R^2-Gaussian: Rectifying Radiative Gaussian Splatting for Tomographic Reconstruction

Ruyi Zha[1] **Tao Jun Lin**[1] **Yuanhao Cai**[2,*] **Jiwen Cao**[1]
Yanhao Zhang[3] **Hongdong Li**[1]

[1]The Australian National University [2]Johns Hopkins University
[3]Robotics Institute, University of Technology Sydney
{ruyi.zha, taojun.lin, jiwen.cao, hongdong.li}@anu.edu.au
caiyuanhao1998@gmail.com yanhao.zhang@uts.edu.au

Abstract

3D Gaussian splatting (3DGS) has shown promising results in image rendering and surface reconstruction. However, its potential in volumetric reconstruction tasks, such as X-ray computed tomography, remains under-explored. This paper introduces R^2-Gaussian, the first 3DGS-based framework for sparse-view tomographic reconstruction. By carefully deriving X-ray rasterization functions, we discover a previously unknown *integration bias* in the standard 3DGS formulation, which hampers accurate volume retrieval. To address this issue, we propose a novel rectification technique via refactoring the projection from 3D to 2D Gaussians. Our new method presents three key innovations: (1) introducing tailored Gaussian kernels, (2) extending rasterization to X-ray imaging, and (3) developing a CUDA-based differentiable voxelizer. Experiments on synthetic and real-world datasets demonstrate that our method outperforms state-of-the-art approaches in accuracy and efficiency. Crucially, it delivers high-quality results in 4 minutes, which is $12\times$ faster than NeRF-based methods and on par with traditional algorithms. Code and models are available on the project page `https://github.com/Ruyi-Zha/r2_gaussian`.

1 Introduction

Computed tomography (CT) is an essential imaging technique for noninvasively examining the internal structure of objects. Most CT systems use X-rays as the imaging source thanks to their ability to penetrate solid substances [20]. During a CT scan, an X-ray machine captures multi-angle 2D projections that measure ray attenuation through the material. As the core of CT, tomographic reconstruction aims to recover the 3D density field of the object from its projections. This task is challenging in two aspects. Firstly, the harmful X-ray radiation limits the acquisition of sufficient and noise-free projections, making reconstruction a complex and ill-posed problem. Secondly, time-sensitive applications like medical diagnosis require algorithms to deliver results promptly.

Existing tomography methods suffer from either suboptimal reconstruction quality or slow processing speed. Traditional CT algorithms [13, 2, 55] deliver results in minutes but induce serious artifacts. Supervised learning-based approaches [32, 33, 10, 35] achieve promising outcomes by learning semantic priors but struggle with out-of-distribution objects. Recently, neural radiance fields (NeRF) [43] have been applied to tomography and perform well in per-case reconstruction [67, 66, 48, 6, 54]. However, they are very time-consuming (> 30 minutes) because a huge amount of points have to be sampled for volume rendering.

[*] Yuanhao Cai is the corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

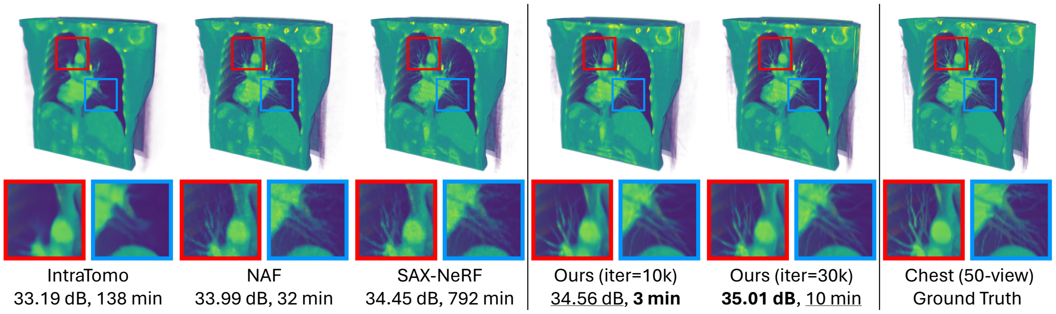

Figure 1: We compare our method to state-of-the-art NeRF-based methods (IntraTomo [66], NAF [67], SAX-NeRF [6]) in terms of visual quality, PSNR (dB), and training time (minute). Our method achieves the highest reconstruction quality and is significantly faster than other methods.

Recently, 3D Gaussian splatting (3DGS) [23] has outperformed NeRF in both quality and efficiency for view synthesis [64, 38, 31] and surface reconstruction [16, 18, 65]. However, attempts to apply the 3DGS technique to volumetric reconstruction tasks, such as X-ray tomography, are limited and ineffective. Some concurrent works [7, 14] empirically modify 3DGS for X-ray view synthesis, but they treat it solely as a data augmentation tool for traditional tomography algorithms. To date, there is no 3DGS-based method for direct CT reconstruction.

In this paper, we reveal an inherent **integration bias** in 3DGS. This bias, despite having a negligible impact on image rendering, critically hampers volumetric reconstruction. To be more specific, we will show in Sec. 4.2.1 that the standard 3DGS overlooks a covariance-related scaling factor when splatting a 3D Gaussian kernel onto the 2D image plane. This formulation leads to inconsistent volumetric properties queried from different views. Besides the integration bias, there are other challenges in applying 3DGS to tomography, such as the difference between natural light and X-ray imaging and the lack of an effective technique to query volumes from kernels.

We propose R^2-Gaussian (Rectified Radiative Gaussians) to extend 3DGS to sparse-view tomographic reconstruction. R^2-Gaussian achieves a bias-free training pipeline with three significant improvements. **Firstly**, we introduce a novel radiative Gaussian kernel, which acts as a local density field parameterized by central density, position, and covariance. We initialize Gaussian parameters using the analytical method FDK [13] and optimize them with photometric losses. **Secondly**, we rectify the 3DGS rasterizer to support X-ray imaging. This is achieved by deriving new X-ray rendering functions and correcting the integration bias for accurate density retrieval. **Thirdly**, we develop a CUDA-based differentiable voxelizer, which not only extracts 3D volumes from Gaussians but also enables voxel-based regularization during training. We evaluate R^2-Gaussian on both synthetic and real-world datasets. Extensive experiments demonstrate that our method surpasses state-of-the-art (SOTA) methods within 4 minutes, which is 12× faster than the most efficient NeRF-based solution, NAF [67] and comparable to traditional algorithms. It converges to optimal results in 15 minutes, improving PSNR by 0.6 dB compared to SOTA methods. A visual comparison is shown in Fig. 1.

Our contributions can be summarized as follows: (1) We discover a previously unknown integration bias in 3DGS that impedes volumetric reconstruction. (2) We propose the first 3DGS-based tomography framework by introducing new kernels, extending rasterization to X-ray imaging, and developing a differentiable voxelizer. (3) Our method significantly outperforms state-of-the-art methods in both reconstruction quality and training speed, highlighting its practical value.

2 Related work

Tomographic reconstruction Computed tomography (CT) is widely used for non-intrusive inspection in medicine [17, 22], biology [12, 39, 24], and industry [11]. Conventional fan-beam CT produces a 3D volume by reconstructing each slice from 1D projection arrays. Recently, the cone-beam scanner has become popular for its fast scanning and high resolution [52], leading to the demand for 3D tomography, i.e., recovering the volume directly from 2D projection images. Our work focuses on 3D sparse-view reconstruction where less than a hundred projections are captured to reduce radiation exposure. Traditional algorithms are mainly grouped into analytical and iterative methods. Analytical methods like filtered back projection (FBP) and its 3D variant FDK [13] produce results instantly (< 1 second) by solving the Radon transform and its inverse [46]. However, they introduce serious streak artifacts in sparse-view scenarios. Iterative methods [2, 55, 40, 51] formulate tomography as a maximum-a-posteriori problem and iteratively minimize the energy function with regularizations. They successfully suppress artifacts but take longer time (< 10 minutes) and lose structure details. Deep learning methods can be categorized as supervised and self-supervised families. Supervised methods learn semantic priors from CT datasets. They then use the trained networks to inpaint projections [3, 15], denoise volumes [10, 28, 35, 37] or directly output results [19, 63, 1, 32, 33]. Supervised learning methods perform well in cases similar to training sets but suffer from poor generation ability when applied to unseen data. To overcome this limitation, some studies [67, 66, 48, 6, 54] handle tomography in a self-supervised learning fashion. Inspired by NeRF [43], they model the density field with coordinate-based networks and optimize them with photometric losses. Although NeRF-based methods excel in per-case reconstruction, they are time-consuming (>30 minutes) due to the extensive point sampling in volume rendering. Our work can be put into the self-supervised learning family, but it greatly accelerates the training process and improves reconstruction quality.

3DGS 3D Gaussian splatting [23] outperforms NeRF in speed by leveraging highly parallelized rasterization for image rendering. 3DGS represents objects with a set of trainable Gaussian-shaped primitives. It has achieved great success in RGB tasks, including surface reconstruction [16, 18, 65], dynamic scene modeling [60, 34, 61], human avatar [36, 30, 27], 3D generation [57, 62, 9], etc. Some concurrent works have extended 3DGS to X-ray imaging. X-Gaussian [7] modify 3DGS to synthesize novel-view X-ray projections. Gao et al. [14] improve X-Gaussian by considering complex noise-inducing physical effects. While they produce plausible 2D X-ray projections, they cannot directly extract 3D density volumes from trained Gaussians. Instead, they first augment projections with 3DGS, and then use traditional algorithms such as FDK for CT reconstruction, which is neither efficient nor effective. Li et al. [29] represent the density field with customized Gaussian kernels, but they replace the efficient rasterization with existing CT simulators. In comparison, our work can both rasterize X-ray projections and voxelize density volumes from Gaussians.

3 Preliminary

3.1 X-ray imaging

A projection $\mathbf{I} \in \mathbb{R}^{H \times W}$ measures ray attenuation through the material as shown in Fig. 2. For an X-ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d} \in \mathbb{R}^3$ with initial intensity I_0 and path bounds t_n and t_f, the corresponding raw pixel value $I'(\mathbf{r})$ is given with the Beer-Lambert Law [20] by: $I'(\mathbf{r}) = I_0 \exp(-\int_{t_n}^{t_f} \sigma(\mathbf{r}(t))\, dt)$. Here, $\sigma(\mathbf{x})$ is the isotropic density (or attenuation coefficient in physics) at position $\mathbf{x} \in \mathbb{R}^3$. Tomography typically transforms raw data to the logarithmic space for computational simplicity, i.e.,

$$I(\mathbf{r}) = \log I_0 - \log I'(\mathbf{r}) = \int_{t_n}^{t_f} \sigma(\mathbf{r}(t))dt, \quad (1)$$

where each pixel value $I(\mathbf{r})$ represents the density integral along the ray path. Except otherwise specified, we use the logarithmic projections as inputs. The goal of tomographic reconstruction is to estimate the 3D distribution of $\sigma(\mathbf{x})$, output as a discrete volume, with X-ray projections $\{\mathbf{I}_i\}_{i=1,\cdots,N}$ captured from N different angles. Note that real-world projections contain minor anisotropic physical effects such as Compton scattering. Following previous works [13, 2, 55, 67], we do not explicitly model them but treat them as noise during the reconstruction.

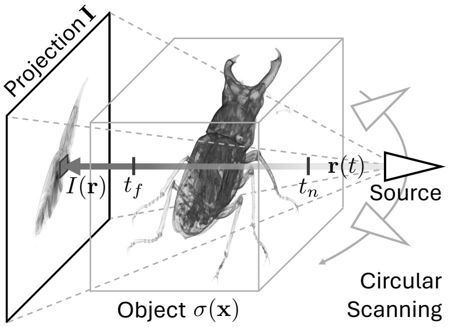

Figure 2: A detection plane captures the attenuation of X-rays emitted from different angles.

3.2 3D Gaussian splatting

3D Gaussian splatting [23] models the scene with a set of 3D Gaussian kernels $\mathbb{G}^3 = \{G_i^3\}_{i=1,\cdots,M}$, each parameterized by position, covariance, color, and opacity. A rasterizer $\mathcal{R}$ renders an RGB image

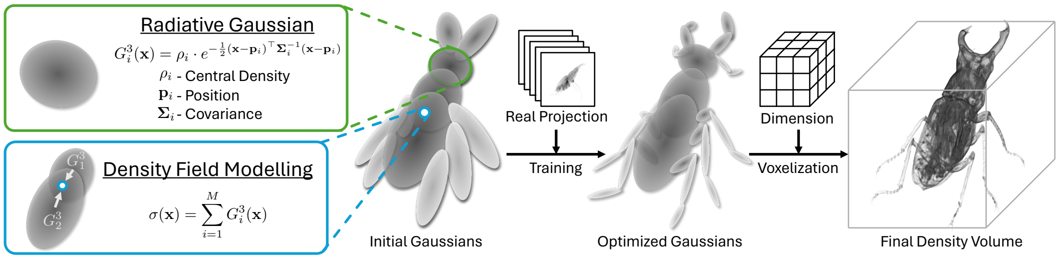
Radiative Gaussian
$G_i^3(\mathbf{x}) = \rho_i \cdot e^{-\frac{1}{2}(\mathbf{x}-\mathbf{p}_i)^\top \mathbf{\Sigma}_i^{-1}(\mathbf{x}-\mathbf{p}_i)}$
ρ_i - Central Density
$\mathbf{p}_i$ - Position
$\mathbf{\Sigma}_i$ - Covariance
Density Field Modelling
G_1^3
G_2^3
$\sigma(\mathbf{x}) = \sum_{i=1}^{M} G_i^3(\mathbf{x})$
Initial Gaussians
Real Projection
Training
Optimized Gaussians
Dimension
Voxelization
Final Density Volume

Figure 3: We represent the scanned object as a set of radiative Gaussians. We optimize them using real X-ray projections and finally retrieve the density volume with voxelization.

$\mathbf{I}_{rgb} \in \mathbb{R}^{H\times W\times 3}$ from these Gaussians, formulated as

$$\mathbf{I}_{rgb} = \mathcal{R}(\mathbb{G}^3) = \mathcal{C} \circ \mathcal{P} \circ \mathcal{T}(\mathbb{G}^3), \tag{2}$$

where $\mathcal{T}$, $\mathcal{P}$, and $\mathcal{C}$ are the transformation, projection, and composition modules, respectively. First, $\mathcal{T}$ transforms the 3D Gaussians into the ray space, aligning viewing rays with the coordinate axis to enhance computational efficiency. The transformed 3D Gaussians are then projected onto the image plane: $\mathbb{G}^2 = \mathcal{P}(\mathbb{G}^3)$. The projected 2D Gaussian retains the same opacity and color as its 3D counterpart but omits the third row and column of position and covariance. An RGB image is then rendered by compositing these 2D Gaussians using alpha-blending [45]: $\mathbf{I}_{rgb} = \mathcal{C}(\mathbb{G}^2)$. The rasterizer is differentiable, allowing for the optimization of kernel parameters using photometric losses. 3DGS initializes sparse Gaussians with structure-from-motion (SfM) points [53]. During training, an adaptive control strategy dynamically densifies Gaussians to improve scene representation.

4 Method

In this section, we first introduce radiative Gaussian as a novel object representation in Sec. 4.1. Next, we adapt 3DGS to tomography in Sec. 4.2. Specifically, we derive new rasterization functions and analyze the integration bias of standard 3DGS in Sec. 4.2.1. We further develop a differentiable voxelizer for volume retrieval in Sec. 4.2.2. The optimization strategy is elaborated in Sec. 4.2.3.

4.1 Representing objects with radiative Gaussians

As shown in Fig. 3, we represent the target object with a group of learnable 3D kernels $\mathbb{G}^3 = \{G_i^3\}_{i=1,\cdots,M}$ that we term as radiative Gaussians. Each kernel G_i^3 defines a local Gaussian-shaped density field, i.e.,

$$G_i^3(\mathbf{x}|\rho_i, \mathbf{p}_i, \mathbf{\Sigma}_i) = \rho_i \cdot \exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{p}_i)^\top \mathbf{\Sigma}_i^{-1}(\mathbf{x}-\mathbf{p}_i)\right), \tag{3}$$

where ρ_i, $\mathbf{p}_i \in \mathbb{R}^3$ and $\mathbf{\Sigma}_i \in \mathbb{R}^{3\times 3}$ are learnable parameters representing central density, position and covariance, respectively. For optimization purposes, we follow [23] to further decompose the covariance matrix $\mathbf{\Sigma}_i$ into the rotation matrix $\mathbf{R}_i$ and scale matrix $\mathbf{S}_i$: $\mathbf{\Sigma}_i = \mathbf{R}_i\mathbf{S}_i\mathbf{S}_i^\top\mathbf{R}_i^\top$. The overall density at position $\mathbf{x} \in \mathbb{R}^3$ is then computed by summing the density contribution of kernels:

$$\sigma(\mathbf{x}) = \sum_{i=1}^{M} G_i^3(\mathbf{x}|\rho_i, \mathbf{p}_i, \mathbf{\Sigma}_i). \tag{4}$$

Compared with standard 3DGS, our kernel formulation removes view-dependent color because X-ray attenuation depends only on isotropic density, as shown in Eq. (1). More importantly, we define the density query function (Eq. (4)) for radiative Gaussians, making them useful for both 2D image rendering and 3D volume reconstruction. In contrast, the opacity in 3DGS is empirically designed for RGB rendering, leading to challenges when extracting 3D models such as meshes from Gaussians [16, 8, 65]. Concurrent work [29] also explores kernel-based representation but uses simplified isotropic Gaussians. Our work employs a general Gaussian distribution, offering more flexibility and precision in modeling complex structures.

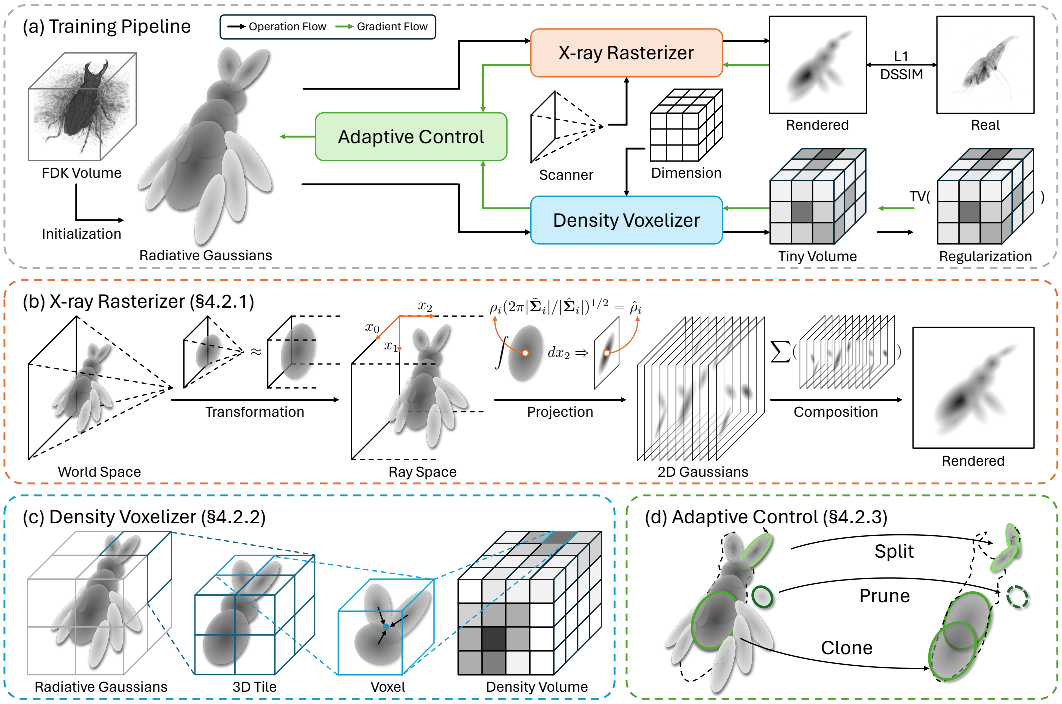

Figure 4: Training pipeline of R^2-Gaussian. (a) Overall training pipeline. (b) X-ray rasterization for projection rendering. (c) Density voxelization for volume retrieval. (d) Modified adaptive control.

Initialization 3DGS initializes Gaussians with SfM points, which is not applicable to volumetric tomography. Instead, we initialize our radiative Gaussians using preliminary results obtained from the analytical method. Specifically, we use FDK [13] to reconstruct a low-quality volume in less than 1 second. We then exclude empty spaces with a density threshold τ and randomly sample M points as kernel positions. Following [23], we set the scales of Gaussians as the nearest neighbor distances and assume no rotation. The central densities are queried from the FDK volume. We empirically scale down the queried densities with k to compensate for the overlay between kernels.

4.2 Training radiative Gaussians

Our training pipeline is shown in Fig. 4. Radiative Gaussians are first initialized from an FDK volume. We then rasterize projections for photometric losses and voxelize tiny density volumes for 3D regularization. Adaptive control is used to densify Gaussians for better representation. After training, we voxelize density volumes of the target size for evaluation.

4.2.1 X-ray rasterization

This section focuses on the theoretical derivation of X-ray rasterization $\mathcal{R}$. As discussed in Sec. 3.1, the pixel value of a projection is the integral of density along the corresponding ray path. We substitute Eq. (4) into Eq. (1), yielding

$$I_r(\mathbf{r}) = \int \sum_{i=1}^{M} G_i^3(\mathbf{r}(t)|\rho_i, \mathbf{p}_i, \mathbf{\Sigma}_i)dt = \sum_{i=1}^{M} \int G_i^3(\mathbf{r}(t)|\rho_i, \mathbf{p}_i, \mathbf{\Sigma}_i)dt, \tag{5}$$

where $I_r(\mathbf{r})$ is the rendered pixel value. This implies that we can individually integrate each 3D Gaussian to rasterize an X-ray projection. Note that t_n and t_f in Eq. (1) are neglected because we assume all Gaussians are bounded inside the target space.

Transformation Since a cone-beam X-ray scanner can be modeled similarly to a pinhole camera, we follow [69] to transfer Gaussians from the world space to the ray space. In ray space, the viewing rays are parallel to the third coordinate axis, facilitating analytical integration. Due to the non-Cartesian nature of ray space, we employ the local affine transformation to Eq. (5), yielding

$$I_r(\mathbf{r}) \approx \sum_{i=1}^{M} \int G_i^3(\tilde{\mathbf{x}}|\rho_i, \underbrace{\phi(\mathbf{p})}_{\tilde{\mathbf{p}}_i}, \underbrace{\mathbf{J}_i\mathbf{W}\mathbf{\Sigma}_i\mathbf{W}^\top\mathbf{J}_i^\top}_{\tilde{\mathbf{\Sigma}}_i}) dx_2, \tag{6}$$

where $\tilde{\mathbf{x}} = [x_0, x_1, x_2]^\top$ is a point in ray space, $\tilde{\mathbf{p}}_i \in \mathbb{R}^3$ is the new Gaussian position obtained through projective mapping ϕ, and $\tilde{\mathbf{\Sigma}}_i \in \mathbb{R}^{3\times3}$ is the new Gaussian covariance controlled by local approximation matrix $\mathbf{J}_i$ and viewing transformation matrix $\mathbf{W}$. Refer to Appendix A for determining ϕ, $\mathbf{J}_i$, and $\mathbf{W}$ from scanner parameters.

Projection and composition A good property of *normalized* 3D Gaussian distribution is that its integral along one coordinate axis yields a normalized 2D Gaussian distribution. Substitute Eq. (3) into Eq. (6) and we have

$$\begin{aligned}
I_r(\mathbf{r}) &\approx \sum_{i=1}^{M} \rho_i (2\pi)^{\frac{3}{2}} |\tilde{\mathbf{\Sigma}}_i|^{\frac{1}{2}} \int \underbrace{\frac{1}{(2\pi)^{\frac{3}{2}}|\tilde{\mathbf{\Sigma}}_i|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\tilde{\mathbf{x}}-\tilde{\mathbf{p}}_i)^\top \tilde{\mathbf{\Sigma}}_i^{-1}(\tilde{\mathbf{x}}-\tilde{\mathbf{p}}_i)\right)}_{\text{Normalized 3D Gaussian distribution}} dx_2 \\
&= \sum_{i=1}^{M} \rho_i (2\pi)^{\frac{3}{2}} |\tilde{\mathbf{\Sigma}}_i|^{\frac{1}{2}} \underbrace{\frac{1}{2\pi|\hat{\mathbf{\Sigma}}_i|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\hat{\mathbf{x}}-\hat{\mathbf{p}}_i)^\top \hat{\mathbf{\Sigma}}_i^{-1}(\hat{\mathbf{x}}-\hat{\mathbf{p}}_i)\right)}_{\text{Normalized 2D Gaussian distribution}} \\
&= \sum_{i=1}^{M} G_i^2(\hat{\mathbf{x}}| \underbrace{\sqrt{\frac{2\pi|\tilde{\mathbf{\Sigma}}_i|}{|\hat{\mathbf{\Sigma}}_i|}}\rho_i}_{\hat{\rho}_i}, \hat{\mathbf{p}}_i, \hat{\mathbf{\Sigma}}_i),
\end{aligned} \tag{7}$$

where $\hat{\mathbf{x}} \in \mathbb{R}^2$, $\hat{\mathbf{p}} \in \mathbb{R}^2$, $\hat{\mathbf{\Sigma}} \in \mathbb{R}^{2\times2}$ are obtained by dropping the third rows and columns of their counterparts $\tilde{\mathbf{x}}$, $\tilde{\mathbf{p}}$, and $\tilde{\mathbf{\Sigma}}$, respectively. Eq. (7) shows that an X-ray projection can be rendered by simply summing 2D Gaussians instead of alpha-compositing them in natural light imaging.

Integration bias During the projection, a key difference between our 2D Gaussian and the original one in 3DGS is the central density (opacity) $\hat{\rho}_i$. As shown in Eq. (7), we scale the density with a covariance-related factor $\mu_i = (2\pi|\tilde{\mathbf{\Sigma}}_i|/|\hat{\mathbf{\Sigma}}_i|)^{1/2}$: $\hat{\rho}_i = \mu_i\rho_i$, while 3DGS does not. This implies that 3DGS, in fact, learns an integrated density in the 2D image plane rather than the actual one in 3D space. This integration bias, though having a negligible impact on imaging rendering, leads to significant inconsistency in density retrieval. We demonstrate the inconsistency with a simplified 2D-to-1D projection in Fig. 5. When attempting to recover the central density ρ in 3D space with $\rho_i = \hat{\rho}_i/\mu_j$, we find different views (μ_j) lead to different results. This violates the isotropic nature of ρ_i, preventing us from determining the correct value. In contrast, our method assigns the actual 3D density to the kernel and forwardly computes the 2D projection, thus fundamentally solving the issue. While conceptually simple, implementing our idea requires substantial engineering efforts, including reprogramming all backpropagation routines in CUDA.

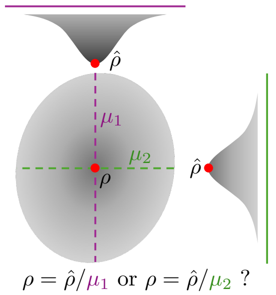

Figure 5: Density inconsistency in 3DGS.

4.2.2 Density voxelization

We develop a voxelizer $\mathcal{V}$ to efficiently query a density volume $\mathbf{V} \in \mathbb{R}^{X\times Y\times Z}$ from radiative Gaussians: $\mathbf{V} = \mathcal{V}(\mathbb{G}^3)$. Inspired by voxelizers used in RGB tasks [57], our voxelizer first partitions the target space into multiple $8\times8\times8$ 3D tiles. It then culls Gaussians, retaining those with a 99% confidence of intersecting the tile. In each 3D tile, voxel values are parallelly computed by summing the contributions of nearby kernels with Eq. (4). We implement the voxelizer and its backpropagation in CUDA, making it differentiable for optimization. This design not only accelerates the query process (> 100 FPS) but also allows us to regularize radiative Gaussians with 3D priors.

Table 1: Quantitative results on sparse-view tomography. We colorize the best, second-best, and third-best numbers.

Methods	75-view			50-view			25-view		
	PSNR↑	SSIM↑	Time↓	PSNR↑	SSIM↑	Time↓	PSNR↑	SSIM↑	Time↓
Synthetic dataset									
FDK [13]	28.63	0.497	-	26.50	0.422	-	22.99	0.317	-
SART [2]	36.06	0.897	4m41s	34.37	0.875	3m36s	31.14	0.825	1m47s
ASD-POCS [55]	36.64	0.940	2m25s	34.34	0.914	1m52s	30.48	0.847	56s
IntraTomo [66]	35.42	0.924	2h7m	35.25	0.923	2h9m	34.68	0.914	2h19m
NAF [67]	37.84	0.945	30m43s	36.65	0.932	32m4s	33.91	0.893	31m1s
SAX-NeRF [6]	38.07	0.950	13h5m	36.86	0.938	13h5m	34.33	0.905	13h3m
Ours (iter=10k)	38.29	0.954	2m38s	37.63	0.949	2m35s	35.08	0.922	2m35s
Ours (iter=30k)	38.88	0.959	8m21s	37.98	0.952	8m14s	35.19	0.923	8m28s
Real-world dataset									
FDK [13]	30.03	0.535	-	27.38	0.449	-	23.30	0.335	-
SART [2]	34.42	0.845	5m11s	33.61	0.827	3m28s	31.52	0.790	1m47s
ASD-POCS [55]	36.33	0.868	2m43s	34.58	0.861	1m49s	31.32	0.810	56s
IntraTomo [66]	36.79	0.858	2h25m	36.99	0.854	2h19m	35.85	0.835	2h18m
NAF [67]	38.58	0.848	51m28s	36.44	0.818	51m31s	32.92	0.772	51m24s
SAX-NeRF [6]	34.93	0.854	13h21m	34.89	0.840	13h23m	33.49	0.793	13h25m
Ours (iter=10k)	38.10	0.872	3m39s	37.52	0.866	3m37s	35.10	0.840	3m23s
Ours (iter=30k)	39.40	0.875	14m16s	38.24	0.864	13m52s	34.83	0.833	12m56s

4.2.3 Optimization

We optimize radiative Gaussians using stochastic gradient descent. Besides photometric L1 loss $\mathcal{L}_1$ and D-SSIM loss $\mathcal{L}_{ssim}$ [59], we further incorporate a 3D total variation (TV) regularization [49] $\mathcal{L}_{tv}$ as a homogeneity prior for tomography. At each training iteration, we randomly query a tiny density volume $\mathbf{V}_{tv} \in \mathbb{R}^{D\times D\times D}$ (same spacing as the target output) and minimize its total variation. The overall training loss is defined as:

$$\mathcal{L}_{total} = \mathcal{L}_1(\mathbf{I}_r, \mathbf{I}_m) + \lambda_{ssim}\mathcal{L}_{ssim}(\mathbf{I}_r, \mathbf{I}_m) + \lambda_{tv}\mathcal{L}_{tv}(\mathbf{V}_{tv}), \tag{8}$$

where $\mathbf{I}_r$, $\mathbf{I}_m$, λ_{ssim} and λ_{tv} are rendered projection, measured projection, D-SSIM weight, and TV weight, respectively. Adaptive control is employed during training to enhance object representation. We remove empty Gaussians and densify (clone or split) those with large loss gradients. Considering objects such as human organs have extensive homogeneous areas, we do not prune large Gaussians. As for densification, we halve the densities of both original and replicated Gaussians. This strategy mitigates the sudden performance drop caused by new Gaussians and hence stabilizes training.

5 Experiments

5.1 Experimental settings

Dataset We conduct experiments on both synthetic and real-world datasets. For the synthetic dataset, we collect 15 real CT volumes, ranging from organisms to artificial objects. We then use the tomography toolbox TIGRE [5] to synthesize X-ray projections and add Compton scatter and electric noise. For real-world experiments, we use three cases from the FIPS dataset [56], each with 721 real projections. Since ground truth volumes are unavailable, we use FDK [13] to create pseudo-ground truth using all views and then subsample views for sparse-view experiments. We set 75, 50, and 25 views for both synthetic and real-world data as three sparse-view scenarios. Refer to Appendix B for more details of datasets.

Implementation details Our R^2-Gaussian is implemented in PyTorch [44] and CUDA [50], and trained with the Adam optimizer [25] for 30k iterations. Learning rates for position, density, scale, and rotation are initially set as 0.0002, 0.01, 0.005, and 0.001, respectively, and exponentially to 0.1 of their initial values. Loss weights are $\lambda_{ssim} = 0.25$ and $\lambda_{tv} = 0.05$. We initialize $M = 50$k Gaussians with a density threshold $\tau = 0.05$ and scaling term $k = 0.15$. The TV volume size is

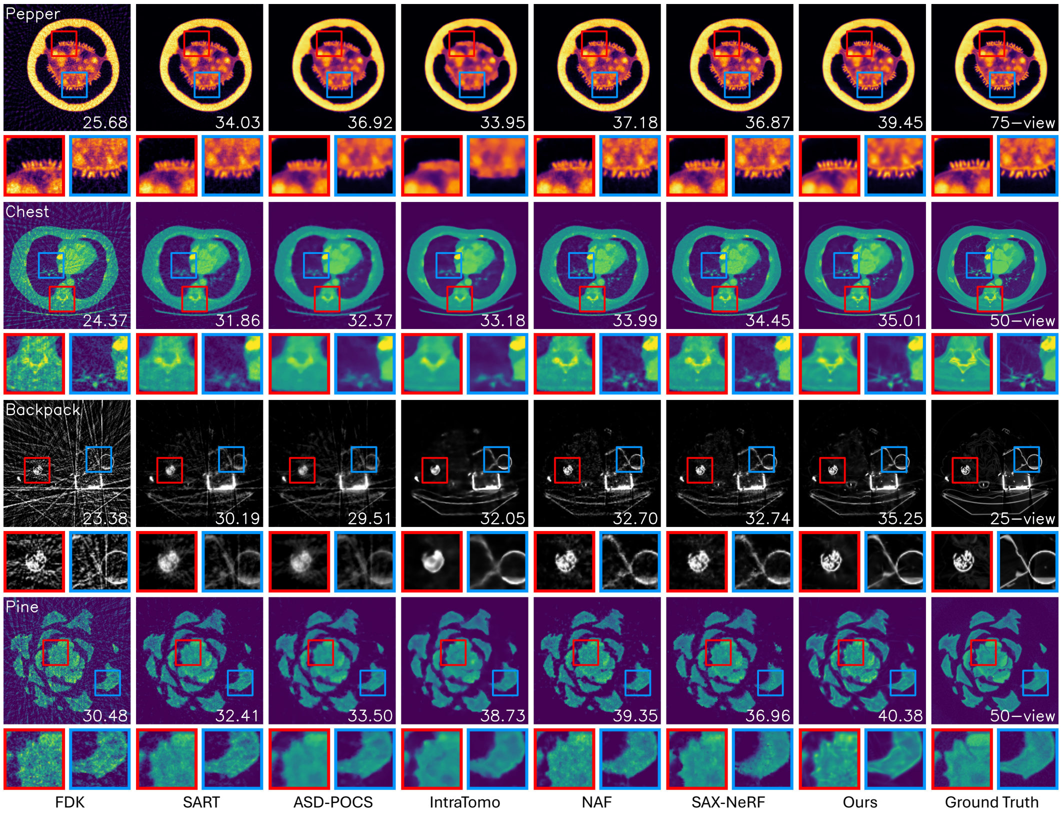

Figure 6: Colorized slice examples of different methods with PSNR (dB) shown at the bottom right of each image. The first three rows are from the synthetic dataset and the last row is from the real-world dataset. Our method recovers more details and suppresses artifacts.

$D = 32$. Adaptive control runs from 500 to 15k iterations with a gradient threshold of 0.00005. All methods run on a single RTX3090 GPU. We evaluate reconstruction quality using PSNR and SSIM [59], with PSNR calculated in 3D volume and SSIM averaged over 2D slices in axial, coronal, and sagittal directions. We also report the running time as a reflection of efficiency.

5.2 Results and evaluation

For fairness, we do not compare methods that require external training data but focus on those that solely use 2D projections of arbitrary objects. We compare R^2-Gaussian with three traditional methods (FDK [13], SART [2], ASD-POCS [55]) and three SOTA NeRF-based methods (IntraTomo [66], NAF [67], SAX-NeRF [6]). Tab. 1 reports the quantitative results on sparse-view tomography. Note that we do not report the running time for FDK as it is instant. R^2-Gaussian achieves the best performance across all synthetic and most real-world experiments. Specifically, our method delivers a 0.93 dB higher PSNR than SAX-NeRF, on the synthetic dataset, and a 0.95 dB improvement over IntraTomo on the real-world dataset. It is also worth noting that our 50-view results are already on par with the 75-view results of other methods. Regarding efficiency, our method converges to optimal results in 15 minutes, which is 3.7× faster than the most efficient NeRF-based method, NAF. Surprisingly, it takes less than 4 minutes to surpass other methods, which is even faster than the traditional algorithm SART. Fig. 6 shows the visual comparisons of different methods. FDK and SART introduce streak artifacts, while ASD-POCS and IntraTomo blur structural details. NAF and SAX-NeRF are better than other baseline methods but have salt-and-pepper noise. In comparison, our method successfully recovers sharp details, e.g., ovules of pepper, and maintains good smoothness for homogeneous areas, e.g., muscles in the chest.

Table 2: Quantitative results of X-3DGS and our method on the synthetic dataset.

	75-view		50-view		25-view	
	X-3DGS	Ours	X-3DGS	Ours	X-3DGS	Ours
2D PSNR↑	49.97	50.54	47.26	49.70	39.84	46.28
2D SSIM↑	0.987	0.986	0.984	0.986	0.967	0.982
3D PSNR↑	23.40	38.86	21.24	37.98	14.07	35.17
3D SSIM↑	0.660	0.959	0.562	0.952	0.408	0.923

5.3 Ablation study

Integration bias To demonstrate the impact of integration bias discussed in Sec. 4.2.1, we develop an X-ray version of 3DGS (X-3DGS) that uses X-ray rendering while retaining the biased 3D-to-2D Gaussian projection. We use the same voxelizer in Sec. 4.2.2 to extract volumes. Before voxelization, we divide the learned density of each Gaussian by the mean scaling factor μ of all training views. Tab. 2 shows that rectifying integration bias benefits both 2D rendering (+3.15 dB PSNR) and 3D reconstruction (+17.77 dB PSNR). Fig. 7 visualize rendering and reconstruction results. While X-3DGS renders reasonable 2D projections, its reconstruction quality is significantly worse than ours. Besides, there are notable discrepancies in slices queried from different views. The conflicting 2D and 3D performances indicate that X-3DGS, despite fitting images well, does not accurately model the density field. In contrast, our method learns the actual view-independent density, eliminating inconsistencies and ensuring unbiased object representation.

Component analysis We conduct ablation experiments to assess the effect of FDK initialization (Init.), modified adaptive control (AC), and total variation regularization (Reg.) on performance. The baseline model excludes these components and uses randomly generated Gaussians for initialization. Experiments are performed under the 50-view condition, evaluating PSNR, SSIM, training time, and Gaussian count (Gau.). Results are listed in Tab. 3. FDK initialization boosts PSNR by 0.9 dB. Adaptive control improves quality but prolongs training due to more Gaussians. TV regularization increases SSIM by reducing artifacts and promoting smoothness. Overall, our full model outperforms the baseline, improving PSNR by 1.51 dB and SSIM by 0.018, with training time under 9 minutes.

Parameter analysis We perform parameter analysis on the number of initialized Gaussians M, TV loss weight λ_{tv}, and TV volume size D. The results are shown in the last three blocks of Tab. 3. R^2-Gaussian achieves good quality-efficiency balance at 50k initialized Gaussians. A TV loss weight of $\lambda_{tv} = 0.05$ improves reconstruction, but larger values can lead to degradation. The training time increases with TV volume size while the performance peaks at $D = 32$.

Convergence analysis Fig. 8 compares the results of NeRF-based methods and our R^2-Gaussian method at different iterations. Our method, both with and without FDK initialization, converges significantly faster, displaying sharp details by the 500th iteration when other methods still exhibit artifacts and blurriness. Notably, the FDK initialization offers a rough structure before training, which

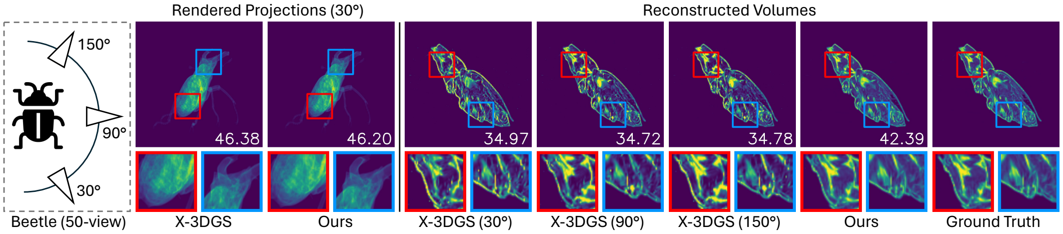

Figure 7: Results of X-3DGS and our method with PSNR (dB) indicated on each image. We show slices of X-3DGS queried from three viewing angles. Although X-3DGS can produce plausible X-ray projections, its reconstructed volume lacks view consistency and exhibits poor quality.

Table 3: Ablation results with our choices in bold.

Synthetic	PSNR↑	SSIM↑	Time↓	Gau.
Baseline (B)	36.47	0.934	4m57s	50k
B+Init.	37.37	0.944	5m29s	50k
B+AC	37.33	0.942	7m33s	70k
B+Reg.	36.79	0.943	6m30s	50k
Full model	37.98	0.952	8m37s	68k
M=5k	37.44	0.946	9m18s	32k
M=10k	37.56	0.948	8m59s	35k
$\boldsymbol{M}$**=50k**	37.98	0.952	8m14s	68k
M=100k	38.03	0.953	9m4s	112k
M=200k	37.82	0.949	9m54s	206k
λ_{tv}=0	37.66	0.948	7m9s	68k
λ_{tv}=0.01	37.88	0.950	8m21s	68k
$\boldsymbol{\lambda_{tv}}$**=0.05**	37.98	0.952	8m14s	68k
λ_{tv}=0.1	37.73	0.951	8m11s	68k
λ_{tv}=0.15	37.40	0.949	8m27s	69k
D=8	37.74	0.949	7m56s	68k
D=16	37.94	0.950	8m18s	68k
$\boldsymbol{D}$**=32**	37.98	0.952	8m14s	68k
D=48	37.90	0.951	9m34s	67k
D=64	37.82	0.949	11m35s	67k

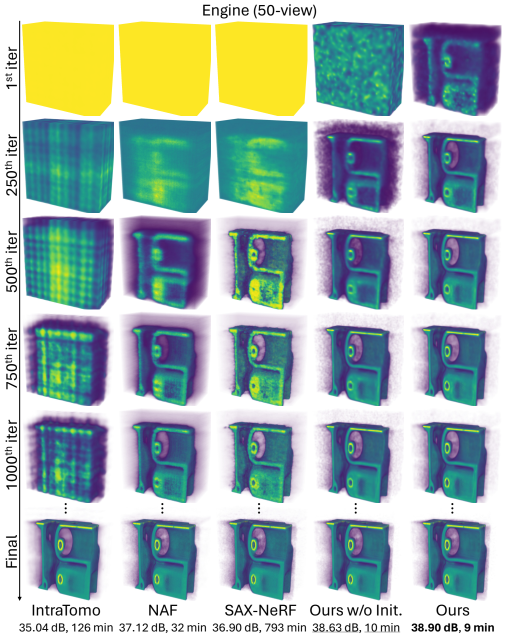

Figure 8: Results of NeRF-based methods and our R^2-Gaussian at different iterations.

further accelerates convergence and enhances reconstruction quality. Finally, our method outperforms others in both performance and efficiency, achieving the highest PSNR of 38.90 dB in 9 minutes.

6 Discussion and conclusion

Discussion R^2-Gaussian inherits some limitations from 3DGS, such as varying training time across modalities, needle-like artifacts under extremely sparse-view conditions, and suboptimal extrapolation for other tomography tasks. Besides, we have not considered calibration errors regarding the scanned geometry and anisotropic physical effects such as Compton scattering. More details are discussed in Appendix E. Despite these limitations, our method's superior performance and fast speed make it valuable for real-world applications for medical diagnosis and industrial inspection.

Conclusion This paper presents R^2-Gaussian, a novel 3DGS-based framework for sparse-view tomographic reconstruction. We identify and rectify a previously overlooked integration bias of standard 3DGS, which hinders accurate density retrieval. Furthermore, we enhance 3DGS for tomography by introducing new kernels, devising X-ray rasterization functions, and developing a differentiable voxelizer. Our R^2-Gaussian surpasses state-of-the-art methods in both reconstruction quality and training speed, demonstrating its potential for real-world applications. Crucially, we speculate that the newly found integration bias may be pervasive across all 3DGS-related research. Consequently, our rectification technique could benefit more tasks beyond computed tomography.

Acknowledgments

The research is funded in part by ARC Discovery Grant (grant ID: DP220100800) of the Australia Research Council.

References

[1] Jonas Adler and Ozan Öktem. Learned primal-dual reconstruction. *IEEE transactions on medical imaging*, 37(6):1322–1332, 2018.

[2] Anders H Andersen and Avinash C Kak. Simultaneous algebraic reconstruction technique (sart): a superior implementation of the art algorithm. *Ultrasonic imaging*, 6(1):81–94, 1984.

[3] Rushil Anirudh, Hyojin Kim, Jayaraman J Thiagarajan, K Aditya Mohan, Kyle Champley, and Timo Bremer. Lose the views: Limited angle ct reconstruction via implicit sinogram completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6343–6352, 2018.

[4] Samuel G Armato III, Geoffrey McLennan, Luc Bidaut, Michael F McNitt-Gray, Charles R Meyer, Anthony P Reeves, Binsheng Zhao, Denise R Aberle, Claudia I Henschke, Eric A Hoffman, et al. The lung image database consortium (lidc) and image database resource initiative (idri): a completed reference database of lung nodules on ct scans. *Medical physics*, 38(2):915–931, 2011.

[5] Ander Biguri, Manjit Dosanjh, Steven Hancock, and Manuchehr Soleimani. Tigre: a matlab-gpu toolbox for cbct image reconstruction. *Biomedical Physics & Engineering Express*, 2(5):055010, 2016.

[6] Yuanhao Cai, Jiahao Wang, Alan Yuille, Zongwei Zhou, and Angtian Wang. Structure-aware sparse-view x-ray 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[7] Yuanhao Cai, Yixun Liang, Jiahao Wang, Angtian Wang, Yulun Zhang, Xiaokang Yang, Zongwei Zhou, and Alan Yuille. Radiative gaussian splatting for efficient x-ray novel view synthesis. In *European Conference on Computer Vision*, pages 283–299. Springer, 2025.

[8] Hanlin Chen, Chen Li, and Gim Hee Lee. Neusg: Neural implicit surface reconstruction with 3d gaussian splatting guidance. *arXiv preprint arXiv:2312.00846*, 2023.

[9] Zilong Chen, Feng Wang, and Huaping Liu. Text-to-3d using gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[10] Hyungjin Chung, Dohoon Ryu, Michael T McCann, Marc L Klasky, and Jong Chul Ye. Solving 3d inverse problems using pre-trained 2d diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22542–22551, 2023.

[11] Leonardo De Chiffre, Simone Carmignato, J-P Kruth, Robert Schmitt, and Albert Weckenmann. Industrial applications of computed tomography. *CIRP annals*, 63(2):655–677, 2014.

[12] Philip CJ Donoghue, Stefan Bengtson, Xi-ping Dong, Neil J Gostling, Therese Huldtgren, John A Cunningham, Chongyu Yin, Zhao Yue, Fan Peng, and Marco Stampanoni. Synchrotron x-ray tomographic microscopy of fossil embryos. *Nature*, 442(7103):680–683, 2006.

[13] Lee A Feldkamp, Lloyd C Davis, and James W Kress. Practical cone-beam algorithm. *Josa a*, 1 (6):612–619, 1984.

[14] Zhongpai Gao, Benjamin Planche, Meng Zheng, Xiao Chen, Terrence Chen, and Ziyan Wu. Ddgs-ct: Direction-disentangled gaussian splatting for realistic volume rendering. *arXiv preprint arXiv:2406.02518*, 2024.

[15] Muhammad Usman Ghani and W Clem Karl. Deep learning-based sinogram completion for low-dose ct. In *2018 IEEE 13th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)*, pages 1–5. IEEE, 2018.

[16] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[17] Godfrey N Hounsfield. Computed medical imaging. *Science*, 210(4465):22–28, 1980.

[18] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024. doi: 10.1145/3641519.3657428.

[19] Kyong Hwan Jin, Michael T McCann, Emmanuel Froustey, and Michael Unser. Deep convolutional neural network for inverse problems in imaging. *IEEE transactions on image processing*, 26(9):4509–4522, 2017.

[20] Avinash C Kak and Malcolm Slaney. *Principles of computerized tomographic imaging*. SIAM, 2001.

[21] Emma Kamutta, Sofia Mäkinen, and Alexander Meaney. Cone-Beam Computed Tomography Dataset of a Seashell, August 2022. URL `https://doi.org/10.5281/zenodo.6983008`.

[22] Shigehiko Katsuragawa and Kunio Doi. Computer-aided diagnosis in chest radiography. *Computerized Medical Imaging and Graphics*, 31(4-5):212–223, 2007.

[23] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023.

[24] Timo Kiljunen, Touko Kaasalainen, Anni Suomalainen, and Mika Kortesniemi. Dental cone beam ct: A review. *Physica Medica*, 31(8):844–860, 2015.

[25] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA, 2015.

[26] Pavol Klacansky. Open scivis datasets, December 2017. URL `https://klacansky.com/open-scivis-datasets/`. `https://klacansky.com/open-scivis-datasets/`.

[27] Muhammed Kocabas, Jen-Hao Rick Chang, James Gabriel, Oncel Tuzel, and Anurag Ranjan. HUGS: Human gaussian splatting. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. URL `https://arxiv.org/abs/2311.17910`.

[28] Suhyeon Lee, Hyungjin Chung, Minyoung Park, Jonghyuk Park, Wi-Sun Ryu, and Jong Chul Ye. Improving 3d imaging with pre-trained perpendicular 2d diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10710–10720, 2023.

[29] Yingtai Li, Xueming Fu, Shang Zhao, Ruiyang Jin, and S Kevin Zhou. Sparse-view ct reconstruction with 3d gaussian volumetric representation. *arXiv preprint arXiv:2312.15676*, 2023.

[30] Zhe Li, Zerong Zheng, Lizhen Wang, and Yebin Liu. Animatable gaussians: Learning pose-dependent gaussian maps for high-fidelity human avatar modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[31] Zhihao Liang, Qi Zhang, Ying Feng, Ying Shan, and Kui Jia. Gs-ir: 3d gaussian splatting for inverse rendering. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[32] Yiqun Lin, Zhongjin Luo, Wei Zhao, and Xiaomeng Li. Learning deep intensity field for extremely sparse-view cbct reconstruction. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 13–23. Springer, 2023.

[33] Yiqun Lin, Jiewen Yang, Hualiang Wang, Xinpeng Ding, Wei Zhao, and Xiaomeng Li. Cˆ 2rv: Cross-regional and cross-view learning for sparse-view cbct reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11205–11214, 2024.

[34] Youtian Lin, Zuozhuo Dai, Siyu Zhu, and Yao Yao. Gaussian-flow: 4d reconstruction with dynamic 3d gaussian particle. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[35] Jiaming Liu, Rushil Anirudh, Jayaraman J Thiagarajan, Stewart He, K Aditya Mohan, Ulugbek S Kamilov, and Hyojin Kim. Dolce: A model-based probabilistic diffusion framework for limited-angle ct reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10498–10508, 2023.

[36] Xian Liu, Xiaohang Zhan, Jiaxiang Tang, Ying Shan, Gang Zeng, Dahua Lin, Xihui Liu, and Ziwei Liu. Humangaussian: Text-driven 3d human generation with gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[37] Zhengchun Liu, Tekin Bicer, Rajkumar Kettimuthu, Doga Gursoy, Francesco De Carlo, and Ian Foster. Tomogan: low-dose synchrotron x-ray tomography with generative adversarial networks: discussion. *JOSA A*, 37(3):422–434, 2020.

[38] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[39] Vladan Lučić, Friedrich Förster, and Wolfgang Baumeister. Structural studies by electron tomography: from cells to molecules. *Annu. Rev. Biochem.*, 74:833–865, 2005.

[40] Stephen H Manglos, George M Gagne, Andrzej Krol, F Deaver Thomas, and Rammohan Narayanaswamy. Transmission maximum-likelihood reconstruction with ordered subsets for cone beam ct. *Physics in Medicine & Biology*, 40(7):1225, 1995.

[41] Alexander Meaney. Cone-Beam Computed Tomography Dataset of a Pine Cone, August 2022. URL `https://doi.org/10.5281/zenodo.6985407`.

[42] Alexander Meaney. Cone-beam computed tomography dataset of a walnut, August 2022. URL `https://doi.org/10.5281/zenodo.6986012`.

[43] 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. In *ECCV*, 2020.

[44] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[45] Thomas Porter and Tom Duff. Compositing digital images. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 253–259, 1984.

[46] Johann Radon. On the determination of functions from their integral values along certain manifolds. *IEEE transactions on medical imaging*, 5(4):170–176, 1986.

[47] Holger Roth, Amal Farag, Evrim B. Turkbey, Le Lu, Jiamin Liu, and Ronald M. Summers. Data from pancreas-ct, 2016. URL `https://www.cancerimagingarchive.net/collection/pancreas-ct/`.

[48] Darius Rückert, Yuanhao Wang, Rui Li, Ramzi Idoughi, and Wolfgang Heidrich. Neat: Neural adaptive tomography. *ACM Transactions on Graphics (TOG)*, 41(4):1–13, 2022.

[49] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.

[50] Jason Sanders and Edward Kandrot. *CUDA by example: an introduction to general-purpose GPU programming*. Addison-Wesley Professional, 2010.

[51] Ken Sauer and Charles Bouman. A local update strategy for iterative reconstruction from projections. *IEEE Transactions on Signal Processing*, 41(2):534–548, 1993.

[52] William C Scarfe, Allan G Farman, Predag Sukovic, et al. Clinical applications of cone-beam computed tomography in dental practice. *Journal-Canadian Dental Association*, 72(1):75, 2006.

[53] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016.

[54] Liyue Shen, John Pauly, and Lei Xing. Nerp: implicit neural representation learning with prior embedding for sparsely sampled image reconstruction. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[55] Emil Y Sidky and Xiaochuan Pan. Image reconstruction in circular cone-beam computed tomography by constrained, total-variation minimization. *Physics in Medicine & Biology*, 53(17):4777, 2008.

[56] The Finnish Inverse Problems Society. X-ray tomographic datasets, 2024. URL `https://fips.fi/category/open-datasets/x-ray-tomographic-datasets/`.

[57] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=UyNXMqnN3c`.

[58] Pieter Verboven, Bart Dequeker, Jiaqi He, Michiel Pieters, Leroi Pols, Astrid Tempelaere, Leen Van Doorselaer, Hans Van Cauteren, Ujjwal Verma, Hui Xiao, et al. www. x-plant. org-the ct database of plant organs. In *6th Symposium on X-ray Computed Tomography: Inauguration of the KU Leuven XCT Core Facility, Location: Leuven, Belgium*, 2022.

[59] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[60] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[61] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[62] Taoran Yi, Jiemin Fang, Junjie Wang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. Gaussiandreamer: Fast generation from text to 3d gaussians by bridging 2d and 3d diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[63] Xingde Ying, Heng Guo, Kai Ma, Jian Wu, Zhengxin Weng, and Yefeng Zheng. X2ct-gan: reconstructing ct from biplanar x-rays with generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10619–10628, 2019.

[64] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[65] Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient high-quality compact surface reconstruction in unbounded scenes. *arXiv preprint arXiv:2404.10772*, 2024.

[66] Guangming Zang, Ramzi Idoughi, Rui Li, Peter Wonka, and Wolfgang Heidrich. Intratomo: self-supervised learning-based tomography via sinogram synthesis and prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1960–1970, 2021.

[67] Ruyi Zha, Yanhao Zhang, and Hongdong Li. Naf: Neural attenuation fields for sparse-view cbct reconstruction. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 442–452. Springer, 2022.

[68] Zheng Zhang, Wenbo Hu, Yixing Lao, Tong He, and Hengshuang Zhao. Pixel-gs: Density control with pixel-aware gradient for 3d gaussian splatting. *arXiv preprint arXiv:2403.15530*, 2024.

[69] Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. Ewa splatting. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):223–238, 2002.

A Transformation module in X-ray rasterization

The configuration of a cone beam CT scanner is shown in Fig. 9. The X-ray source and detector plane rotate around the z-axis, resembling a pinhole camera model. Therefore, we can formulate the field-of-view (FOV) of a scanner as

$$FOV_x = 2 \cdot \arctan(\frac{D_x}{2L_{SD}}), FOY_y = 2 \cdot \arctan(\frac{D_y}{2L_{SD}}). \tag{9}$$

Here, (D_x, D_y) is the physical size of the detector plane, and L_{SD} is the distance between the source and the detector. Following [23], we then use FOVs to determine the projection mapping ϕ.

To get Gaussians in the ray space, we first transfer them from the world space to the scanner space. The scanner space is defined such that its origin is the X-ray source, and its z-axis points to the projection center. The transformation matrix $\mathbf{T}$ from the world space to the scanner space is

$$\mathbf{T} = \begin{bmatrix} \mathbf{W} & \mathbf{t} \\ 0 & 1 \end{bmatrix}, \mathbf{W} = \begin{bmatrix} -\sin\theta & \cos\theta & 0 \\ 0 & 0 & -1 \\ -\cos\theta & -\sin\theta & 0 \end{bmatrix}, \mathbf{t} = \begin{bmatrix} 0 \\ 0 \\ L_{SO} \end{bmatrix}. \tag{10}$$

Here, ϕ is the rotation angle, and L_{SO} is the distance between the source and the object. Next, we apply local approximation on each Gaussian. The Jacobian of the affine approximation $\mathbf{J}_i$ is the same as Eq. (29) in [69]. Finally, we have the Gaussian in the ray space with new position $\tilde{\mathbf{p}}$ and covariance $\tilde{\mathbf{\Sigma}}_i$ as

$$\tilde{\mathbf{p}}_i = \phi(\mathbf{p}), \tilde{\mathbf{\Sigma}}_i = \mathbf{J}_i \mathbf{W} \mathbf{\Sigma}_i \mathbf{W}^\top \mathbf{J}_i^\top. \tag{11}$$

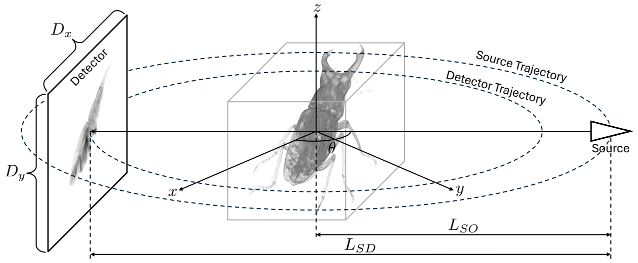

Figure 9: Configuration of a cone-beam CT scanner.

B Details of dataset

Synthetic data We evaluate methods with various modalities, covering major CT applications such as medical diagnosis, biological research, and industrial inspection. The synthetic dataset consists of 15 cases across three categories: human organs (chest, foot, head, jaw, and pancreas), animals and plants (beetle, bonsai, broccoli, kingsnake, and pepper), and artificial objects (backpack, engine, present, teapot, and mount). The chest and pancreas scans are from LIDC-IDRI [4] and Pancreas-CT [47], respectively. Broccoli and pepper are obtained from X-Plant [58], and the rest are from SciVis [26]. Following [67, 6], we preprocess raw data by normalizing densities to $[0, 1]$ and resizing volumes to $256 \times 256 \times 256$. We then use the tomography toolbox TIGRE [5] to capture 512×512 projections in the range of $0° \sim 360°$. We add two types of noise: Gaussian (mean 0, standard deviation 10) as electronic noise of the detector and Poisson (lambda 1e5) as photon scattering noise. All volumes and their projection examples are shown in Fig. 10.

Real-world data We use FIPS [56], a public dataset providing real 2D X-ray projections. FIPS includes three objects (pine [41], seashell [21], and walnut [42]). Each case has 721 projections in the range of 0°-360°. We preprocess 2D projections by resizing them to $560x560$ and normalizing them to $[0, 1]$. Since ground truth volumes are unavailable, we use FDK to create pseudo-ground truth with all views and then subsample 75/50/25 views for sparse-view experiments. The size of the target volume is $256 \times 256 \times 256$.

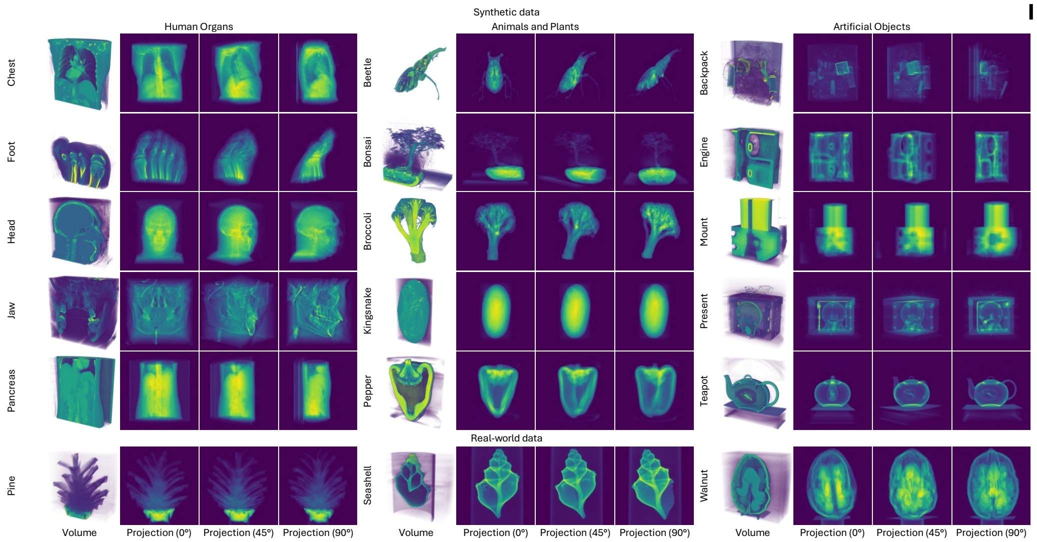

Figure 10: Datasets used for experiments. We show half volume and projection examples for each case.

C Implementation details of baseline methods

For fairness, we do not compare methods that require external training data but focus on those that solely use 2D projections of arbitrary objects. We run traditional algorithms FDK, SART, and ASD-POCS with GPU-accelerated tomographic toolbox TIGRE [5], and select three SOTA NeRF-based tomography methods. IntraTomo models the density field with a large MLP. NAF accelerates the training process by hash encoding. SAX-NeRF achieves plausible results with a line segment-based transformer. We use the official code of NAF and SAX-NeRF and conduct experiments with default hyperparameters. The IntraTomo implementation is sourced from the NAF repository. The training iterations of NeRF-based methods are set to 150k (default of NAF and SAX-NeRF). All methods are run on a single RTX 3090 GPU.

D More qualitative results

Main results We visualize more reconstruction results in Fig. 11 and Fig. 12. FDK and SART introduce notable streak artifacts, while ASD-POCS and IntraTomo blur structural details. NeRF-based solutions perform better than traditional methods but exhibit salt-and-pepper noise. In comparison, our method successfully recovers sharp details and maintains smoothness in homogeneous areas.

Integration bias We show more qualitative comparisons of X-3DGS and ours in Fig. 13. Our method outperforms X-3DGS in both 2D rendering and 3D reconstruction.

Components and parameters We visually compare different components and parameters in Fig. 14. Our newly introduced components improve the reconstruction quality. Our parameter setting also yields the best performance.

Convergence analysis We show the PSNR and SSIM plots in Fig. 15. Our method converges significantly faster than NeRF-based methods and outperforms them in only 3000 iterations.

E More discussion of limitation

Varying time We present the training times for all cases in Fig. 16. The training time varies across cases, primarily due to the different numbers of kernels used. Our method takes more time on objects

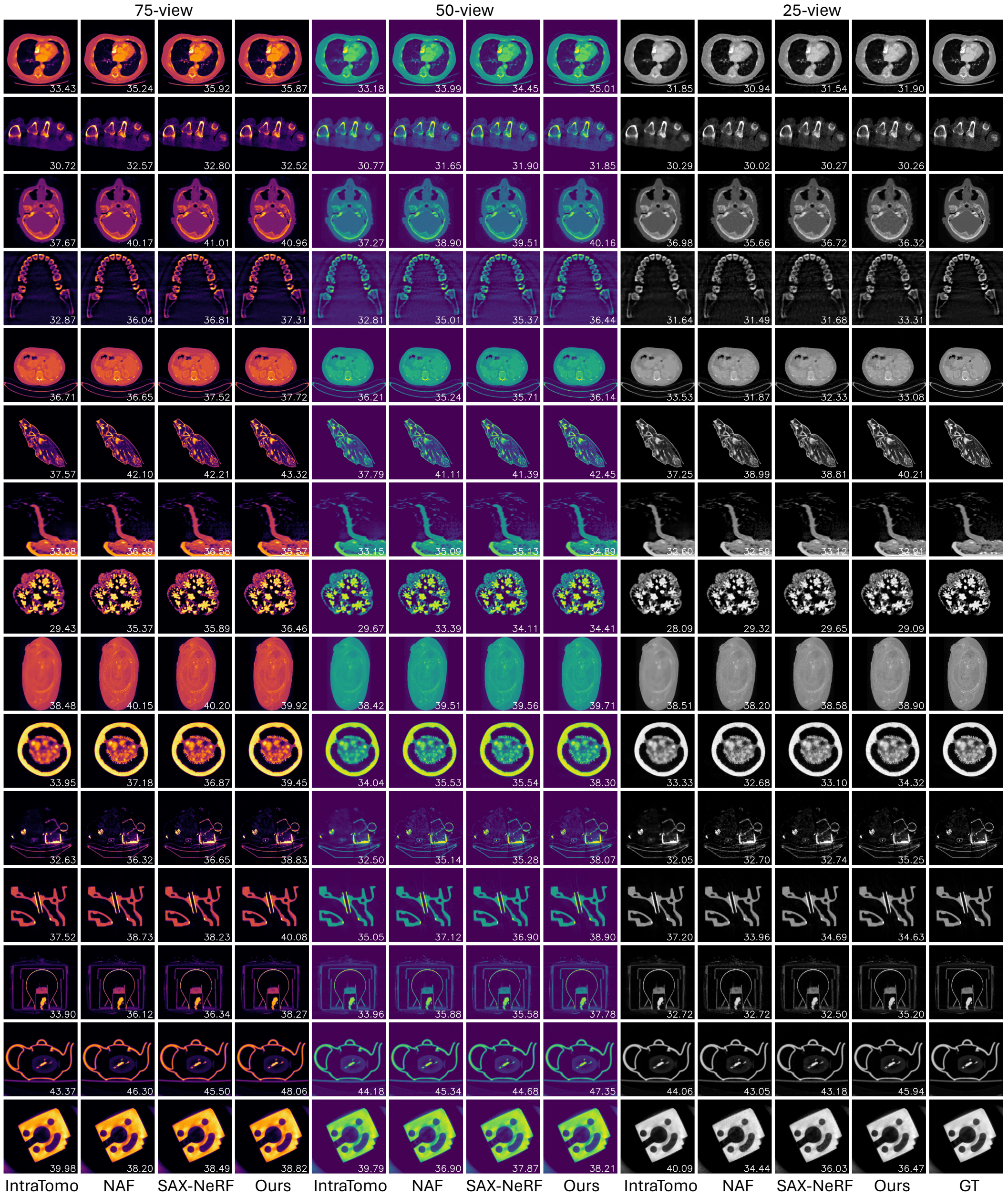

Figure 11: Reconstruction results of NeRF-based methods and our method on the synthetic dataset.

with large homogeneous areas, such as the chest, pancreas, and mount, and less time on those with sparse structures, such as the beetle, backpack, and present.

Needle-like artifacts While our method achieves the highest reconstruction quality, it introduces needle-like artifacts, especially under the 25-view condition (Fig. 17). This suggests that some Gaussians may overfit specific X-rays. Similar artifacts are also observed in 3DGS [68].

Extrapolation ability While this paper focuses on sparse-view CT (SVCT), we also test R^2-Gaussian on limited-angle CT (LACT), where the scanning range is constrained to less than $180°$. Unlike SVCT, which highlights the interpolation ability of methods, LACT challenges their extrapolation ability, i.e., estimating unseen areas outside the scanning range. We generate 100 projections within ranges of $0° \sim 150°$, $0° \sim 120°$, and $0° \sim 90°$. The quantitative results in Tab. 4 show that

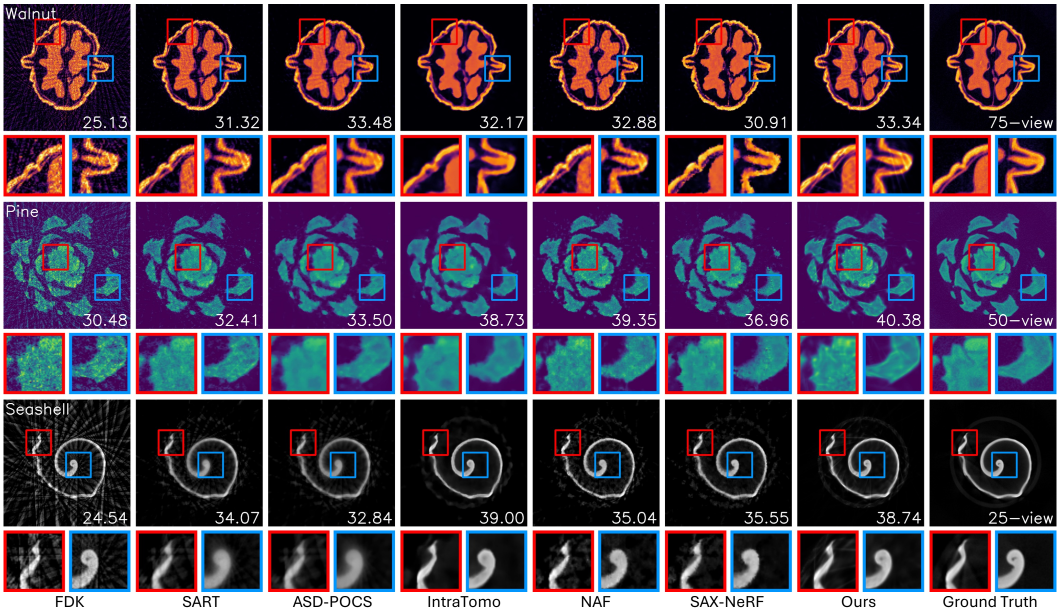

Figure 12: Reconstruction results on the real-world dataset.

our method has slightly lower PSNR but higher SSIM than the NeRF-based method NAF. Visualization results in Fig. 18 indicate that our method recovers more details in scanned areas but exhibits blurred artifacts in unseen areas. We attribute this performance drop to the nature of the networks and kernels. Given the gradient of a ray, NeRF updates the entire network while 3DGS individually optimizes intersected kernels. Thus, NeRF has better global awareness and consistency, while 3DGS is more local-oriented and has suboptimal extrapolation ability.

Calibration error In real-world applications, calibration errors can affect reconstruction quality. For example, tomography requires a reference image I_0 in Eq. (1) to represent the illumination pattern without the object. This reference image may have artifacts, such as intensity dropoff towards the image boundaries and other non-uniformities in illumination. Additionally, scanner extrinsics and intrinsics may vary during scanning due to heat expansion and mechanical vibrations. Addressing these practical challenges will be the focus of future work.

Anisotropic effects Following existing CT reconstruction methods, we work under the isotropic assumption of X-ray imaging. However, in the real world, some X-ray transport effects, such as Compton scattering, are anisotropic. We do not explicitly model them but treat them as noises on X-ray projections. This is a necessary simplification for CT reconstruction but may induce inaccuracy for novel-view X-ray synthesis. Readers may refer to [7, 14] for using 3DGS for X-ray view synthesis.

Table 4: Quantitative evaluation on limited-angle tomography. We colorize the best, second-best, and third-best numbers.

Methods	$0^\circ \sim 150^\circ$			$0^\circ \sim 120^\circ$			$0^\circ \sim 90^\circ$		
	PSNR↑	SSIM↑	Time↓	PSNR↑	SSIM↑	Time↓	PSNR↑	SSIM↑	Time↓
FDK [13]	26.83	0.570	-	24.00	0.566	-	21.22	0.547	-
SART [2]	33.34	0.883	7m9s	30.21	0.847	7m8s	26.71	0.795	7m59s
ASD-POCS [55]	33.16	0.913	3m41s	29.76	0.875	3m39s	26.34	0.812	4m8s
NAF [67]	36.29	0.940	27m18s	33.35	0.922	27m6s	29.89	0.884	27m25s
Ours	36.12	0.948	9m3s	32.68	0.923	8m36s	29.21	0.886	8m28s

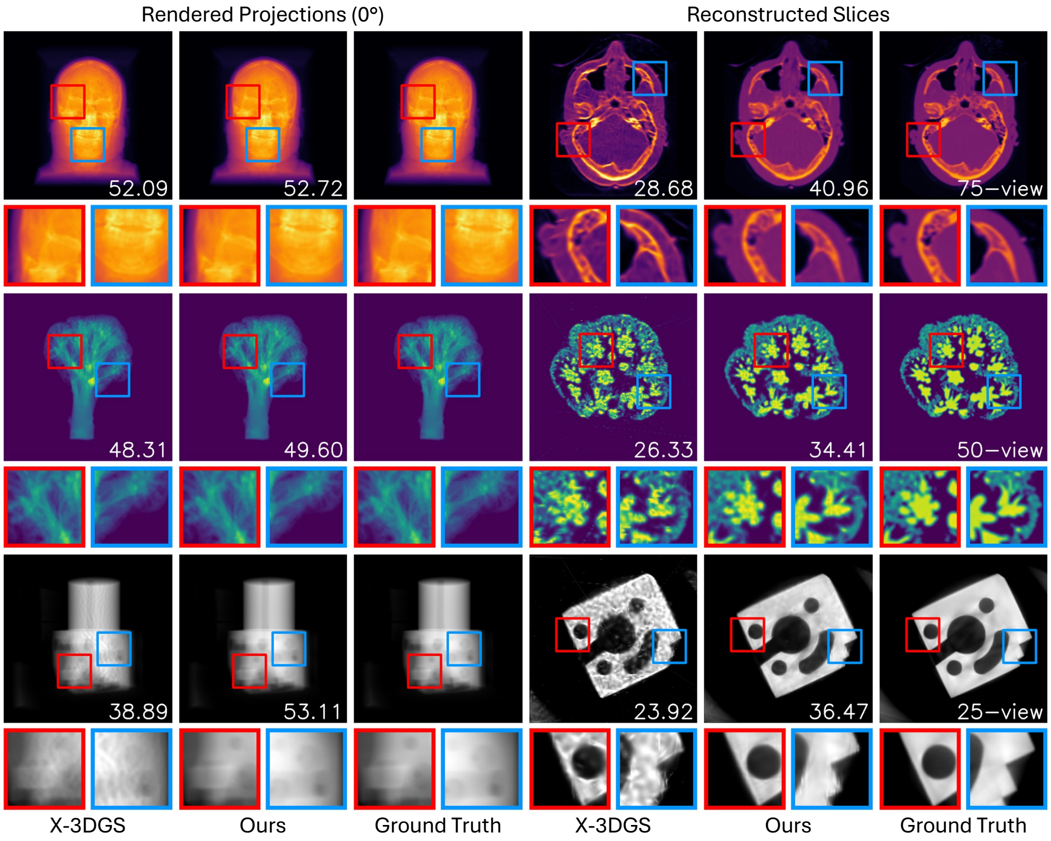

Figure 13: Qualitative comparison of X-3DGS and our method.

F Broader impacts

Impacts on real-world applications Computed tomography is an essential imaging technique that is widely used in fields including medicine, biology, industry, etc. Our R^2-Gaussian enjoys superior reconstruction performance and fast convergence speed, making it promising to be implemented in real-world applications such as medical diagnosis and industrial inspection.

Impacts on research community We discover a previously unknown integration bias problem in currently popular 3DGS. we speculate that this problem could be universal across all 3DGS-related works. Therefore, our rectification technique may apply to wider practical domains, not limited to tomography but also other tasks such as magnetic resonance imaging (MRI) reconstruction and volumetric-based surface reconstruction.

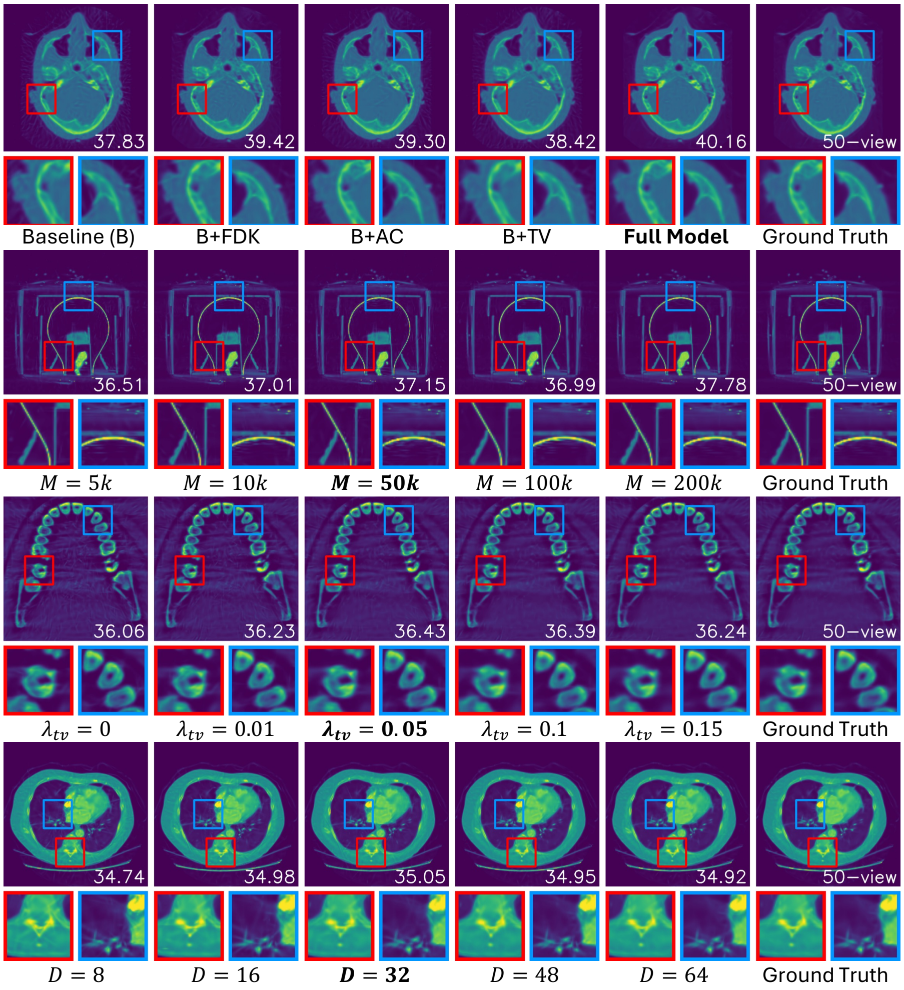

Figure 14: Quantitative comparison of different components and parameters.

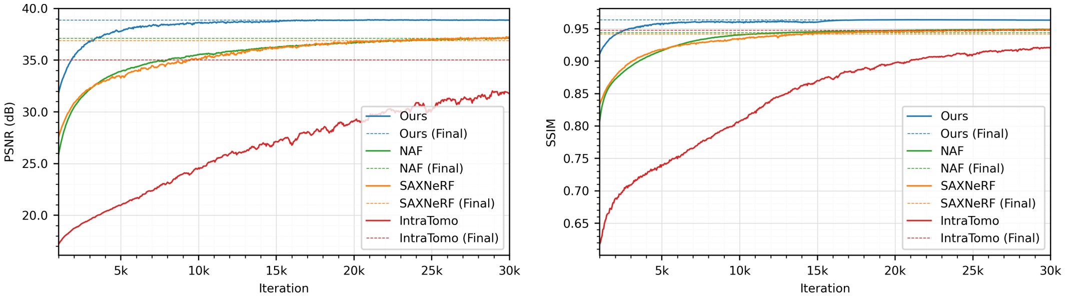

Figure 15: PSNR-iteration and SSIM-iteration plots of case *engine, 50-view*.

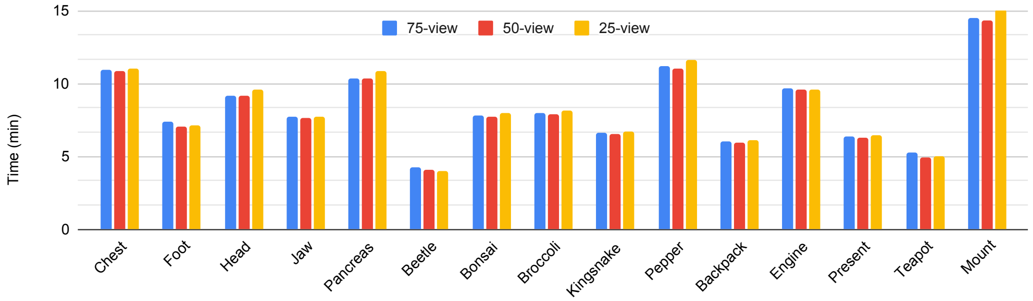

Figure 16: Training time on the synthetic dataset.

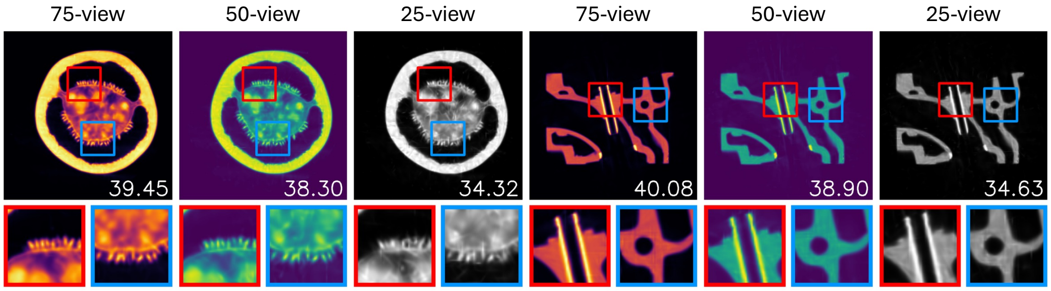

Figure 17: 3DGS-based methods tend to introduce needle-like artifacts when there are insufficient amounts of images. PNSR (dB) is shown at the bottom right of each image.

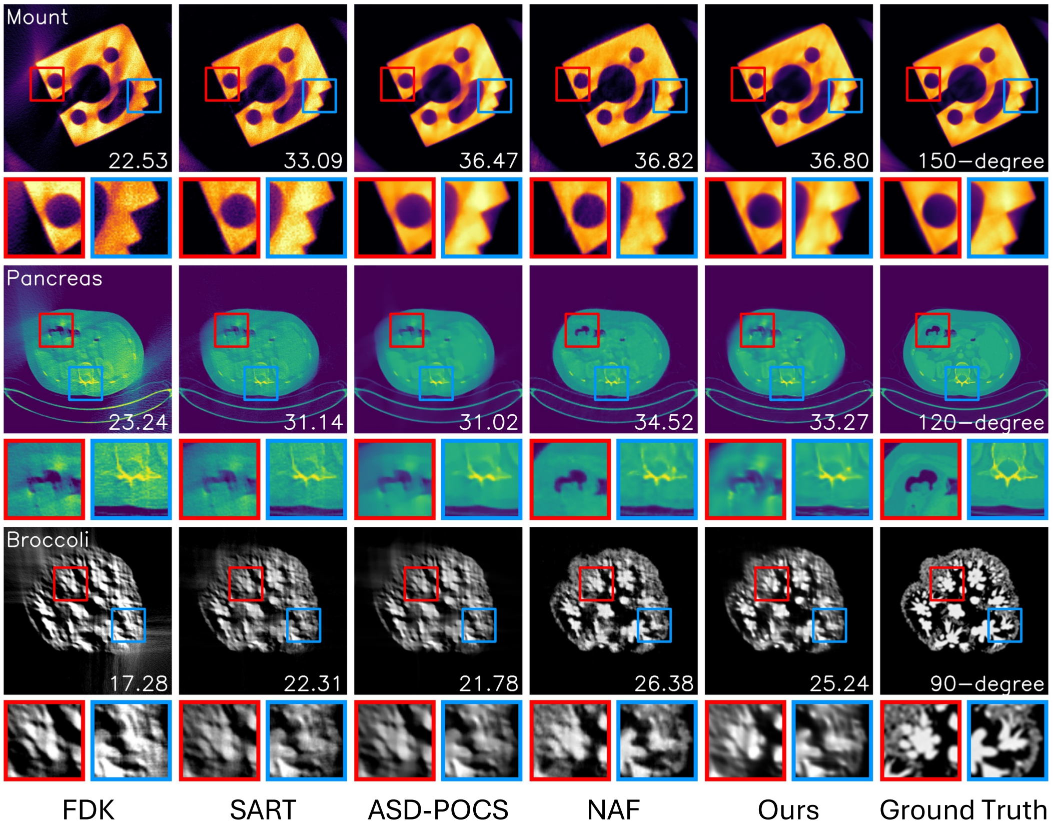

Figure 18: Visualization of reconstruction results under limited-angle scenarios. PNSR (dB) is shown at the bottom right of each image.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: In the abstract and introduction, we clearly highlight our contribution and scope as a discovery of integration bias problem in 3DGS and a novel 3DGS-based framework for tomographic reconstruction. We discuss details of our claim in Sec. 4 and validate it in Sec. 5.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitation in Sec. 6. We further show more quantitative and qualitative analysis of limitations in Appendix E.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We carefully derive the X-ray rasterization functions with clear assumptions and numbered formulas as shown in Sec. 4.2.1 and Appendix A.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Our work is based on the 3DGS framework with crucial modifications clearly described in Sec. 4. We provide implementation details, including all hyperparameters and training strategies in Sec. 5.1. We also attach our code in the supplementary material for reproducibility.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We will release the data and code upon acceptance. We attach the code in the supplementary material to reproduce experimental results.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify all training and testing details, including dataset setting, hyperparameters, and type of optimizer in Sec. 5.1. We discuss hyperparameter analysis in Sec. 5.3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: While our work and baseline methods involve some random operations, their influence on final results is very limited, so we do not include the error bars. Besides, baseline methods do not provide error bars in their papers for reference.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We clearly provide details of computer resources for our method and baseline methods in Sec. 5.1 and Appendix C.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research strictly follows the NeurIPS Code of Ethics. We ensure anonymity for the submission.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We briefly introduce the potential societal impacts in Sec. 6 and discuss them in detail in Appendix F.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We properly credit the original codes and datasets in our paper and code. The license and terms of use are mentioned along with the code in the supplementary material.

Guidelines:

- The answer NA means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide a detailed readme file along with the code in the supplementary material, which shows details about training, license, etc.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our work does not involve crowdsourcing nor research with human subjects

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer:[NA]

    Justification: Our work does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.