STRinGS: Selective Text Refinement in Gaussian Splatting

Abhinav Raundhal[*] Gaurav Behera[*]
P. J. Narayanan Ravi Kiran Sarvadevabhatla Makarand Tapaswi
CVIT, IIIT Hyderabad, India
STRinGS-official.github.io

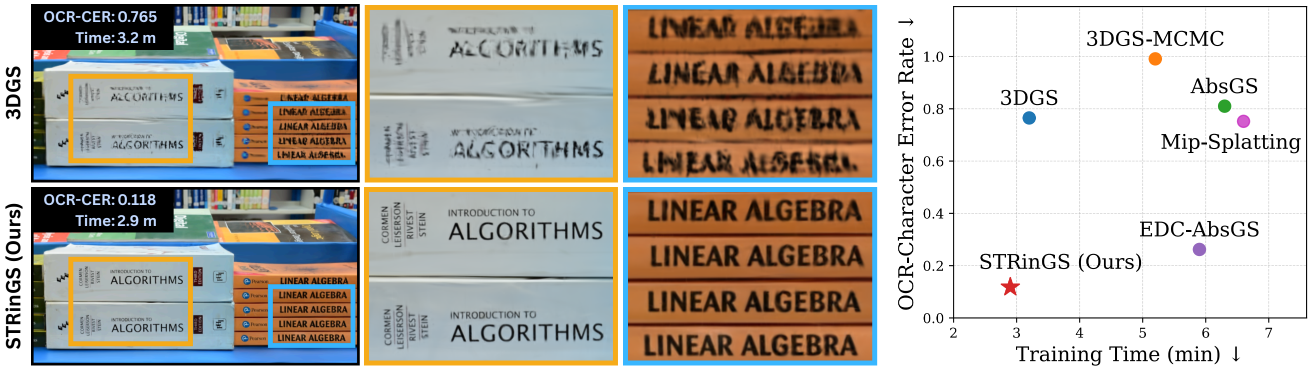


Figure 1. Qualitative and quantitative comparison of Gaussian Splatting methods on text reconstruction at 7K iterations. **Left:** On a novel view from the *Shelf* dataset that features library books on a shelf, our approach STRinGS (bottom) produces sharper and readable text as compared to vanilla 3DGS (top). **Right:** We quantify text reconstruction using Character Error Rate (CER) used in Optical Character Recognition (OCR). The accompanying scatter plot presents readability (CER, lower is better) *vs.* training time. STRinGS achieves the best performance both in terms of lowest error and fastest training time.

Abstract

Text as signs, labels, or instructions is a critical element of real-world scenes as they can convey important contextual information. 3D representations such as 3D Gaussian Splatting (3DGS) struggle to preserve fine-grained text details, while achieving high visual fidelity. Small errors in textual element reconstruction can lead to significant semantic loss. We propose STRinGS, a text-aware, selective refinement framework to address this issue for 3DGS reconstruction. Our method treats text and non-text regions separately, refining text regions first and merging them with non-text regions later for full-scene optimization. STRinGS produces sharp, readable text even in challenging configurations. We introduce a text readability measure OCR Character Error Rate (CER) to evaluate the efficacy on text regions. STRinGS results in a 63.6% relative improvement over 3DGS at just 7K iterations. We also introduce a curated dataset STRinGS-360 with diverse text scenarios to evaluate text readability in 3D reconstruction. Our method and dataset together push the boundaries of 3D scene understanding in text-rich environments, paving the way for more robust text-aware reconstruction methods.

1. Introduction

Capturing 3D scenes from multi-view images for reconstruction and novel view generation is an important problem with applications in mixed reality, robotics, entertainment, archaeology and beyond. Early methods that used explicit geometry [22] were tedious. After this, neural scene representations such as NeRF (Neural Radiance Fields) and its variants [2, 20, 21] dominated the field. More recently, 3D Gaussian Splatting (3DGS) [14] was proposed that uses a geometry-neural hybrid representation. 3DGS also achieved real-time novel-view rendering with state-of-the-art visual fidelity.

3DGS represents scenes using 3D Gaussians and progressively optimizes them, using a coarse-to-fine strategy. This strategy often struggles with high-frequency details such as in fine textured regions and text present in the scene. In particular, many real-world scenes contain text in different ways that are useful for downstream applications. For example, in autonomous navigation, text is essential for interpreting road signs and waypoint recognition, while in VR, clear text improves user experience, and in robotics, it aids object identification and manipulation. Fig. 1 (top) shows the low quality of text reconstructed using 3DGS.

[*] Equal contribution

Can 3DGS be given a pair of reading glasses to enhance visual quality and readability of text regions in the scene? We address this problem in this paper. We present Selective Text Refinement in Gaussian Splatting (STRinGS), a novel framework for improving text readability in 3DGS reconstructions. Prior related approaches attempted to enhance high-frequency regions [6, 31] or improve texture detail [4, 23, 29]. STRinGS identifies text regions and selectively refines them following a two-phase strategy (Sec. 4): (i) Phase 1 isolates text regions and selectively reconstructs them; and (ii) Phase 2 performs a global scene refinement that maintains background fidelity while preserving improved text quality.

Standard 3D reconstruction datasets [1, 10, 13, 17, 18] contain sparse or no text, limiting their use for evaluating our approach. We introduce STRinGS-360, a curated dataset of *five* text-rich 3D scenes (Sec. 3) to address this. Traditional image fidelity based evaluation metrics (*e.g.* PSNR) are also insufficient to evaluate text readability. We introduce OCR Character Error Rate (OCR-CER) as a text readability measure to compare rendered and ground-truth images using a standard Optical Character Recognizer [5]. STRinGS achieves an average of 23.0% relative improvement in OCR-CER over standard 3DGS [14] at 30K iterations and 63.6% relative improvement in OCR-CER at 7K training iterations. Fig. 1 shows the qualitative and quantitative improvement in text readability for a novel view at 7K iterations of training with STRinGS.

The key contributions of our work are given below.

1. We propose STRinGS, the first framework for explicit text refinement in 3DGS, enabling accurate and readable text in rendered novel views.
2. We introduce STRinGS-360, a curated benchmark to evaluate 3D reconstruction methods on text-rich scenes and propose OCR-CER to quantify text readability.
3. We demonstrate that STRinGS enables superior text readability without compromising image quality compared to existing high-frequency enhancement or densification strategies. Furthermore, this is achieved in early stages of training, a critical requirement for time-constrained applications.

2. Related Work

Traditional 3D reconstruction uses Structure-from-Motion (SfM) [25] and Multi-View Stereo (MVS) [26] pipelines to recover camera poses and sparse point clouds from input images. Neural Radiance Fields (NeRFs) [9, 20] from the last few years are a paradigm shift as they represent scenes as volumetric fields using MLPs, enabling photo-realistic novel view synthesis at the cost of slow training. While methods like Instant-NGP [21] improve rendering speed, real-time rendering remains challenging. 3D Gaussian Splatting (3DGS) [14] addresses this by adopting anisotropic 3D Gaussians to represent 3D scenes that enable fast differentiable rasterization. However, 3DGS struggles to preserve high-frequency details, as the coarse-to-fine optimization favors global fidelity over local structure.

3DGS improvements. Recent works extend 3DGS to improve overall scene reconstruction quality and address these limitations. Mip-Splatting [32] tackles aliasing and scale inconsistencies by introducing filters that make 3DGS more robust across zoom levels. 3DGS-MCMC [15] introduces a sampling-based formulation to improve Gaussian initialization, while AbsGS [31] addresses the over-reconstruction of fine structures by revising the gradient-based densification strategy. Mini-Splatting [7, 8] proposes guided densification and simplification pipelines that maintain scene fidelity with fewer primitives. Efficient Density Control (EDC) [6] is a plug-and-play module that enhances various 3DGS variants [7, 19, 31] by incorporating targeted pruning and splitting operations to improve scene fidelity and efficiency. Several other approaches densify Gaussians across the scene based on visibility, reconstruction error, or color cues to improve fidelity in detail-rich areas [3, 16, 24, 34].

Extensions to texture. To address the limited expressivity of standard Gaussians, *texture-based extensions* have also emerged. GSTex [23] and HDGS [27] augment 2D Gaussian splatting [11] by attaching learnable texture maps to each primitive. Texture-GS [29] and Textured Gaussians [4] extend this paradigm to 3DGS, enabling better disentanglement of geometry and appearance. Textured-GS [12] further enhances this with spherical harmonics for spatially-varying color and opacity. Billboard Splatting [28] proposes a new representation using textured planar primitives, offering improved quality at the cost of increased training time.

STRinGS focuses on text. While these works enhance overall visual fidelity, they do not explicitly target semantic regions such as text, which are vital for downstream applications. In contrast, our method introduces selective refinement for text regions in 3DGS. By decoupling the optimization of text and non-text regions, STRinGS achieves sharper and more readable textual content with fewer training iterations and without degrading overall scene quality.

3. STRinGS-360 Dataset

Existing 3D scene datasets often lack semantically meaningful text, *i.e.*, text that provides information relevant to the scene, on foreground objects. When present, text is typically sparse and relegated to the background, making these datasets unsuitable for evaluating methods that target text-specific refinement. Moreover, datasets such as DL3DV-10K Benchmark [18] offer only flat or panned views rather than full 360° coverage, restricting the ability to assess text reconstruction across diverse viewpoints.

To address these limitations, we introduce STRinGS-

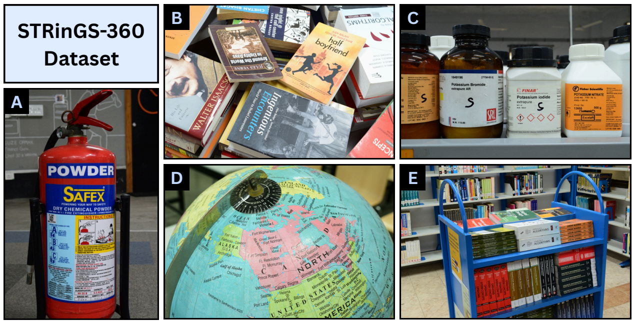


Figure 2. Overview of the scenes in our STRinGS-360 dataset. Each scene contains semantically meaningful text elements: (A) Extinguisher, (B) Books, (C) Chemicals, (D) Globe, and (E) Shelf. The dataset is designed to evaluate text reconstruction performance under diverse layouts and text orientations.

360, a curated dataset of *five* indoor scenes designed to benchmark text readability in 3D Gaussian Splatting (Fig. 2). Each scene centers on a single or a set of object(s) containing dense, semantically meaningful text exhibiting several challenges. A. *Extinguisher* features instructional text on a curved cylindrical surface; B. *Books* contains flat, densely packed book titles with author names; C. *Chemicals* presents chemical compositions on labeled bottles in a laboratory shelf; D. *Globe* includes geographical names on a spherical surface; and E. *Shelf* shows stacks of academic books in a structured and sometimes occluded setting, with repeated titles commonly found in libraries. These scenes span flat, cylindrical, and spherical configurations and offer a diverse and realistic benchmark for evaluating fine-grained textual fidelity in 3D reconstructions.

4. STRinGS Methodology

We present an overview of STRinGS in Fig. 3. We begin with preprocessing: SfM and text segmentation (Sec. 4.1) followed by segmenting text regions in 3D (Sec. 4.2). Next, we propose our two-phase optimization that selectively refines text regions (Sec. 4.3) followed by integration with non-text regions and full scene optimization (Sec. 4.4).

4.1. Preprocessing

COLMAP SfM. Given n input images $\mathcal{I} = \{I_1, \dots, I_n\}$ of a static scene captured from different viewpoints, 3DGS begins by extracting geometric information required for initialization. Specifically, we obtain a sparse 3D point cloud of m points $\mathcal{P} = \{\mathbf{P}_1, \dots, \mathbf{P}_m\}$, camera poses associated with the images $\mathcal{C} = \{C_1, \dots, C_n\}$, and the camera intrinsics K using the COLMAP pipeline [25, 26]. Additionally, for each point $\mathbf{P}_i$, COLMAP provides a visibility set $V_i \subseteq \{1, \dots, n\}$ indexing the subset of images in which the point is observed. We denote the collection of these visibility sets as $\mathcal{V} = \{V_1, \dots, V_m\}$.

Text segmentation. To identify and isolate textual regions in the undistorted images output by COLMAP, we employ Hi-SAM [30], a model capable of segmenting text at multiple scales and orientations. We refer to the binary mask for image I_j as M_j, and the set of all masks as $\mathcal{M} = \{M_1, \dots, M_n\}$.

Algorithm 1: Text Segmentation in 3D

```
Input: Point cloud $\mathcal{P}$; camera intrinsics $K$; camera
       poses $\mathcal{C}$; text masks $\mathcal{M}$; visibility sets $\mathcal{V}$;
       visibility threshold $\tau$
Output: $\mathcal{P}_{\text{text}}, \mathcal{P}_{\text{non-text}}$
$\mathcal{P}_{\text{text}} \leftarrow \emptyset$
for each point $\mathbf{P}_i \in \mathcal{P}$, where $i = 1$ to $m$ do
    count $\leftarrow 0$
    for each image index $j \in V_i$ do
        // Perspective Projection
        $\mathbf{u}_{ij} \leftarrow \pi(K, C_j, \mathbf{P}_i)$
        if $M_j(\mathbf{u}_{ij}) = 1$ then
            count $\leftarrow$ count $+ 1$
    if count $\geq \tau$ then
        $\mathcal{P}_{\text{text}} \leftarrow \mathcal{P}_{\text{text}} \cup \{\mathbf{P}_i\}$
$\mathcal{P}_{\text{non-text}} \leftarrow \mathcal{P} \setminus \mathcal{P}_{\text{text}}$
Return $\mathcal{P}_{\text{text}}, \mathcal{P}_{\text{non-text}}$
```

4.2. Text Segmentation in 3D

To enable text-aware reconstruction in our pipeline, we first identify the subset of 3D points that correspond to text regions in the scene. This is done by projecting each 3D point (from COLMAP) into all images where it is visible, and checking whether its 2D projection falls inside the corresponding Hi-SAM text mask. A point is classified as a text point if it lies within the text region in at least τ images. In our method, we set the visibility threshold $\tau = 1$. The set of *text points* is denoted as $\mathcal{P}_{\text{text}} \subseteq \mathcal{P}$, and its complement as $\mathcal{P}_{\text{non-text}} = \mathcal{P} \setminus \mathcal{P}_{\text{text}}$. The pseudo-code for this process is provided in Algorithm 1.

The Gaussians used in 3DGS are initialized directly from the sparse point cloud $\mathcal{P}$, with each point providing the 3D location (x, y, z) of a Gaussian. Leveraging the text/non-text partitioning from above, we define $\mathcal{G}_{\text{text}}$ and $\mathcal{G}_{\text{non-text}}$ as the initial sets of Gaussians corresponding to $\mathcal{P}_{\text{text}}$ and $\mathcal{P}_{\text{non-text}}$ respectively. These subsets serve as the basis of our two-phase training strategy described next.

4.3. Phase 1: Selective Text Reconstruction

We start GS training using the text Gaussians $\mathcal{G}_{\text{text}}$, obtained through the 3D text segmentation process above. This phase runs for T_1 iterations (3K), and optimization is performed on the subset of images with non-empty text masks.

Densification of text Gaussians. Since the initialization is based on a sparse point cloud, high-frequency structures

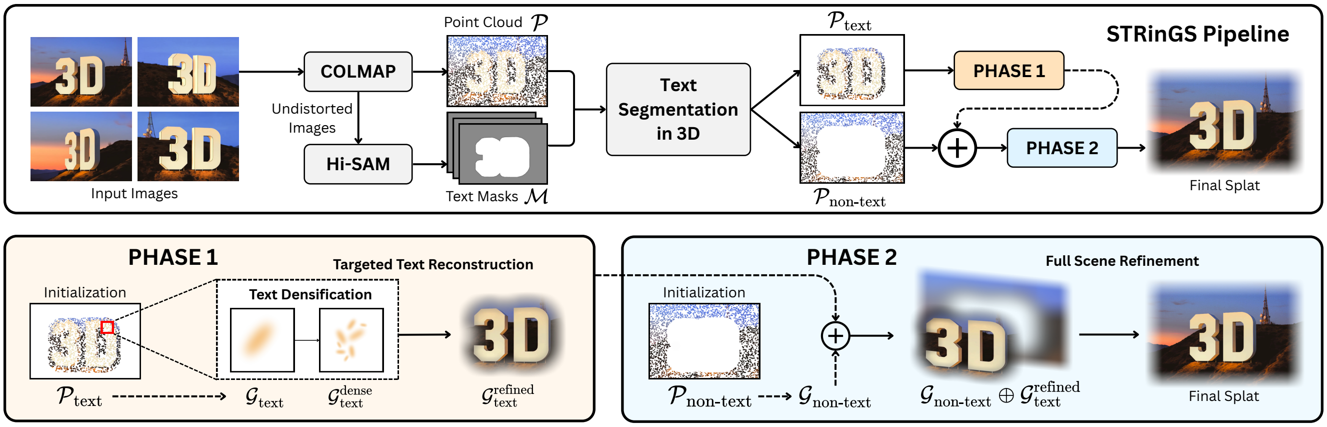

Input Images
COLMAP
Undistorted Images
Hi-SAM
Point Cloud $\mathcal{P}$
Text Masks $\mathcal{M}$
Text Segmentation in 3D
$\mathcal{P}_{\text{text}}$
$\mathcal{P}_{\text{non-text}}$
PHASE 1
PHASE 2
STRinGS Pipeline
Final Splat
PHASE 1
Targeted Text Reconstruction
Initialization
Text Densification
$\mathcal{P}_{\text{text}}$
$\mathcal{G}_{\text{text}}$
$\mathcal{G}_{\text{text}}^{\text{dense}}$
$\mathcal{G}_{\text{text}}^{\text{refined}}$
PHASE 2
Full Scene Refinement
Initialization
$\mathcal{P}_{\text{non-text}}$
$\mathcal{G}_{\text{non-text}}$
$\mathcal{G}_{\text{non-text}} \oplus \mathcal{G}_{\text{text}}^{\text{refined}}$
Final Splat

Figure 3. **STRinGS overview.** Given n input images, we use COLMAP to obtain a point cloud $\mathcal{P}$ and undistorted images, which are passed to Hi-SAM [30] to obtain text masks $\mathcal{M}$. $\mathcal{P}$ and $\mathcal{M}$ are passed to the Text Segmentation in 3D module (Sec. 4.2, Algorithm 1) to obtain partitioned text and non-text point clouds. These are processed through a two-phase pipeline. In phase 1 (Sec. 4.3), we perform targeted densification and reconstruction of text Gaussians. In phase 2 (Sec. 4.4), we perform full scene refinement, where text and non-text Gaussians are optimized with distinct learning strategies, enabling targeted enhancement of text without compromising scene quality. The final output is a text-refined Gaussian Splat representation with enhanced text readability while preserving overall scene fidelity.

(text) may be underrepresented, especially in cases where the number of viewpoints observing the text is small. To address this, we adopt a visibility-based densification strategy at the start of phase 1. Note, this is a one-time densification in addition to the standard densification process used in 3DGS. Specifically, the number of duplicates N_i for each Gaussian $\mathbf{g}_i \in \mathcal{G}_{\text{text}}$ is inversely proportional to its visibility:

$$N_i = \left\lfloor \frac{1/c_i - \min_k(1/c_k)}{\max_k(1/c_k) - \min_k(1/c_k)} \cdot (N_{\max} - 1) + 1 \right\rfloor . \quad (1)$$

$c_i = |V_i|$ is the visibility count of point $\mathbf{P}_i$ and corresponding Gaussian $\mathbf{g}_i$. The parameter $N_{\max}$ defines the maximum densification factor, chosen to be between 15-25 based on the density of text in the scene.

We apply the densify-and-split strategy to each Gaussian, guided by its densification factor. This results in multiple smaller Gaussians at slightly perturbed positions that cover the same volume, thereby enabling an efficient representation of text. The result of this process is an augmented set of text Gaussians, denoted as $\mathcal{G}_{\text{text}}^{\text{dense}}$. The necessity and effectiveness of this densification are discussed in Sec. 5.3.

Text region loss. To ensure that the optimization focuses on text regions, we use the segmented text masks to modify the loss function. Specifically, for an image I_j and its rendered counterpart R_j, the reconstruction loss is:

$$\mathcal{L}_1^{\text{text}} = \|I_j \odot M_j - R_j \odot M_j\|_1 . \quad (2)$$

where $\odot$ denotes element-wise multiplication and M_j is the binary text mask. This replaces the standard photometric loss formulation in 3DGS that combines $\mathcal{L}_1$ and D-SSIM terms over the entire image [14].

Locking position parameters. 3DGS typically employs a coarse-to-fine optimization schedule where the position parameters of Gaussians are updated with relatively high learning rates (LRs) at the start. This often causes them to drift away from high-frequency regions such as text. As our text Gaussians are initialized at text regions, we lock them in position by setting their position LR to zero, while allowing other parameters to be updated.

The output of phase 1 is a refined set of text Gaussians, denoted $\mathcal{G}_{\text{text}}^{\text{refined}}$, used in phase 2 for full scene optimization.

4.4. Phase 2: Full Scene Refinement

We now focus on jointly optimizing both text and non-text regions of the scene. The refined text Gaussians $\mathcal{G}_{\text{text}}^{\text{refined}}$ are combined with initial non-text Gaussians $\mathcal{G}_{\text{non-text}}$ obtained from 3D text segmentation process. After T_1 (3K) iterations of phase 1, phase 2 runs up to T_2 (30K) iterations.

In this phase of training, we maintain the same loss function as in 3DGS [14], including the D-SSIM component, to

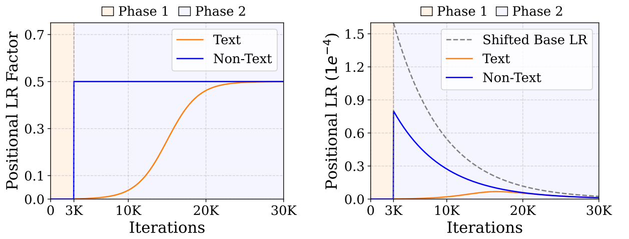


Figure 4. Learning rate (LR) of the position parameter for Gaussians in STRinGS (see Eq. (3)). **Left:** Learning rate scaling factor $\eta_r(t)$ for text and non-text Gaussians. **Right:** Effective LR obtained by modulating a shifted base exponential decay schedule $\eta_{\text{opt}}(t)$ from 3DGS with these factors. α=0.5, β=0.0005, γ=15000. Note, phase 1 sets the position learning rate of $\mathcal{G}_{\text{text}}$ to 0 while $\mathcal{G}_{\text{non-text}}$ is not optimized. In phase 2, we introduce differentiated learning for text and non-text content.

OCR-CER ↓	TandT		DL3DV-10K		STRinGS-360	
	7K	30K	7K	30K	7K	30K
3DGS SIGGRAPH'23	0.209	0.121	0.392	0.157	0.736	0.148
Mip-Splatting CVPR'24	0.222	0.125	0.392	0.149	0.748	0.129
3DGS-MCMC NeurIPS'24	0.272	0.120	0.511	0.142	0.927	0.110
AbsGS ACMMM'24	0.249	0.137	0.411	0.160	0.768	0.143
EDC-AbsGS arXiv'25	0.142	0.118	0.239	0.162	0.328	0.116
STRinGS (Ours)	0.122	0.099	0.187	0.123	0.177	0.106

Table 1. OCR-based Character Error Rate (CER ↓) on rendered images at 7K and 30K training iterations averaged over all scenes in the dataset. Lower CER indicates better text readability. Red, orange, and yellow highlights indicate the first, second, and third best performing technique.

ensure full scene refinement. We also follow the standard procedures for densification, splitting, and cloning Gaussians as 3DGS.

Modulating position learning rates. A key concern is preserving the quality of $\mathcal{G}_{\text{text}}^{\text{refined}}$ that may drift from their position if updated indiscriminately to minimize global photometric loss. To address this, we apply a text region dependent LR for the positions of text and non-text Gaussians separately.

For $\mathcal{G}_{\text{text}}$, we propose an *increasing LR factor* as a sigmoid function. This results in conservative early updates that preserve existing structure while providing flexibility later. Conversely, for $\mathcal{G}_{\text{non-text}}$, we apply a constant multiplier α to ensure compatibility with the lowered LR for $\mathcal{G}_{\text{text}}$ and avoid destabilizing updates.

The region-specific LR factor $\eta_r(t)$ for the position of a Gaussian $\mathbf{g}$ at iteration $t \in [T_1, T_2]$ is:

$$\eta_r(t) = \begin{cases} \dfrac{\alpha}{1 + e^{-\beta \cdot (t - \gamma)}} & \text{if } \mathbf{g} \in \mathcal{G}_{\text{text}}^{\text{refined}}\,, \\ \alpha & \text{if } \mathbf{g} \in \mathcal{G}_{\text{non-text}}\,. \end{cases} \tag{3}$$

Next, let $\eta_{\text{base}}(t)$ be the LR schedule adopted by vanilla 3DGS. We shift this by T_1 iterations to obtain $\eta_{\text{opt}}(t)$. Then, the effective LR used to update the position of each Gaussian is $\eta_{\text{effective}}(\mathbf{g}, t) = \eta_r(t) \cdot \eta_{\text{opt}}(t)$, and is illustrated in Fig. 4. We explain hyperparameter choices in Appendix A.

Overall, STRinGS's hybrid strategy enables targeted and region-aware optimization, ensuring sharp and readable text while preserving overall scene quality.

5. Experiments and Results

Following standard protocol in the 3DGS literature [14], every 8th image is held out as an evaluation view to assess novel view synthesis performance. Each scene is trained for T_2 (30K) iterations. To evaluate results on early text reconstruction, we also report results at 7K iterations. All experiments are conducted on an Nvidia RTX 3090 Ti GPU with 24GB VRAM.

The pipeline involves running COLMAP to obtain the sparse point cloud, camera poses, and undistorted images. The undistorted images are passed to the Hi-SAM-L [30] model which outputs tight polygonal text masks. These are dilated using a circular kernel with a diameter equal to 5% of the image width, thereby spanning the visual footprint of a text region, which includes the text strokes and immediate background context. This is followed by the two-stage training procedure outlined in Sec. 4.

Training Time	TandT		DL3DV-10K		STRinGS-360	
(in minutes) ↓	7K	30K	7K	30K	7K	30K
3DGS SIGGRAPH'23	2.0	13.8	2.8	15.1	2.5	17.2
Mip-Splatting CVPR'24	3.4	20.8	6.3	30.4	5.7	31.7
3DGS-MCMC NeurIPS'24	3.1	19.6	5.2	28.3	5.6	34.0
AbsGS ACMMM'24	2.6	12.7	5.3	20.7	5.4	22.5
EDC-AbsGS arXiv'25	2.8	12.7	6.0	22.2	5.5	23.2
STRinGS (Ours)	1.1	9.6	2.1	11.4	1.9	12.6

Table 2. Training time in minutes at 7K and 30K training iterations, averaged over all scenes in the dataset.

5.1. OCR-based Evaluation

3D reconstruction quality is typically measured using image-based metrics such as PSNR, SSIM, and LPIPS [33], which quantify similarity between rendered and ground-truth images. They are computed by averaging pixel-level or perceptual differences over entire images, often dominated by background non-textual regions. While effective at assessing global appearance, these metrics fall short in evaluating the semantic fidelity of reconstructed text.

In our scenes, even if text occupies a small fraction of the images, it has high semantic importance. Character-level distortions, misalignments, or partial blurring may severely impair text legibility, however, barely affects PSNR or SSIM scores. To address this limitation, we introduce an OCR-based evaluation score that measures the quality of text reconstruction. Specifically, we run Google OCR API [5] on the rendered views and the corresponding ground-truth images. For each evaluation image, we compute the Character Error Rate (CER): the normalized Levenshtein distance between recognized and ground-truth text, using a recall-based approach that penalizes missing and mismatched ground-truth characters. OCR-CER reflects how well the reconstructed image retains readable and accurate textual information. The CER scores are aggregated across all evaluation views within each scene. Additional details are provided in Appendix C.

5.2. Comparison with Existing Works

Baselines. We compare against vanilla 3DGS [14] and other recent methods. While there are no existing methods targeting text reconstruction, Mip-Splatting [32], 3DGS-MCMC [15], AbsGS[31], and EDC-AbsGS [6] [1] serve as strong baselines as they refine the overall scene.

[1] By EDC-AbsGS, we refer to this implementation https://github.com/XiaoBin2001/EDC linked in their arXiv preprint.

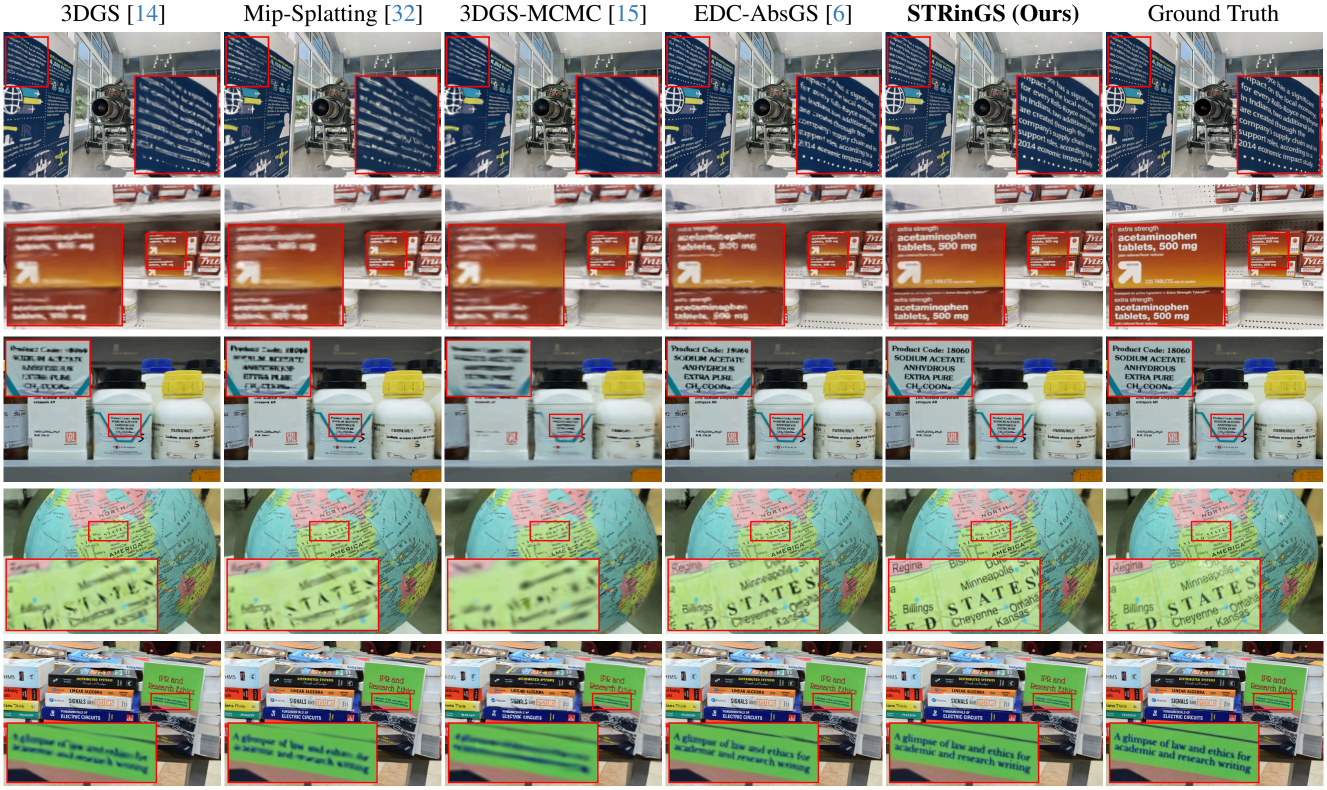


Figure 5. Qualitative comparison of different methods at 7K training iterations on scenes from the DL3DV-10K Benchmark [18] (rows 1, 2) and our STRinGS-360 (rows 3-5) datasets. While existing methods struggle to reconstruct text accurately at this early stage, our STRinGS framework produces significantly sharper and more legible text regions. (Best seen on screen)

Datasets. We evaluate all methods on a diverse set of 14 scenes drawn from existing benchmarks and STRinGS-360. This includes 2 scenes from the Tanks and Temples dataset [17], 7 selected scenes from the DL3DV-10k Benchmark [18] that feature varying amounts of textual content, and 5 scenes from our STRinGS-360 dataset, consisting of sharp, dense, and semantically meaningful text.

Text reconstruction results. We compare model performance at two stages: 7K and 30K iterations. Tab. 1 shows that STRinGS achieves the lowest OCR-CER, with a big gap at 7K iterations. The relative improvements, averaged over all datasets are: 63.6% 3DGS, 64.3% Mip-Splatting, 71.6% 3DGS-MCMC, 66.0% AbsGS, and 31.4% EDC-AbsGS. Fig. 5 visualizes the noticeably sharper and readable text at 7K iterations for various scenes. STRinGS does especially well on reconstructing small text such as “acetaminophen” (row 2), “product code 18060” (row 3), or names on the globe such as “Minneapolis” (row 4).

While other methods bridge the gap at 30K iterations, STRinGS still outperforms them with a relative improvement in OCR-CER scores: 23.0% 3DGS, 18.6% Mip-Splatting, 11.8% 3DGS-MCMC, 25.4% AbsGS, and 17.2% EDC-AbsGS. A few examples are visualized in Fig. 6.

STRinGS is most effective when text regions contain few points at initialization or when the text is visible in a small subset ($< 5\%$) of images, where other methods tend to fail. Importantly, this targeted text refinement results in comparable overall scene quality and fewer Gaussians (Tab. 4). What distinguishes our method from others is its ability to accurately reconstruct small text, whereas other methods can already handle large text reasonably well, as detailed in Appendix D.2. We also demonstrate the effectiveness of our method on multilingual text refinement in Appendix D.1.

5.3. Ablations and Key Highlights

Effect of text densification. To address sparse points at initialization in text regions leading to under reconstruction, we introduce a targeted text densification step in phase 1 (Sec. 4.3). As illustrated in Fig. 7, the benefits of text-aware densification are evident. Vanilla 3DGS fails to reconstruct the text even after 30K iterations, while STRinGS without text densification also fails to reconstruct the text. Our approach with densification successfully reconstructs sharp and accurate text at 30K iterations while clearly showing a few letters even at earlier stages of training. Results showing the effect of text densification are presented in Tab. 3. We see consistent improvements in OCR-CER indicating better text reconstruction across all datasets.

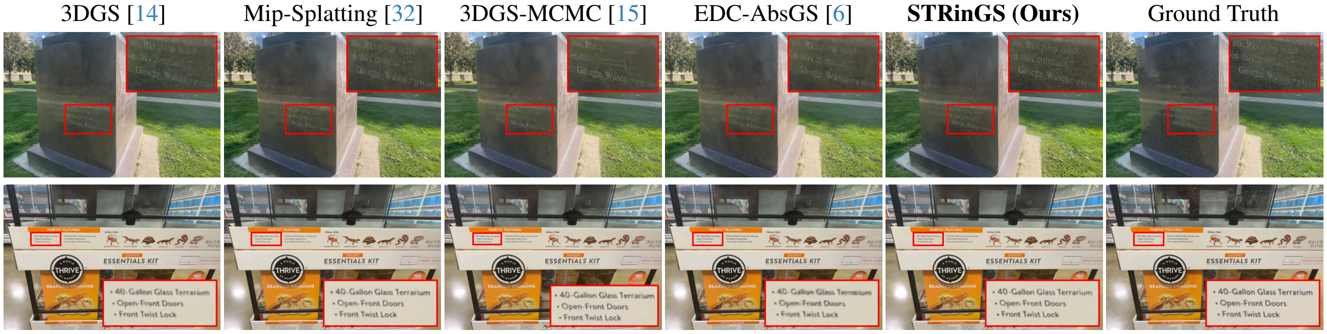


Figure 6. Qualitative comparison of different methods at 30K training iterations on scenes from DL3DV-10K Benchmark [18]. STRinGS consistently preserves text clarity, even in visually challenging regions where other methods miss fine textual details. (Best seen on screen)

Effect of position LR of Gaussians. To assess the impact of the position LR during phase 1 (Sec. 4.3), we evaluate the outputs at the end of this phase using OCR-CER. As shown in Fig. 9, using a non-zero LR for the positions of Gaussians leads to significant degradation in text reconstruction. This is especially important in our setting, where Gaussians are already densely placed over text regions through explicit densification. By setting the position LR to zero, we freeze their locations, allowing the optimization of other parameters such as scale, opacity, and spherical harmonic coefficients leading to sharper text reconstruction. Results in Tab. 3 show that zero position LR is crucial for improving text quality from the early stages (3K iterations of phase 1) indicated by the significantly improved OCR-CER.

Training speed. Our method achieves better text reconstruction quality with significantly lower training time, compared to existing densification-based approaches (Tab. 2). Densification in standard 3DGS relies on large positional gradients to dynamically add Gaussians during training, which introduces significant computational overhead. In contrast, STRinGS sets the position LR to zero in the first phase and keeps it lower than 3DGS in the second, effectively limiting unnecessary densification. Since we explicitly add Gaussians in text regions at the start of phase 1, we avoid the need for extensive gradient-driven densification, leading to faster and more efficient training.

While EDC-AbsGS is the strongest baseline in terms of CER, compared to STRinGS, it requires 2.8× training time for 7K iterations and 1.7× for 30K iterations. On the other hand, 3DGS is closest to STRinGS in training time (only 1.4× at both 7K and 30K), but performs significantly worse in text reconstruction quality (Tab. 1). These results highlight that STRinGS performs the best in terms of both efficiency and accuracy. The trade-off between OCR-CER and training time across methods is visualized in Fig. 1. A detailed breakdown of the time required for preprocessing (COLMAP and text segmentation) and training (phases 1 and 2) is provided in Appendix D.3.

Dataset	TandT		DL3DV-10K		STRinGS-360	
	Effect of text densification					
OCR-CER ↓	w/o	Ours	w/o	Ours	w/o	Ours
(7K iterations)	0.196	**0.122**	0.316	**0.187**	0.437	**0.177**
	Effect of zero position LR of Gaussians					
OCR-CER ↓	w/o	Ours	w/o	Ours	w/o	Ours
(3K iterations)	0.342	**0.289**	0.618	**0.278**	0.948	**0.347**

Table 3. Ablations. The effect of text densification and the effect of zero position LR of Gaussians in phase 1. The OCR-CER values, averaged over all scenes in the datasets demonstrate the necessity of both components for accurate text reconstruction.

Early text reconstruction. We demonstrate the evolution of text reconstruction quality over training iterations on the *Extinguisher* scene from our dataset. Our method achieves noticeably better text reconstruction at early stages (3K and 7K iterations) compared to vanilla 3DGS (Fig. 8). The accompanying plot illustrates the evolution of OCR-CER across iterations, showing that our method reconstructs text accurately much earlier.

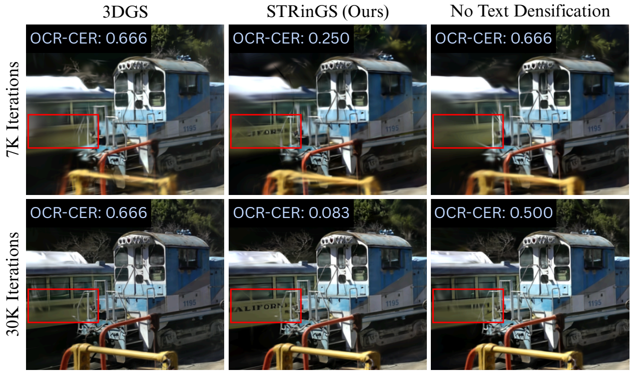


Figure 7. Effect of text densification on a scene from the Tanks&Temples [17] dataset. **Left**: Vanilla 3DGS fails to reconstruct readable text even after 30k iterations, resulting in high OCR-CER of 0.666. **Middle**: STRinGS (Ours) with text densification achieves sharp and semantically meaningful text as early as 7K iterations which improves further at 30K iterations (0.083 CER). **Right**: Without text densification, our method struggles to produce accurate and legible text, demonstrating the importance of targeted densification of text regions.

Method	Tanks&Temples				DL3DV-10K Benchmark				STRinGS-360 (Ours)			
	PSNR↑	SSIM↑	LPIPS↓	Points↓	PSNR↑	SSIM↑	LPIPS↓	Points↓	PSNR↑	SSIM↑	LPIPS↓	Points↓
3DGS SIGGRAPH'23	23.73	0.8524	0.1692	1576K	30.20	0.9348	0.1456	1175K	28.85	0.9126	0.2107	1391K
Mip-Splatting CVPR'24	23.81	0.8596	0.1563	2366K	30.47	0.9390	0.1329	1610K	28.80	0.9142	0.2012	1875K
3DGS-MCMC NeurIPS'24	24.43	0.7688	0.1508	1550K	30.46	0.9390	0.1394	1182K	29.85	0.9234	0.1971	1388K
AbsGS ACMMM'24	23.64	0.8526	0.1616	1297K	30.18	0.9360	0.1368	874K	28.77	0.9111	0.2044	1240K
EDC-AbsGS arXiv'25	23.73	0.8595	0.1557	1382K	30.45	0.9400	0.1321	857K	29.30	0.9183	0.1992	1041K
STRinGS (Ours)	23.88	0.8513	0.1767	1354K	30.14	0.9338	0.1477	918K	29.00	0.9138	0.2166	965K

Table 4. Comparison of reconstruction quality and number of Gaussians (Points) at 30K iterations across three datasets: Tanks&Temples [17], DL3DV-10K [18], and STRinGS-360. Our method achieves comparable PSNR, SSIM, and LPIPS scores, indicating no degradation in overall scene quality, while requiring slightly lesser Points especially in text-rich scenes (STRinGS-360 dataset).

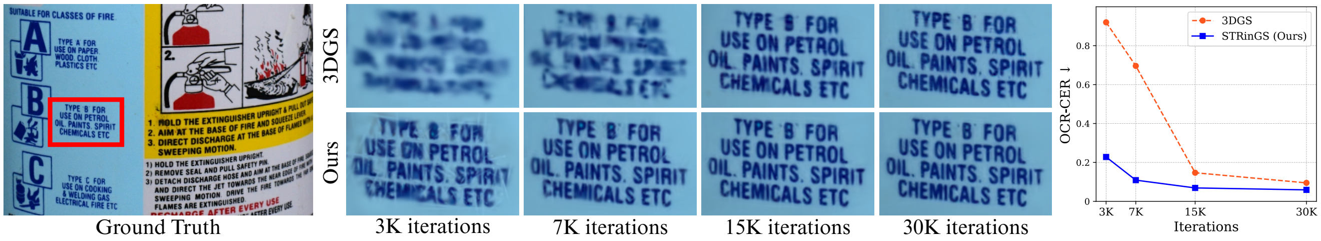


Figure 8. Text reconstruction across training iterations on the *Extinguisher* scene from our STRinGS-360 dataset. STRinGS achieves clearer and more accurate text reconstruction earlier than 3DGS, as reflected in the plot for OCR-CER of the scene over iterations.

5.4. Discussion

Applications. STRinGS is well-suited for use cases where both quality and efficiency are critical. For example, autonomous navigation requires early recovery of readable text for tasks like interpreting signs/directions and waypoint recognition. In robotics, clear reconstruction of text assists in scene understanding and labeled object identification. In AR/VR environments, user experience is enhanced by good quality of reconstructed text. Further, STRinGS may prove valuable in cultural heritage applications, where reconstructing inscriptions such as ancient stone carvings, temple wall engravings, or historical monument plaques as 3D models can aid archival and restoration efforts.

Limitations. STRinGS uses Hi-SAM for 2D text segmentation that introduces computational overhead during preprocessing and may miss text in cluttered scenes. However, this can be swapped out for future models that improve text segmentation. Future work could focus on reducing Hi-SAM's computational overhead, for instance by performing 3D text segmentation on only a strategically chosen subset of images rather than the full set. Additionally, STRinGS fails when text in input images is unreadable due to low resolution, making reconstruction inherently limited.

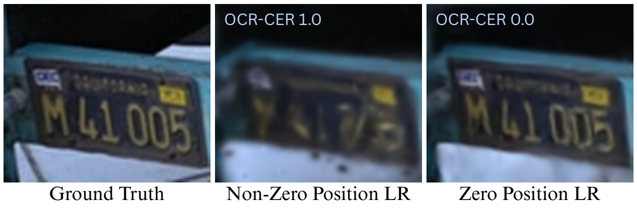


Figure 9. Effect of position learning rate at the end of phase 1 (3K iterations) on a scene from the Tanks&Temples [17] dataset. A non-zero LR causes Gaussians to drift, leading to poor text reconstruction (CER = 1.0). Instead, freezing their positions (zero LR) preserves spatial alignment, enabling text readability (CER = 0.0).

6. Conclusion

We introduced STRinGS, a novel text-aware refinement framework that explicitly focuses on reconstructing sharp, clear and readable text. By treating text and non-text regions separately, our two-phase optimization enables early recovery of textual content. Extensive evaluations across diverse text-rich scenes demonstrated that STRinGS consistently outperforms baselines, achieving significantly lower OCR-based Character Error Rates, particularly at early iterations, highlighting its potential for time-sensitive applications. We also proposed STRinGS-360, a curated dataset specifically designed for evaluating text readability in 3D reconstructions. By using OCR-CER as a measure for text readability, we quantitatively validated the improvements offered by our method over vanilla 3DGS and its variants. In summary, STRinGS establishes a new direction for text-aware 3D scene understanding, highlighting the importance of semantic detail preservation in 3D scene reconstruction.

Acknowledgements. We thank Harshavardhan P. for feedback and guidance throughout this project. RKS thanks Digital India Bhashini Division, Ministry of Electronics and Information Technology (MeiTY), Government of India for supporting the project, and MT thanks Adobe Research for travel support.

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 *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-NeRF: Anti-Aliased Grid-Based Neural Radiance Fields. In *International Conference on Computer Vision (ICCV)*, 2023. 1

[3] Kin-Chung Chan, Jun Xiao, Hana Lebeta Goshu, and Kin-Man Lam. Point Cloud Densification for 3D Gaussian Splatting from Sparse Input Views. In *ACM Multimedia (MM)*, 2024. 2

[4] Brian Chao, Hung-Yu Tseng, Lorenzo Porzi, Chen Gao, Tuotuo Li, Qinbo Li, Ayush Saraf, Jia-Bin Huang, Johannes Kopf, Gordon Wetzstein, et al. Textured Gaussians for Enhanced 3D Scene Appearance Modeling. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 2

[5] Google Cloud. Cloud Vision API Documentation. `https://cloud.google.com/vision/`. 2, 5

[6] Xiaobin Deng, Changyu Diao, Min Li, Ruohan Yu, and Duanqing Xu. Efficient Density Control for 3D Gaussian Splatting. *arXiv preprint arXiv:2411.10133*, 2024. 2, 5, 6, 7, 4, 8

[7] Guangchi Fang and Bing Wang. Mini-Splatting: Representing Scenes with a Constrained Number of Gaussians. In *European Conference on Computer Vision (ECCV)*, 2024. 2

[8] Guangchi Fang and Bing Wang. Mini-Splatting2: Building 360 Scenes within Minutes via Aggressive Gaussian Densification. *arXiv preprint arXiv:2411.12788*, 2024. 2

[9] Kyle Gao, Yina Gao, Hongjie He, Dening Lu, Linlin Xu, and Jonathan Li. NeRF: Neural Radiance Field in 3D Vision, a Comprehensive Review. *arXiv preprint arXiv:2210.00379*, 2022. 2

[10] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep Blending for Free-Viewpoint Image-Based Rendering. *ACM Transactions on Graphics (TOG)*, 37(6):1–15, 2018. 2

[11] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2D Gaussian Splatting for Geometrically Accurate Radiance Fields. In *SIGGRAPH Conference Papers*, 2024. 2

[12] Zhentao Huang and Minglun Gong. Textured-GS: Gaussian Splatting with Spatially Defined Color and Opacity. *arXiv preprint arXiv:2407.09733*, 2024. 2

[13] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large Scale Multi-View Stereopsis Evaluation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 2

[14] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics (TOG)*, 42(4):139–1, 2023. 1, 2, 4, 5, 6, 7, 8

[15] Shakiba Kheradmand, Daniel Rebain, Gopal Sharma, Weiwei Sun, Yang-Che Tseng, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. 3D Gaussian Splatting as Markov Chain Monte Carlo. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 2, 5, 6, 7, 4, 8

[16] Sieun Kim, Kyungjin Lee, and Youngki Lee. Color-Cued Efficient Densification Method for 3D Gaussian Splatting. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[17] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and Temples: Benchmarking Large-Scale Scene Reconstruction. *ACM Transactions on Graphics (TOG)*, 36(4):1–13, 2017. 2, 6, 7, 8, 5

[18] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. DL3DV-10K: A Large-Scale Scene Dataset for Deep Learning-Based 3D Vision. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 6, 7, 8, 4, 5

[19] Saswat Subhajyoti Mallick, Rahul Goel, Bernhard Kerbl, Markus Steinberger, Francisco Vicente Carrasco, and Fernando De La Torre. Taming 3DGS: High-Quality Radiance Fields with Limited Resources. In *SIGGRAPH Asia Conference Papers*, 2024. 2

[20] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2

[21] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Transactions on Graphics (TOG)*, 41(4):1–15, 2022. 1, 2

[22] PJ Narayanan, Peter W Rander, and Takeo Kanade. Constructing Virtual Worlds Using Dense Stereo. In *International Conference on Computer Vision (ICCV)*, 1998. 1

[23] Victor Rong, Jingxiang Chen, Sherwin Bahmani, Kiriakos N Kutulakos, and David B Lindell. GSTeX: Per-Primitive Texturing of 2D Gaussian Splatting for Decoupled Appearance and Geometry Modeling. In *Winter Conference on Applications of Computer Vision (WACV)*, 2025. 2

[24] Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. Revising Densification in Gaussian Splatting. In *European Conference on Computer Vision (ECCV)*, 2024. 2

[25] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 3

[26] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise View Selection for Unstructured Multi-View Stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 2, 3

[27] Yunzhou Song, Heguang Lin, Jiahui Lei, Lingjie Liu, and Kostas Daniilidis. HDGS: Textured 2D Gaussian Splatting for Enhanced Scene Rendering. *arXiv preprint arXiv:2412.01823*, 2024. 2

[28] David Svitov, Pietro Morerio, Lourdes Agapito, and Alessio Del Bue. BillBoard Splatting (BBSplat): Learnable Textured Primitives for Novel View Synthesis. In *International Conference on Computer Vision (ICCV)*, 2025. 2

[29] Tian-Xing Xu, Wenbo Hu, Yu-Kun Lai, Ying Shan, and Song-Hai Zhang. Texture-GS: Disentangling the Geometry

and Texture for 3D Gaussian Splatting Editing. In *European Conference on Computer Vision (ECCV)*, 2024. 2

[30] Maoyuan Ye, Jing Zhang, Juhua Liu, Chenyu Liu, Baocai Yin, Cong Liu, Bo Du, and Dacheng Tao. Hi-SAM: Marrying Segment Anything Model for Hierarchical Text Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 47(3):1431–1447, 2024. 3, 4, 5

[31] Zongxin Ye, Wenyu Li, Sidun Liu, Peng Qiao, and Yong Dou. ABSGS: Recovering Fine Details in 3D Gaussian Splatting. In *ACM Multimedia (MM)*, 2024. 2, 5

[32] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 5, 6, 7, 4, 8

[33] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 5

[34] Zheng Zhang, Wenbo Hu, Yixing Lao, Tong He, and Hengshuang Zhao. Pixel-GS: Density Control with Pixel-Aware Gradient for 3D Gaussian Splatting. In *European Conference on Computer Vision (ECCV)*, 2024. 2

STRinGS: Selective Text Refinement in Gaussian Splatting
Supplementary Material

A. Experimental Design Choices

Segmentation model and mask dilation. We employ Hi-SAM for image-level text segmentation due to its strong performance in detecting multi-scale, arbitrarily oriented scene text. Hi-SAM produces tight polygonal masks, which helps in precisely isolating text-bearing regions from the background.

To compensate for over-constraining effects of tight Hi-SAM masks, we apply morphological dilation. This adjustment is motivated by the observation that a text region's visual footprint includes not only the strokes themselves but also immediate background context, which is often captured by nearby Gaussians. Dilation improves performance in both 3D text segmentation and mask-based supervision in phase 1.

Our framework remains compatible with any text segmentation model capable of generating binary text masks. Since, Hi-SAM is computationally expensive, it can be replaced by faster models, at the cost of text segmentation quality. Additionally, since consecutive frames contain sufficient overlap between text regions and we set the visibility threshold to 1, text masks can be obtained only for a subset of images instead of the full set. Based on these, the points in 3D can be segmented into text and non-text regions, reducing the text segmentation overload.

Visibility threshold in 3D text segmentation. Each 3D point reconstructed by COLMAP includes a visibility set indicating the subset of images in which it appears. For 3D text segmentation, we project points onto these visible images and check for overlap with the corresponding text masks. A point is marked as text if it lies within a mask in at least τ views, with τ set to 1 in all our experiments.

This low threshold is essential as certain textual elements may be visible only from one or two viewpoints due to occlusions, viewing angle, or lighting. A higher value of τ could result in such points being misclassified as non-text, especially in sparsely observed or cluttered scenes. Setting $\tau = 1$ ensures that points corresponding to text regions are not misclassified.

Densification parameters. Since COLMAP-based initialization is sparse and often under-samples fine structures like text, we adopt a inverse visibility-based densification strategy during phase 1. Each text-associated point is duplicated N_i times, with the duplication factor constrained between 1 and a scene-specific N_{max}.

The maximum densification factor N_{max} is selected adaptively based on the overall textual content in the scene. For text-sparse scenes, we set N_{max} to 15 or 20 to ensure adequate coverage despite fewer text points relative to non-text points. For text-rich environments such as those in our STRinGS-360 dataset, we set $N_{\text{max}} = 25$ to capture the dense and overlapping text structures more accurately. This adaptive strategy improves reconstruction quality across diverse scene types without incurring significant computational overhead.

Text region loss in phase 1. During phase 1, optimization is restricted to text Gaussians and masked text regions. As described in phase 1, we use an $\mathcal{L}_1$ loss restricted to text masks and omit the D-SSIM component by setting $\lambda_{\text{D-SSIM}}$ to 0 (default value of 0.2 in 3DGS). D-SSIM, being a perceptual metric over spatial patches, is unreliable when supervision is confined to sparse or irregular regions such as text boundaries. The $\mathcal{L}_1$ loss ensures stable and interpretable gradients during refinement.

Modulating positional learning rates in phase 2. Since non-text Gaussians are initialized at the beginning of this phase, to align the learning rates of their non-positional parameters like opacity, scale or spherical harmonics, with similar learning rates used in vanilla 3DGS, we shift the base learning rate schedule $\eta_{\text{base}}(t)$ by T_1 iterations. This ensures that non-text Gaussians' parameters are learned consistently, as if they had been optimized with a higher learning rate just after initialization, similar to vanilla 3DGS.

Since we want to maintain a lower learning rate for text Gaussians to preserve their well-initialized positions, and at the same time ensure compatibility with the learning rate for non-text Gaussians, we set $\alpha = 0.5$. This constant learning rate factor for the positions of non-text Gaussians ensures that the updates for non-text regions do not conflict with the more conservative updates for text regions, particularly at the boundaries where overlapping Gaussians may receive differing gradients. Experimental results show that $\alpha = 0.5$ provides an optimal balance, stabilizing non-text regions without impeding the refinement of text regions. The same value of α for both the constant learning rate factor for non-text Gaussians and the sigmoid cap for text Gaussians ensures that, in the later stages of training, text and non-text Gaussians are updated with comparable learning rates, enabling a unified refinement of the overall scene.

For text Gaussians, we introduce a sigmoid-based positional learning rate schedule to gradually change their learning rate over time. The steepness of this sigmoid curve is controlled by the parameter $\beta = 0.0005$, which determines how smoothly the learning rate changes as training progresses. A small value for β ensures that the transition is gradual, allowing for conservative updates in the early

STRinGS-360	ISO	f-number	Focal Length	Shutter Speed	Num Images	Width (px)	Height (px)
Extinguisher	2000	f/10	30 mm	1/15 s	109	1000	1500
Books	800	f/9	31 mm	1/10 s	134	1500	1000
Chemicals	2500	f/8	23 mm	1/30 s	117	1500	1000
Globe	1000	f/8	31 mm	1/30 s	205	1500	1000
Shelf	800	f/9	31 mm	1/10 s	167	1500	1000

Table 5. Metadata for the STRinGS-360 dataset containing text-rich scenes.

stages to preserve the geometry of the text regions. We observe that varying β has negligible impact on overall performance, but a smooth increase is essential to avoid abrupt changes that could destabilize the positions of text Gaussians. The parameter $\gamma = 15000$ was chosen based on experimental results, where we found that allowing the positional learning rate to remain low until this point helps preserve the initial structure of the text Gaussians.

Training efficiency. All experiments were conducted using the accelerated 3DGS-Accel framework for faster training and GPU efficiency. All reported metrics, including training time are measured under this setup.

B. STRinGS-360 Dataset Details

All scenes in the STRinGS-360 dataset were captured using a Nikon D5300 DSLR camera equipped with an 18–55mm lens. To ensure photometric consistency and maintain high reconstruction quality, we fixed the camera to use the sRGB IEC61966-2.1 color profile and captured all images under manual exposure, manual white balance, and manual focus. These manual settings were crucial to avoid photometric inconsistency across frames, which can negatively impact 3D reconstruction methods. Scene-specific values for ISO, f-number, focal length, shutter speed, and resolution are provided in Tab. 5.

Each scene in STRinGS-360 was captured by moving around a central object, collecting images from all angles to ensure full 360-degree coverage. This stands in contrast to many existing datasets such as DL3DV-10K Benchmark, where scenes are often recorded with limited back-and-forth or partial panning trajectories. Such capture styles are insufficient for evaluating text reconstruction, especially when text spans curved or occluded surfaces. Moreover, existing datasets rarely feature foreground objects with dense, semantically meaningful text. STRinGS-360 addresses this gap by providing text-rich scenes with comprehensive multi-view coverage, enabling more realistic and rigorous evaluation of text fidelity in 3D reconstruction.

The dataset comprises five indoor scenes selected for their diverse geometric and textual characteristics. Each scene contains semantically meaningful text on a central object, designed to test different aspects of text fidelity in 3D reconstruction. *Extinguisher* includes instructional text wrapped around a cylindrical surface, introducing perspective distortion and non-planar geometry. *Books* presents a flat arrangement of books densely populated with titles and author names, featuring occlusions. *Chemicals* contains high-frequency chemical labels on chemical bottles including glossy bottle surfaces. *Globe* captures a spherical surface densely annotated with geographical labels, testing the model's ability to preserve curved text across changing orientations. *Shelf* includes repeated textual patterns and occlusions in a realistic setting, resembling a complex academic bookshelf.

Text is particularly sensitive to distortions, misalignments, and blurring, all of which are common challenging modes in existing GS pipelines. Unlike geometric details or surface textures, even small inaccuracies in letter shapes can lead to substantial semantic loss. By focusing on fine-grained, multi-view text capture in challenging environments, STRinGS-360 provides a valuable benchmark for evaluating and advancing text aware 3D reconstruction methods.

C. OCR-Based Evaluation Details

To assess the textual fidelity of reconstructed scenes, we propose an OCR-based evaluation score that compares text recognized from reconstructed images against ground-truth captures. Both rendered outputs and ground-truth images are preprocessed using binary masks generated via Hi-SAM, which isolate text-bearing regions and suppress background clutter. This masking significantly reduces false positive detections by the OCR engine.

We employ Google Cloud Vision OCR for text recognition. Each detected text instance is associated with an oriented bounding box. To align predictions between ground-truth and renderings, we construct a bipartite graph in which nodes correspond to OCR-detected text regions in either image, and edges are added between regions with an Intersection-over-Union (IoU) exceeding a threshold (0.1 in our experiments). Connected components of this graph represent matched groups of text regions that are spatially aligned. This approach enables robust 1-to-N and N-to-1 matching between text blocks (*e.g.* when a single word in the ground truth is split into multiple detections in the rendered image). In contrast to traditional matching strategies like the Hungarian algorithm, which enforce strict one-to-one assignments based on a global cost matrix, our method accommodates more flexible and realistic many-to-many correspondences that often occurs in scene text OCR.

For each matched group, we compute the Character Error Rate (CER) using Levenshtein distance, which measures the minimum number of insertions, deletions, and substitutions required to transform one string into another. The text strings in each group are formed by concatenating the OCR results after sorting them by their spatial layout (either horizontally or vertically, based on the dominant axis), to approximate reading order.

The final CER scores are obtained by aggregating character-level errors across all matched groups, normalized by the total number of characters in the ground-truth regions. Importantly, we adopt a recall-oriented evaluation protocol, where unmatched OCR predictions in the rendered image (false positives) are excluded from error computation. This reflects a conservative evaluation focused on text retention, penalizing missing or corrupted text from the ground truth, while tolerating false predictions that do not interfere with readability. This is particularly important for reconstruction settings where hallucinated background artifacts or fragmentary text are common, and over-penalizing such cases could misrepresent actual recognition quality.

D. Additional Results

D.1. Generalization to Multilingual Text

Our method generalizes to scenes containing text in any language, owing to the robustness of Hi-SAM in generating accurate text segmentation masks across diverse scripts. While quantitative evaluation is limited by the reliability of multi-language OCR tools, qualitative results on a scene from the DL3DV-10K Benchmark dataset at 7K training iterations (Fig. 10) demonstrate that our approach effectively reconstructs text across different scripts.

D.2. OCR-CER and Text Size

Small text in rendered images plays a critical role in OCR-CER scores, as its reconstruction is particularly challenging. To investigate this, we analyze OCR-CER results stratified by text size. Following Google OCR outputs, the size of a text instance is defined as the height of the shorter side of its bounding box, and not the area as it is influenced by word length. We categorize text into two buckets: *small* and *large* text based on observations made from the distribution of the heights.

Larger text tends to be well reconstructed across all methods, whereas fine textual details are often missed, especially in the early stages of training. When comparing STRinGS with 3DGS (Tab. 6), the scores are very close for large text, but there is a substantial gap for small text, which directly drives the overall difference in OCR-CER. At 7K iterations, this gap is particularly large. For example, on the Chemicals scene, the difference in small-text CER between 3DGS and STRinGS is 0.579, while the difference for large text is 0.593. On the Globe scene, the small-text difference is 0.680, compared to 0.748 for large text. This shows that at 7K iterations, the gains from both small-text and large-text is significant. By 30K iterations, the gains for large-text become smaller, but STRinGS still maintains an advantage in small-text. On the Chemicals scene, the small-text CER differs by 0.117 between the two methods, while the large-text CER differs by only 0.009. This consistent trend across scenes shows that large text is generally easy to reconstruct for all methods, while small text remains the main challenge and the primary source of improvement for STRinGS.

D.3. Runtime breakdown

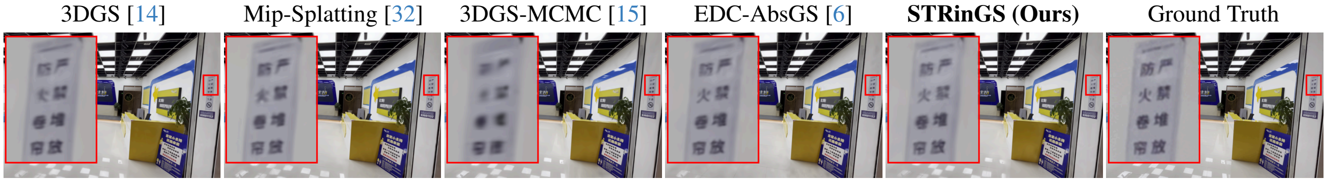
In Tab. 7, we provide a detailed runtime breakdown at the scene level, separating preprocessing and training components. Preprocessing includes both COLMAP reconstruction and text segmentation. Since existing methods report runtime only for the training stage and exclude COLMAP preprocessing, we likewise do not count the text segmentation step toward runtime, as it is a one-time preprocessing step independent of the reconstruction pipeline itself. As expected, the runtime of COLMAP varies significantly with the number of input images and their resolution. While Hi-SAM text segmentation adds a small, parallelizable cost per image, its execution can be scaled according to available compute resources. The table also contains the time breakdown of the two phases of our pipeline. The selective text reconstruction in Phase 1 is highly efficient, accelerating the emergence of fine textual details to yield high-fidelity scenes early in the training process.

D.4. Qualitative Results

Fig. 11 and Fig. 12 (7K iterations) demonstrate the ability of our method to reconstruct semantically meaningful text early in the optimization process, while Fig. 13 and Fig. 14 (30K iterations) shows further improvements, particularly in challenging text regions where sharper and more accurate reconstruction is achieved.

D.5. Quantitative Results

We report the overall quantitative performance aggregated across all scenes in each dataset in Tab. 8 and Tab. 4, which present consolidated metrics for all methods at 7K and 30K training iterations, including PSNR, SSIM, LPIPS, and the number of Gaussians. These results reflect the gen-



3DGS [14] Mip-Splatting [32] 3DGS-MCMC [15] EDC-AbsGS [6] **STRinGS (Ours)** Ground Truth

Figure 10. Qualitative comparisons of different methods at 7K training iterations on Scene 127 from the DL3DV-10K Benchmark [18] dataset with Chinese characters, showing the robustness of our method towards different languages.

Scene (STRinGS-360)	Number of characters			Method	OCR-CER (7K)			OCR-CER (30K)		
	Small text	Large text	Total		Small text	Large text	Total	Small text	Large text	Total
Shelf	11110	4352	15462	3DGS	0.843	0.575	0.765	0.125	0.075	0.107
				STRinGS	0.148	0.074	0.118	0.115	0.063	0.096
Books	5382	3371	8753	3DGS	0.457	0.171	0.343	0.080	0.053	0.060
				STRinGS	0.087	0.062	0.072	0.057	0.044	0.047
Extinguisher	3404	1105	4509	3DGS	0.790	0.430	0.696	0.127	0.032	0.095
				STRinGS	0.144	0.034	0.109	0.081	0.026	0.060
Chemicals	9120	2371	11491	3DGS	0.990	0.676	0.923	0.294	0.059	0.239
				STRinGS	0.411	0.083	0.337	0.177	0.050	0.144
Globe	19018	5056	24074	3DGS	0.957	0.939	0.953	0.270	0.143	0.238
				STRinGS	0.277	0.191	0.251	0.210	0.114	0.182

Table 6. OCR-CER comparison between 3DGS and STRinGS at 7K and 30K iterations, stratified by text size. Each scene lists the number of characters (divided into small, large, and total). Text is divided into the same categories for evaluation, and OCR-CER is reported separately for these bins as well as for the overall scene.

eral reconstruction quality of the scene. Across these standard image-based metrics, our method performs competitively with existing approaches. In some cases, we observe slightly lower PSNR, SSIM, or LPIPS scores, which we attribute to our deliberate emphasis on accurate text reconstruction, at the expense of background regions that typically dominate these global averages. This trade-off is especially relevant in scenes where semantically important text occupies only a small portion of the image and does not significantly influence metrics that are averaged across the entire scene. Notably, our method achieves this while maintaining a relatively low number of Gaussians compared to other methods, especially in text-rich scenes like in our STRinGS-360 dataset, highlighting its efficiency.

We complement these results with detailed per-scene evaluations. Tab. 9 presents OCR-based Character Error Rate (CER), Tab. 10 presents training time, Tab. 11 presents PSNR, Tab. 12 presents SSIM, Tab. 13 presents LPIPS, and Tab. 14 presents the number of Gaussians.

Dataset	**Scene**	**Preprocessing**		**Training**		
		COLMAP	Hi-SAM (per image)	Phase 1	Phase 2	Total
Tanks and Temples	Train	6.3 m	3.5 s	0.3 m	7.7 m	8.0 m
	Truck	4.0 m	2.4 s	0.2 m	11.0 m	11.2 m
	Avg	5.1 m	2.9 s	0.2 m	9.4 m	9.6 m
DL3DV-10K Benchmark	Scene 3	23.5 m	3.9 s	0.9 m	12.2 m	13.1 m
	Scene 21	26.7 m	2.8 s	0.5 m	11.3 m	11.8 m
	Scene 80	17.7 m	3.4 s	0.7 m	8.5 m	9.2 m
	Scene 92	18.2 m	2.8 s	0.6 m	12.7 m	13.3 m
	Scene 107	24.8 m	3.8 s	1.2 m	13.6 m	14.8 m
	Scene 132	9.5 m	3.0 s	0.5 m	7.6 m	8.1 m
	Scene 136	20.0 m	3.8 s	0.6 m	8.9 m	9.5 m
	Avg	20.0 m	3.3 s	0.7 m	10.7 m	11.4 m
STRinGS-360 (Ours)	Shelf	3.3 m	3.9 s	1.0 m	18.0 m	19.0 m
	Books	2.5 m	3.9 s	0.7 m	12.9 m	13.6 m
	Extinguisher	2.1 m	2.8 s	0.4 m	10.4 m	10.8 m
	Chemicals	2.6 m	3.9 s	0.5 m	9.2 m	9.7 m
	Globe	4.1 m	4.1 s	0.7 m	9.3 m	10.0 m
	Avg	2.9 m	3.7 s	0.7 m	11.9 m	12.6 m

Table 7. Scene-level runtime breakdown for preprocessing and training components for STRinGS pipeline. The *Total* column corresponds to training time only (consistent with Tab. 2 at 30K iterations). Preprocessing includes COLMAP and Hi-SAM inference time per image, which is treated separately from training time. Time taken by COLMAP depends on the number of input images and their resolution and hence varies significantly across scenes. HI-SAM can be parallelized across compute. Note that the Hi-SAM model roughly takes 3 seconds to load which is not included in the values above.

Method	Tanks&Temples				DL3DV-10K Benchmark				STRinGS-360 (Ours)			
	PSNR↑	SSIM↑	LPIPS↓	Points↓	PSNR↑	SSIM↑	LPIPS↓	Points↓	PSNR↑	SSIM↑	LPIPS↓	Points↓
3DGS SIGGRAPH'23	21.88	0.7873	0.2556	1217K	27.34	0.8980	0.1974	974K	25.71	0.8508	0.3048	1004K
Mip-Splatting CVPR'24	21.90	0.7945	0.2472	1617K	27.33	0.9002	0.1903	1226K	25.74	0.8513	0.3018	1314K
3DGS-MCMC NeurIPS'24	21.55	0.7565	0.2934	1550K	26.49	0.8782	0.2221	1182K	23.56	0.7907	0.3829	1388K
AbsGS ACMMM'24	21.64	0.7701	0.2725	1235K	26.82	0.8880	0.2090	963K	25.26	0.8356	0.3260	1283K
EDC-AbsGS arXiv'25	22.36	0.8202	0.2148	1142K	27.99	0.9171	0.1682	780K	27.64	0.8936	0.2434	981K
STRinGS (Ours)	21.14	0.7756	0.2740	879K	27.02	0.9005	0.1981	790K	26.65	0.8841	0.2703	790K

Table 8. Comparison of reconstruction quality and number of Gaussians (Points) at 7K iterations across three datasets: Tanks&Temples [17], DL3DV-10K Benchmark [18], and STRinGS-360.

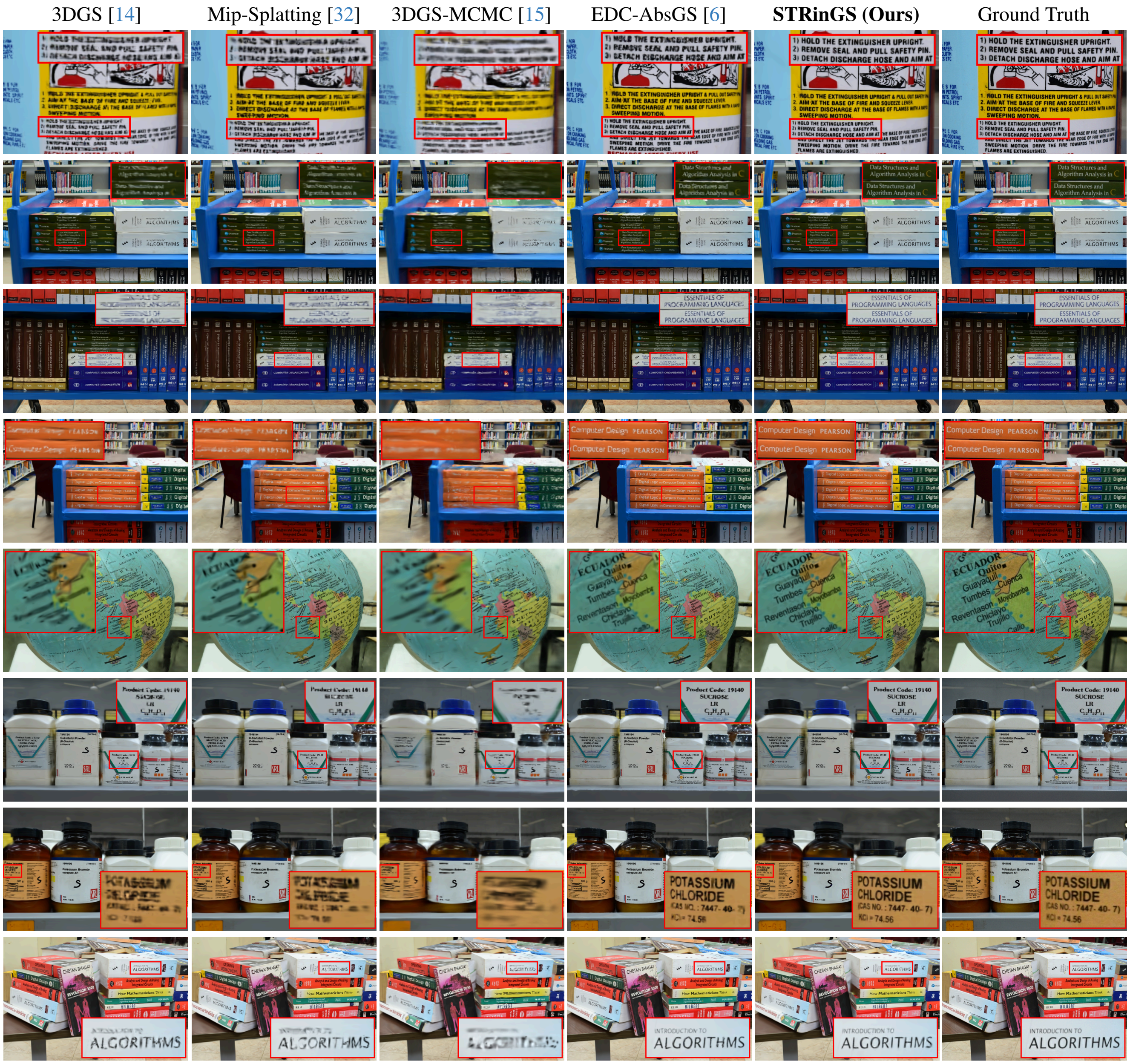


Figure 11. Qualitative comparisons of different methods at 7K training iterations on scenes from our STRinGS-360 dataset.

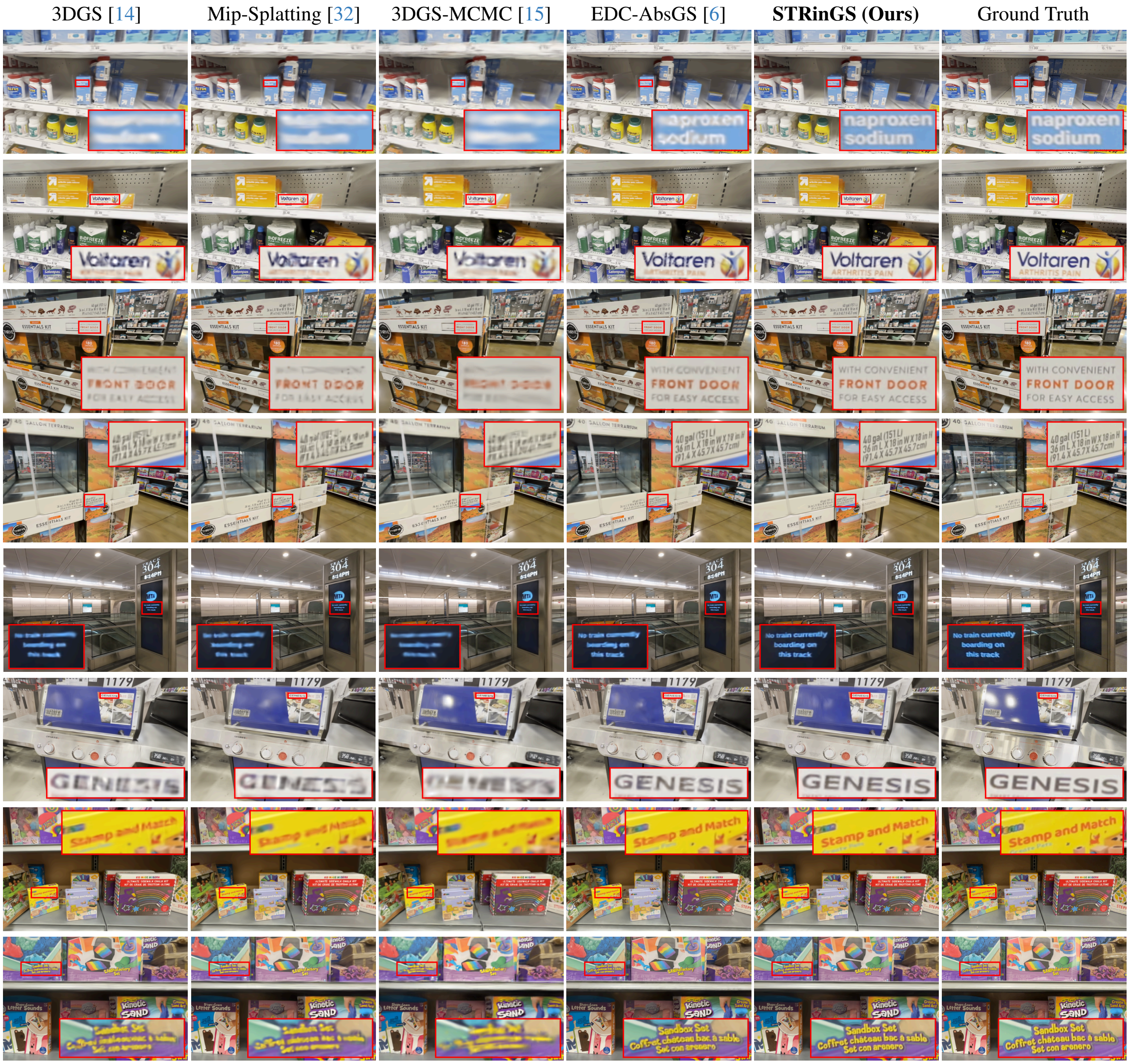


Figure 12. Qualitative comparisons of different methods at 7K training iterations on scenes from the DL3DV-10K Benchmark [18] dataset.

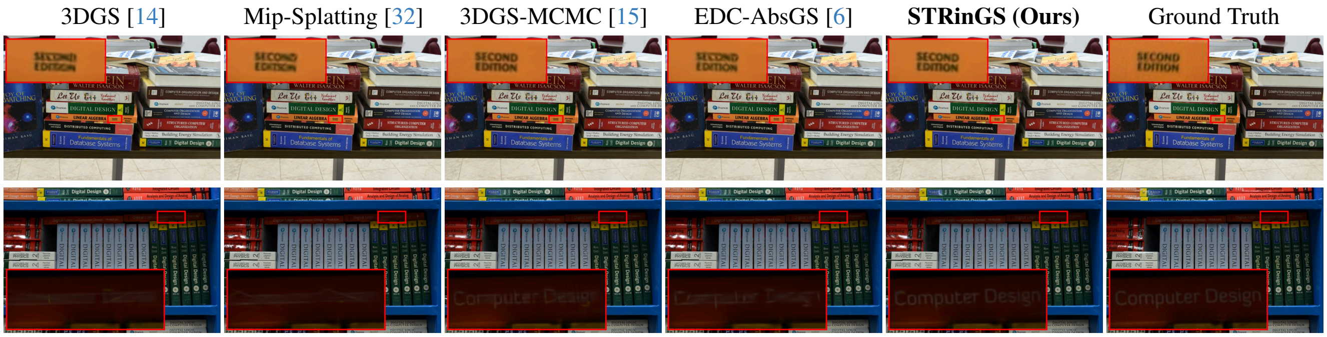


Figure 13. Qualitative comparisons of different methods at 30K training iterations on scenes from our STRinGS-360 dataset.

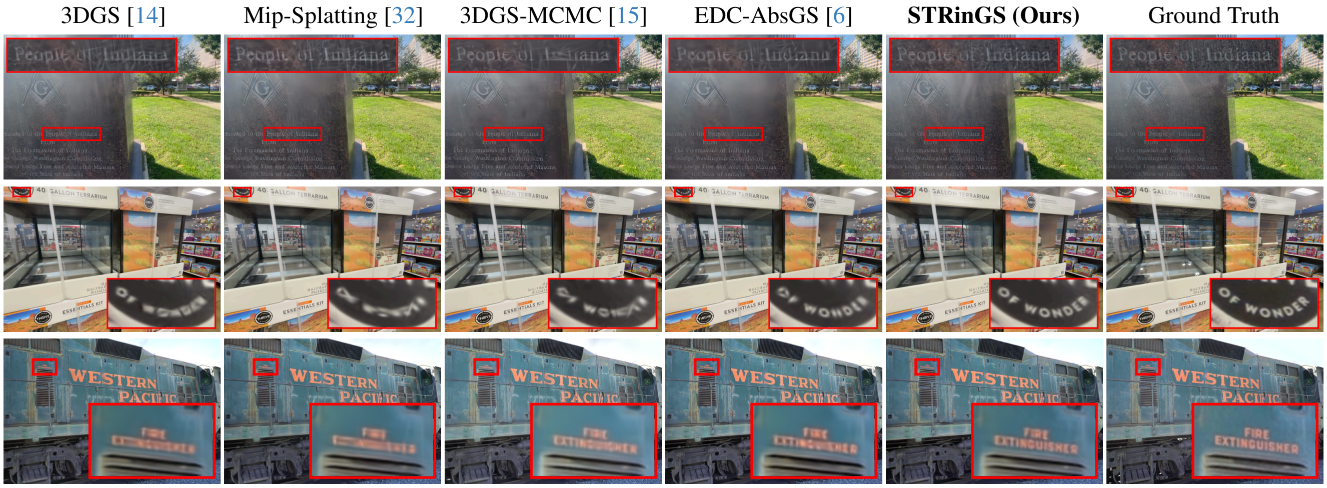


Figure 14. Qualitative comparisons of different methods at 30K training iterations on scenes from the DL3DV-10K Benchmark [18] and Tanks&Temples [17] datasets.

OCR-CER ↓	Scene	3DGS		Mip-Splatting		3DGS-MCMC		AbsGS		EDC-AbsGS		**STRinGS (Ours)**	
		7K	30K	7K	30K	7K	30K	7K	30K	7K	30K	7K	30K
Tanks and Temples	Train	0.282	0.161	0.304	0.174	0.322	0.152	0.327	0.192	0.190	0.170	0.143	0.128
	Truck	0.136	0.080	0.140	0.076	0.221	0.088	0.171	0.082	0.094	0.066	0.100	0.070
	Avg	0.209	0.121	0.222	0.125	0.272	0.120	0.249	0.137	0.142	0.118	0.122	0.099
DL3DV-10K Benchmark	Scene 3	0.403	0.142	0.411	0.124	0.640	0.128	0.439	0.138	0.226	0.149	0.220	0.124
	Scene 21	0.475	0.178	0.382	0.152	0.531	0.160	0.481	0.186	0.271	0.184	0.190	0.136
	Scene 80	0.474	0.170	0.500	0.180	0.608	0.134	0.517	0.194	0.293	0.204	0.162	0.099
	Scene 92	0.416	0.093	0.471	0.096	0.604	0.100	0.476	0.099	0.179	0.103	0.144	0.090
	Scene 107	0.587	0.223	0.600	0.221	0.718	0.195	0.602	0.232	0.386	0.211	0.271	0.175
	Scene 132	0.240	0.172	0.231	0.171	0.263	0.174	0.219	0.159	0.202	0.181	0.208	0.153
	Scene 136	0.151	0.122	0.151	0.102	0.216	0.100	0.160	0.108	0.119	0.105	0.113	0.085
	Avg	0.392	0.157	0.392	0.149	0.511	0.142	0.411	0.160	0.239	0.162	0.187	0.123
STRinGS-360 (Ours)	Shelf	0.765	0.107	0.752	0.095	0.990	0.096	0.810	0.108	0.262	0.097	0.118	0.096
	Books	0.343	0.060	0.348	0.051	0.749	0.054	0.380	0.066	0.073	0.054	0.072	0.047
	Extinguisher	0.696	0.095	0.748	0.069	0.900	0.067	0.738	0.094	0.207	0.082	0.109	0.060
	Chemicals	0.923	0.239	0.934	0.226	0.999	0.154	0.945	0.209	0.626	0.157	0.337	0.144
	Globe	0.953	0.238	0.960	0.204	0.998	0.177	0.968	0.240	0.469	0.190	0.251	0.182
	Avg	0.736	0.148	0.748	0.129	0.927	0.110	0.768	0.143	0.328	0.116	0.177	0.106

Table 9. OCR-CER (↓) comparison across scenes and methods at 7K / 30K iterations.

Training time ↓	Scene	3DGS		Mip-Splatting		3DGS-MCMC		AbsGS		EDC-AbsGS		**STRinGS (Ours)**	
		7K	30K	7K	30K	7K	30K	7K	30K	7K	30K	7K	30K
Tanks and Temples	Train	1.6 m	11.2 m	2.8 m	16.8 m	3.0 m	15.8 m	2.4 m	11.4 m	2.6 m	11.5 m	1.0 m	8.0 m
	Truck	2.5 m	16.4 m	3.9 m	24.8 m	3.2 m	23.5 m	2.9 m	14.0 m	3.0 m	13.9 m	1.2 m	11.2 m
	Avg	2.0 m	13.8 m	3.4 m	20.8 m	3.1 m	19.6 m	2.6 m	12.7 m	2.8 m	12.7 m	1.1 m	9.6 m
DL3DV-10K Benchmark	Scene 3	3.2 m	20.7 m	6.7 m	36.7 m	6.0 m	36.0 m	5.5 m	22.3 m	5.4 m	21.3 m	2.4 m	13.1 m
	Scene 21	2.8 m	15.5 m	6.3 m	32.5 m	4.8 m	26.7 m	5.4 m	23.1 m	8.0 m	27.0 m	1.8 m	11.8 m
	Scene 80	2.2 m	10.6 m	5.2 m	22.5 m	4.2 m	20.9 m	4.6 m	17.6 m	5.0 m	19.0 m	1.9 m	9.2 m
	Scene 92	3.0 m	17.2 m	7.0 m	35.0 m	5.8 m	31.6 m	5.5 m	21.7 m	5.9 m	23.0 m	2.1 m	13.3 m
	Scene 107	3.6 m	20.1 m	7.0 m	33.7 m	6.6 m	34.7 m	5.8 m	22.1 m	6.0 m	22.1 m	3.2 m	14.8 m
	Scene 132	2.2 m	9.5 m	6.4 m	26.0 m	4.4 m	23.2 m	5.4 m	19.4 m	6.1 m	22.0 m	1.6 m	8.1 m
	Scene 136	2.4 m	11.9 m	5.8 m	26.8 m	4.8 m	24.9 m	5.0 m	19.0 m	5.5 m	21.2 m	1.8 m	9.5 m
	Avg	2.8 m	15.1 m	6.3 m	30.4 m	5.2 m	28.3 m	5.3 m	20.7 m	6.0 m	22.2 m	2.1 m	11.4 m
STRinGS-360 (Ours)	Shelf	3.2 m	23.25 m	6.6 m	40.2 m	5.2 m	39.2 m	6.2 m	26.8 m	5.9 m	25.3 m	2.9 m	19.0 m
	Books	2.7 m	18.0 m	5.6 m	31.2 m	5.6 m	39.7 m	5.2 m	21.1 m	5.2 m	21.2 m	2.0 m	13.6 m
	Extinguisher	2.7 m	17.5 m	6.0 m	32.6 m	4.8 m	32.2 m	5.8 m	23.8 m	5.6 m	24.5 m	1.4 m	10.8 m
	Chemicals	2.0 m	14.8 m	5.3 m	28.3 m	5.5 m	28.6 m	4.9 m	20.2 m	5.5 m	22.8 m	1.5 m	9.7 m
	Globe	1.9 m	12.3 m	4.9 m	26.1 m	6.8 m	30.2 m	4.8 m	20.4 m	5.2 m	22.2 m	1.9 m	10.0 m
	Avg	2.5 m	17.2 m	5.7 m	31.7 m	5.6 m	34.0 m	5.4 m	22.5 m	5.5 m	23.2 m	1.9 m	12.6 m

Table 10. Training time (↓) (on RTX 3090 Ti) comparison across scenes and methods at 7K / 30K iterations.

PSNR ↑	Scene	3DGS		Mip-Splatting		3DGS-MCMC		AbsGS		EDC-AbsGS		**STRinGS (Ours)**	
		7K	30K	7K	30K	7K	30K	7K	30K	7K	30K	7K	30K
Tanks and Temples	Train	19.75	21.97	19.58	21.92	19.37	22.61	19.15	21.61	20.03	21.67	19.16	22.32
	Truck	24.00	25.49	24.23	25.70	23.73	26.25	24.13	25.67	24.68	25.78	23.12	25.44
	Avg	21.88	23.73	21.90	23.81	21.55	24.43	21.64	23.64	22.36	23.73	21.14	23.88
DL3DV-10K Benchmark	Scene 3	29.01	33.35	29.03	33.61	26.92	33.52	27.94	33.28	30.44	33.56	29.31	33.02
	Scene 21	25.08	26.87	25.23	27.35	25.74	27.87	25.34	27.40	25.64	27.75	24.03	26.92
	Scene 80	29.51	31.92	29.85	32.35	28.86	31.45	29.36	31.92	29.76	32.06	28.46	31.80
	Scene 92	23.66	26.10	23.41	25.86	22.78	26.36	22.99	26.05	23.97	26.19	23.03	25.93
	Scene 107	28.08	33.20	27.98	33.82	26.10	33.58	26.73	33.26	29.67	33.42	29.54	33.37
	Scene 132	29.30	31.24	29.43	31.46	28.54	31.15	28.90	31.05	21.53	31.27	28.55	31.09
	Scene 136	26.72	28.74	26.40	28.82	26.46	29.28	26.40	28.32	26.90	28.94	26.22	28.82
	Avg	27.34	30.20	27.33	30.47	26.49	30.46	26.82	30.18	27.99	30.45	27.02	30.14
STRinGS-360 (Ours)	Shelf	24.66	29.55	24.44	29.02	21.43	29.74	23.58	29.26	27.02	29.85	27.18	29.57
	Books	25.68	28.82	25.75	28.79	23.64	29.30	24.73	28.71	27.69	29.15	26.41	28.50
	Extinguisher	28.61	30.66	28.38	30.49	27.00	31.95	28.48	30.77	29.92	31.21	29.03	30.80
	Chemicals	24.85	28.52	25.19	28.86	22.80	29.95	24.74	28.65	27.55	29.32	25.68	28.82
	Globe	24.77	26.71	24.93	26.84	22.94	28.32	24.76	26.44	26.02	26.99	24.96	27.28
	Avg	25.71	28.85	25.74	28.80	23.56	29.85	25.26	28.77	27.64	29.30	26.65	29.00

Table 11. PSNR (↑) comparison across scenes and methods at 7K / 30K iterations.

SSIM ↑	Scene	3DGS		Mip-Splatting		3DGS-MCMC		AbsGS		EDC-AbsGS		**STRinGS (Ours)**	
		7K	30K	7K	30K	7K	30K	7K	30K	7K	30K	7K	30K
Tanks and Temples	Train	0.7211	0.8199	0.7256	0.8264	0.6845	0.8404	0.6887	0.8179	0.7659	0.8289	0.7147	0.8201
	Truck	0.8536	0.8850	0.8634	0.8927	0.8286	0.8972	0.8516	0.8874	0.8746	0.8900	0.8365	0.8825
	Avg	0.7873	0.8524	0.7945	0.8596	0.7565	0.7688	0.7701	0.8526	0.8202	0.8595	0.7756	0.8513
DL3DV-10K Benchmark	Scene 3	0.9144	0.9595	0.9138	0.9620	0.8742	0.9619	0.8964	0.9600	0.9404	0.9618	0.9287	0.9578
	Scene 21	0.8204	0.8613	0.8321	0.8750	0.8204	0.8797	0.8280	0.8705	0.8518	0.8826	0.8035	0.8584
	Scene 80	0.9387	0.9570	0.9423	0.9605	0.9311	0.9564	0.9372	0.9579	0.9479	0.9595	0.9353	0.9566
	Scene 92	0.8584	0.9083	0.8560	0.9107	0.8304	0.9134	0.8376	0.9075	0.8769	0.9125	0.8556	0.9064
	Scene 107	0.8950	0.9587	0.8956	0.9626	0.8600	0.9642	0.8707	0.9600	0.9292	0.9619	0.9279	0.9602
	Scene 132	0.9295	0.9476	0.9308	0.9495	0.9134	0.9449	0.9223	0.9661	0.9357	0.9484	0.9238	0.9464
	Scene 136	0.9298	0.9509	0.9308	0.9529	0.9182	0.9527	0.9236	0.9502	0.9381	0.9532	0.9286	0.9507
	Avg	0.8980	0.9348	0.9002	0.9390	0.8782	0.9390	0.8880	0.9360	0.9171	0.9400	0.9005	0.9338
STRinGS-360 (Ours)	Shelf	0.8305	0.9276	0.8307	0.9258	0.7349	0.9324	0.8058	0.9263	0.8918	0.9323	0.8916	0.9268
	Books	0.8815	0.9289	0.8806	0.9301	0.8287	0.9362	0.8617	0.9283	0.9180	0.9333	0.9031	0.9266
	Extinguisher	0.8673	0.9050	0.8664	0.9064	0.8177	0.9193	0.8602	0.9024	0.8976	0.9134	0.8784	0.9073
	Chemicals	0.8527	0.9089	0.8556	0.9118	0.7983	0.9187	0.8414	0.9078	0.8951	0.9134	0.8793	0.9100
	Globe	0.8218	0.8925	0.8232	0.8969	0.7738	0.9106	0.8089	0.8905	0.8655	0.8991	0.8681	0.8982
	Avg	0.8508	0.9126	0.8513	0.9142	0.7907	0.9234	0.8356	0.9111	0.8936	0.9183	0.8841	0.9138

Table 12. SSIM (↑) comparison across scenes and methods at 7K / 30K iterations.

LPIPS ↓	Scene	3DGS		Mip-Splatting		3DGS-MCMC		AbsGS		EDC-AbsGS		**STRinGS (Ours)**	
		7K	30K	7K	30K	7K	30K	7K	30K	7K	30K	7K	30K
Tanks and Temples	Train	0.3164	0.1961	0.3144	0.1892	0.3624	0.1844	0.3536	0.1916	0.2667	0.1816	0.3306	0.2032
	Truck	0.1948	0.1422	0.1800	0.1234	0.2244	0.1171	0.1913	0.1316	0.1628	0.1299	0.2174	0.1502
	Avg	0.2556	0.1692	0.2472	0.1563	0.2934	0.1508	0.2725	0.1616	0.2148	0.1557	0.2740	0.1767
DL3DV-10K Benchmark	Scene 3	0.1453	0.0843	0.1432	0.0768	0.1928	0.0804	0.1720	0.0786	0.1088	0.0776	0.1307	0.0882
	Scene 21	0.2506	0.2002	0.2326	0.1779	0.2483	0.1775	0.2297	0.1817	0.2078	0.1688	0.2753	0.2044
	Scene 80	0.1898	0.1576	0.1732	0.1378	0.1984	0.1582	0.1812	0.1431	0.1664	0.1398	0.1976	0.1585
	Scene 92	0.2201	0.1517	0.2198	0.1438	0.2577	0.1498	0.2556	0.1495	0.1966	0.1435	0.2280	0.1556
	Scene 107	0.1926	0.1070	0.1934	0.0961	0.2310	0.0901	0.2274	0.0982	0.1475	0.0958	0.1568	0.1032
	Scene 132	0.1903	0.1576	0.1869	0.1514	0.2220	0.1691	0.2014	0.1572	0.1786	0.1546	0.2005	0.1609
	Scene 136	0.1933	0.1608	0.1832	0.1463	0.2045	0.1509	0.1954	0.1495	0.1718	0.1446	0.1974	0.1630
	Avg	0.1974	0.1456	0.1903	0.1329	0.2221	0.1394	0.2090	0.1368	0.1682	0.1321	0.1981	0.1477
STRinGS-360 (Ours)	Shelf	0.2783	0.1508	0.2746	0.1433	0.3959	0.1408	0.3223	0.1451	0.1998	0.1404	0.2102	0.1567
	Books	0.2271	0.1556	0.2217	0.1450	0.2893	0.1417	0.2481	0.1471	0.1774	0.1456	0.2138	0.1666
	Extiguisher	0.2733	0.2075	0.2647	0.1897	0.3474	0.1928	0.2725	0.1957	0.2199	0.1864	0.2739	0.2166
	Chemicals	0.3723	0.2727	0.3708	0.2666	0.4454	0.2652	0.3960	0.2653	0.3074	0.2638	0.3400	0.2800
	Globe	0.3730	0.2668	0.3773	0.2614	0.4363	0.2451	0.3912	0.2688	0.3122	0.2600	0.3137	0.2630
	Avg	0.3048	0.2107	0.3018	0.2012	0.3829	0.1971	0.3260	0.2044	0.2434	0.1992	0.2703	0.2166

Table 13. LPIPS (↓) comparison across scenes and methods at 7K / 30K iterations.

Number of Gaussians ↓	Scene	3DGS		Mip-Splatting		3DGS-MCMC		AbsGS		EDC-AbsGS		**STRinGS (Ours)**	
		7K	30K	7K	30K	7K	30K	7K	30K	7K	30K	7K	30K
Tanks and Temples	Train	741K	1093K	883K	1481K	1100K	1100K	836K	949K	843K	1064K	554K	893K
	Truck	1693K	2059K	2351K	3250K	2000K	2000K	1634K	1646K	1440K	1700K	1204K	1816K
	Avg	1217K	1576K	1617K	2366K	1550K	1550K	1235K	1297K	1142K	1382K	879K	1354K
DL3DV-10K Benchmark	Scene 3	1509K	1872K	1804K	2407K	1900K	1900K	1470K	1198K	881K	923K	991K	1109K
	Scene 21	1308K	1509K	1973K	2431K	1500K	1500K	1486K	1474K	1522K	1743K	1029K	1349K
	Scene 80	622K	664K	764K	891K	665K	665K	566K	517K	553K	585K	633K	658K
	Scene 92	1082K	1393K	1322K	1968K	1400K	1400K	1022K	992K	857K	995K	878K	1177K
	Scene 107	1314K	1682K	1472K	2026K	1700K	1700K	1272K	1085K	830K	849K	1170K	1187K
	Scene 132	368K	398K	489K	607K	400K	400K	318K	291K	298K	326K	317K	357K
	Scene 136	618K	710K	757K	936K	710K	710K	609K	563K	524K	579K	518K	587K
	Avg	974K	1175K	1226K	1610K	1182K	1182K	963K	874K	780K	857K	790K	918K
STRinGS-360 (Ours)	Shelf	1448K	2079K	1904K	2829K	2080K	2080K	1942K	1903K	1399K	1403K	1192K	1434K
	Books	916K	1231K	1086K	1526K	1230K	1230K	940K	818K	602K	617K	634K	721K
	Extinguisher	1397K	1616K	2028K	2413K	1610K	1610K	1700K	1564K	1336K	1434K	903K	1137K
	Chemicals	564K	1007K	657K	1233K	1000K	1000K	831K	924K	748K	754K	464K	608K
	Globe	694K	1021K	893K	1373K	1020K	1020K	1002K	990K	818K	1000K	759K	925K
	Avg	1004K	1391K	1314K	1875K	1388K	1388K	1283K	1240K	981K	1041K	790K	965K

Table 14. Number of Gaussians (↓) comparison across scenes and methods at 7K / 30K iterations.