2.2	2.9	2.8	3.0	2.6
EDGS + 3DGS	2.3	2.2	2.5	1.8	2.0	1.3	1.7	1.7	1.3
3DGS-MCMC + EDGS Init	5.9	3.7	5.2	4.8	3.6	1.5	1.3	1.8	1.4

Table A7. Per-scene quantitative results (millions of Gaussians $\#G$) on the Mip-NeRF360.

	bicycle	flowers	garden	stump	treehill	room	counter	kitchen	bonsai
EDGS + 3DGS 5K	9	8	9	7	8	7	7	8	7
EDGS + 3DGS	31	30	34	27	31	20	23	25	22
3DGS-MCMC + EDGS Init	32	22	29	26	21	12	12	14	12

Table A8. Per-scene quantitative results (time in minutes) on the Mip-NeRF360.

Table A9. Notation

Notation	Description
$\mathbb{G}$	Set of 3D Gaussians representing the scene
$\boldsymbol{g}_i$	i-th Gaussian in $\mathbb{G}$, with parameters $\{\boldsymbol{g}_i^x, \boldsymbol{\Sigma}_i, \boldsymbol{g}_i^c, \boldsymbol{g}_i^\alpha\}$
$\boldsymbol{g}_i^x \in \mathbb{R}^3$	3D center of Gaussian i
$\boldsymbol{\Sigma}_i \in \mathbb{R}^7$	Encoded covariance (shape) of Gaussian i
$\boldsymbol{g}_i^c \in \mathbb{R}^3$	RGB color of Gaussian i
$\boldsymbol{g}_i^\alpha \in \mathbb{R}$	Opacity of Gaussian i
p	Pixel location in the rendered image
$C(p)$	Rendered color at pixel p
$(\boldsymbol{p}' - \boldsymbol{g}_i^x)$	Shortest distance between the pixel projection line and $\boldsymbol{g}_i^x$
$\boldsymbol{\sigma}_i(p)$	Contribution of Gaussian i to pixel p
$\boldsymbol{R}_i,\ \boldsymbol{S}_i$	Rotation and scaling for $\boldsymbol{\Sigma}_i$
I^i	Reference image
$\mathbb{I} = \{I^j\}$	Set of neighboring images for I^i
$\boldsymbol{P}^i \in \mathbb{R}^{3\times4}$	Projection matrix of camera i
$\mathcal{M}$	Pretrained dense matching network
$\mathcal{W}^{i\to j} \in \mathbb{R}^{2\times H\times W}$	Warp field from I^i to I^j
$\mathbf{c}^{ij} \in \mathbb{R}^{H\times W}$	Confidence of correspondences between I^i and I^j
$(u_k^i, v_k^i), (u_k^j, v_k^j)$	Matched pixel coordinates in I^i and I^j
$\boldsymbol{g}_k^x \in \mathbb{R}^3$	3D position of the k-th new Gaussian (via triangulation)
w_k^i, w_k^j	Homogeneous-scale factors in projection equations
$\pi(\boldsymbol{P}, \cdot)$	Projection with camera matrix $\boldsymbol{P}$
ε_k^i	reprojection error in the reference image I^i for $\boldsymbol{g}_k$
τ_{corr}	Confidence threshold for sampling 2D correspondences
τ_{proj}	Threshold for reprojection error
$\mathbf{p}_{\text{corr}}^{ij}(u, v)$	Uniform sampling distribution over $\{(u_i^k, v_i^k) \mid \mathbf{c}^{ij}(u, v) > \tau_{corr}\}$
$\mathbf{p}_{\text{proj}}^{ij}(u, v)$	Uniform sampling distribution over $\{(u_i^k, v_i^k) \mid \varepsilon_k^{ij} < \tau_{\text{proj}}\}$
$\mathbf{p}^i(k)$	Combined sampling distribution for image I^i
$\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathbb{R}^3$	View directions
$\mathbf{Y}_k \in \mathbb{R}^{n\times16}$	Spherical-harmonic basis evaluated for n views
$\mathbf{O}_k \in \mathbb{R}^{n\times3}$	Observed RGB colors of splat k in n views
$\hat{\mathbf{H}}_k \in \mathbb{R}^{16\times3}$	Fitted spherical-harmonic coefficients