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

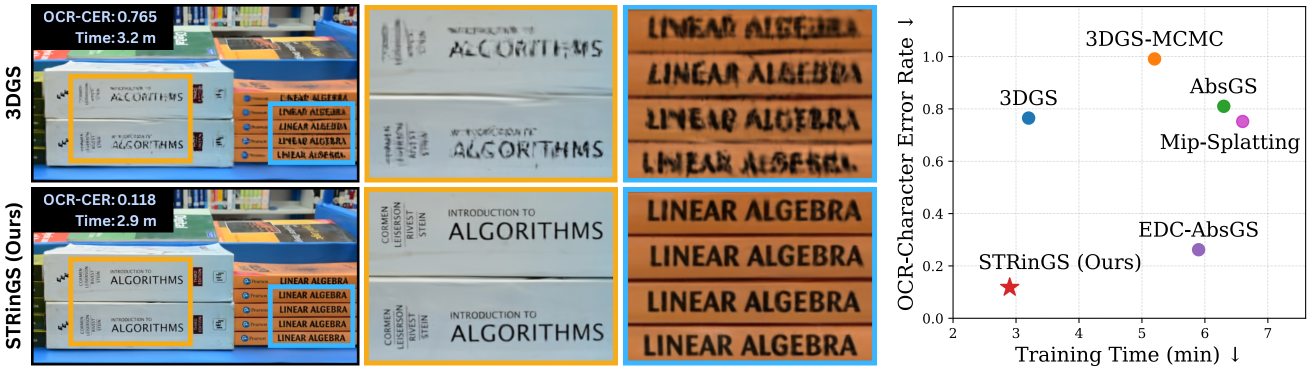

Figure 1. Qualitative and quantitative comparison of Gaussian Splatting methods on text reconstruction at 7K iterations. **Left:** On a novel view from the *Shelf* dataset that features library books on a shelf, our approach STRinGS (bottom) produces sharper and readable text as compared to vanilla 3DGS (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 photorealistic 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-

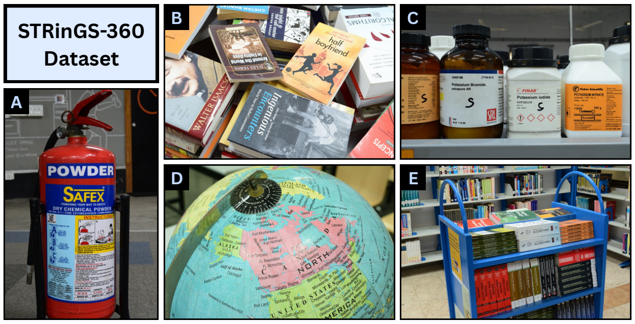

Figure 2. Overview of the scenes in our STRinGS-360 dataset. Each scene contains semantically meaningful text elements: (A) Extinguisher, (B) Books, (C) Chemicals, (D) Globe, and (E) Shelf. The dataset is designed to evaluate text reconstruction performance under diverse layouts and text orientations.

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

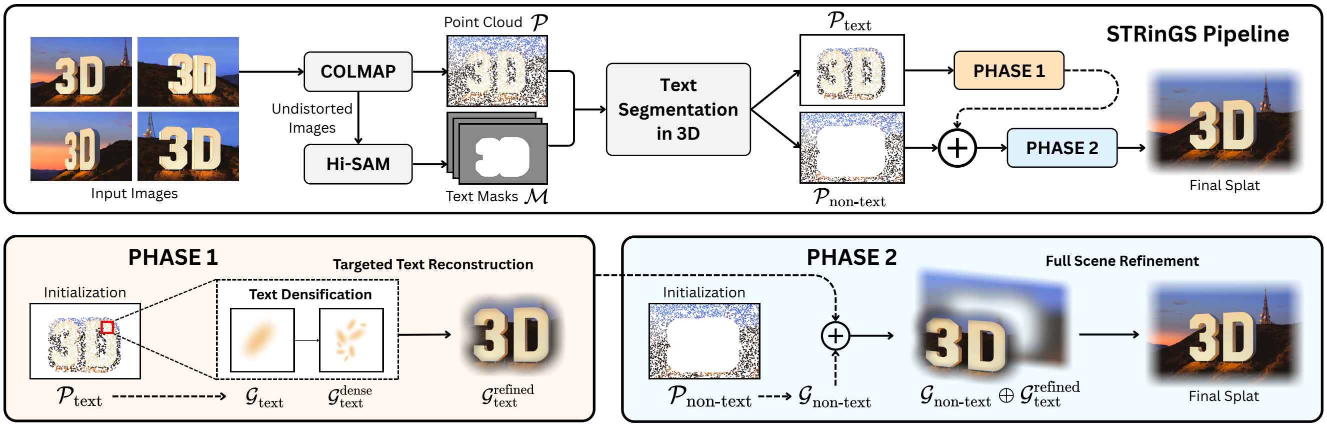
STRinGS Pipeline
Input Images
COLMAP
Undistorted Images
Hi-SAM
Point Cloud $\mathcal{P}$
Text Masks $\mathcal{M}$
Text Segmentation in 3D
$\mathcal{P}_{\text{text}}$
$\mathcal{P}_{\text{non-text}}$
PHASE 1
PHASE 2
Final Splat
PHASE 1
Targeted Text Reconstruction
Initialization
Text Densification
$\mathcal{P}_{\text{text}}$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

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

$c_i = |V_i|$ 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:

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

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

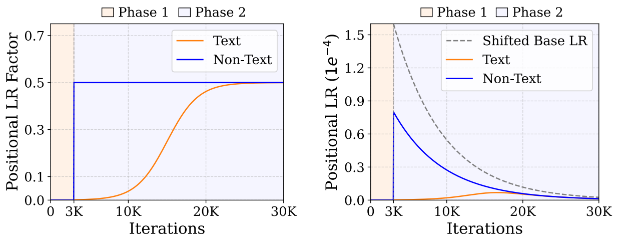

Figure 4. Learning rate (LR) of the position parameter for Gaussians in STRinGS (see Eq. (3)). **Left:** Learning rate scaling factor $\eta_r(t)$ for text and non-text Gaussians. **Right:** Effective LR obtained by modulating a shifted base exponential decay schedule $\eta_{\text{opt}}(t)$ from 3DGS 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.

Training Time (in minutes) ↓	TandT		DL3DV-10K		STRinGS-360	
	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.

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.

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.

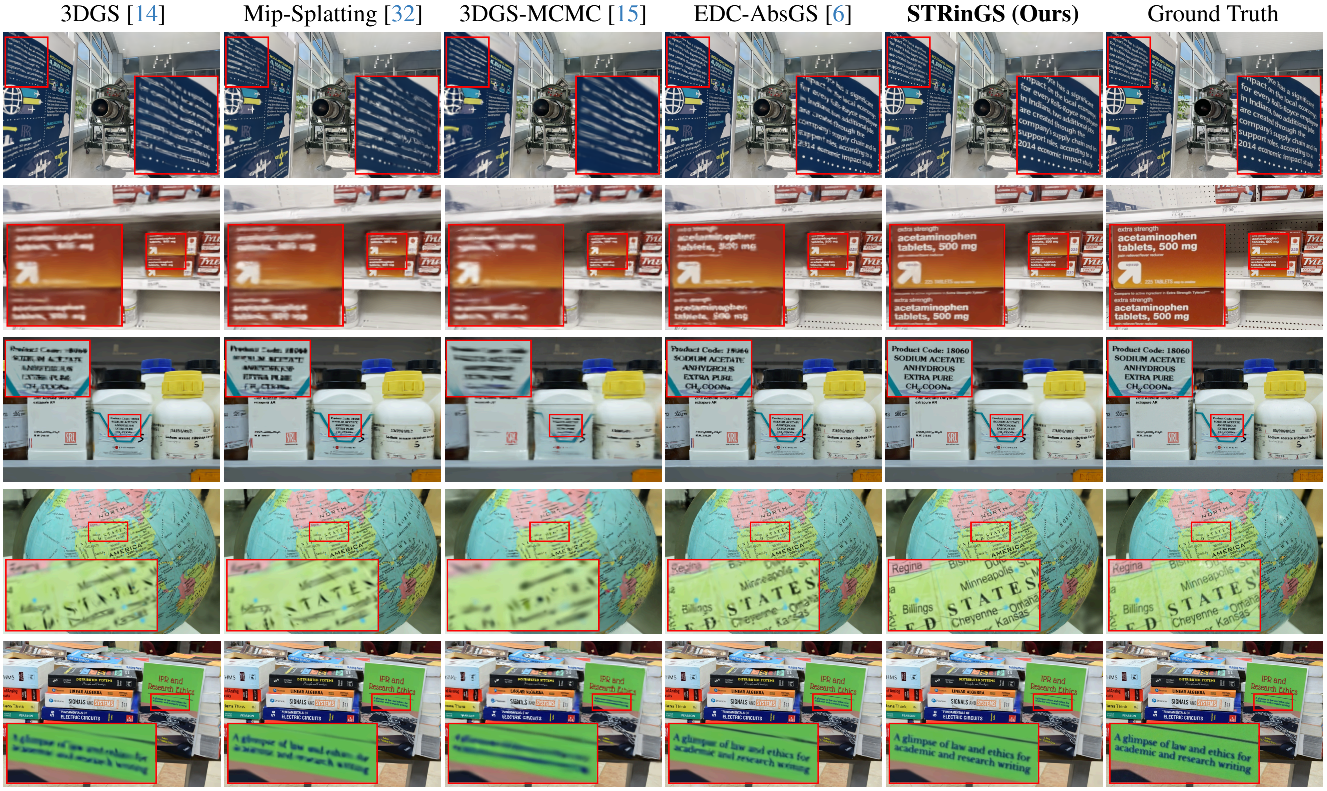

Figure 5. Qualitative comparison of different methods at 7K training iterations on scenes from the DL3DV-10K Benchmark [18] (rows 1, 2) and our STRinGS-360 (rows 3-5) datasets. While existing methods struggle to reconstruct text accurately at this early stage, our STRinGS framework produces significantly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

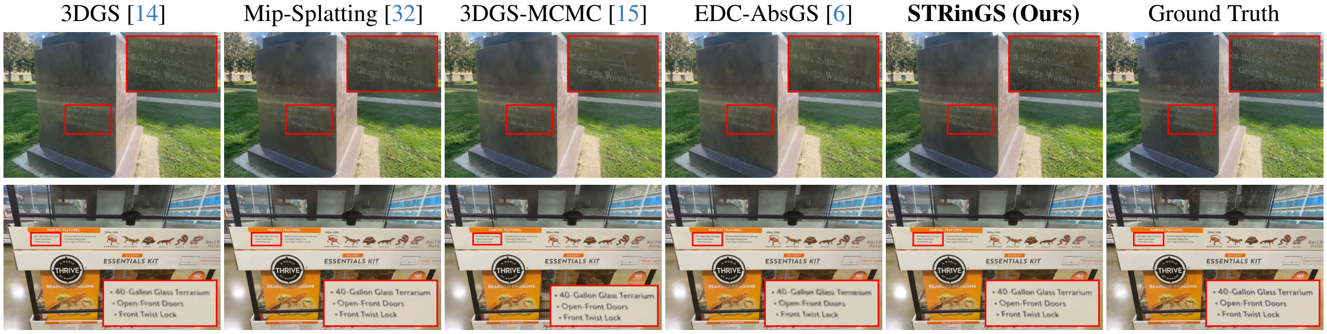

Figure 6. Qualitative comparison of different methods at 30K training iterations on scenes from DL3DV-10K Benchmark [18]. STRinGS consistently preserves text clarity, even in visually challenging regions where other methods miss fine textual details. (Best seen on screen)

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 ↓ (7K iterations)	w/o	Ours	w/o	Ours	w/o	Ours
	0.196	**0.122**	0.316	**0.187**	0.437	**0.177**
	Effect of zero position LR of Gaussians					
OCR-CER ↓ (3K iterations)	w/o	Ours	w/o	Ours	w/o	Ours
	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.

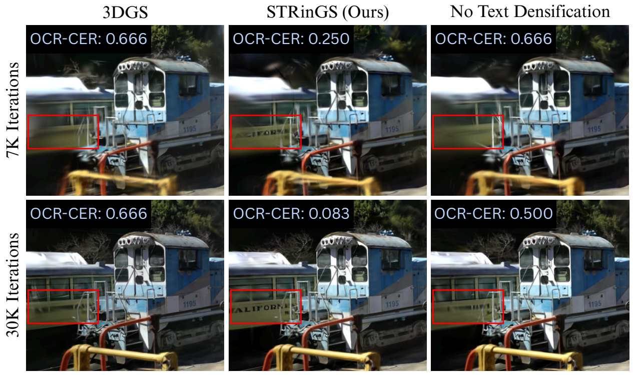

Figure 7. Effect of text densification on a scene from the Tanks&Temples [17] dataset. **Left**: Vanilla 3DGS fails to reconstruct readable text even after 30k iterations, resulting in high OCR-CER of 0.666. **Middle**: STRinGS (Ours) with text densification achieves sharp and semantically meaningful 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).

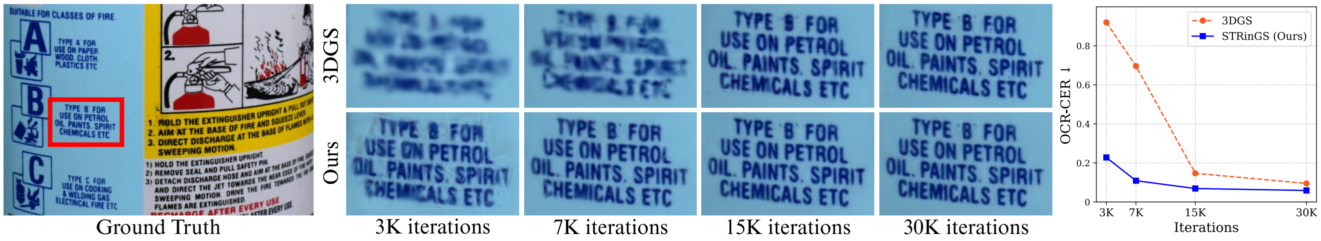

Figure 8. Text reconstruction across training iterations on the *Extinguisher* scene from our STRinGS-360 dataset. STRinGS achieves clearer and more accurate text reconstruction earlier than 3DGS, as reflected in the plot for OCR-CER of the scene over iterations.

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.

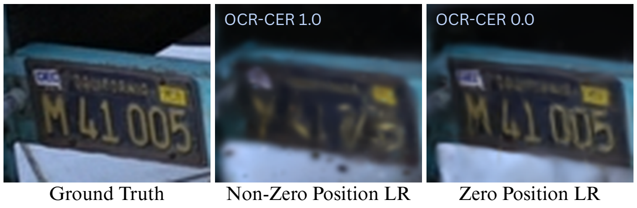

Figure 9. Effect of position learning rate at the end of phase 1 (3K iterations) on a scene from the Tanks&Temples [17] dataset. A non-zero LR causes Gaussians to drift, leading to poor text reconstruction (CER = 1.0). Instead, freezing their positions (zero LR) preserves spatial alignment, enabling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