Appendix

A. Overview

Thank you for reading the Appendix of our research. This appendix is organized as follows:

- Section B provides a detailed explanation of the experimental setup.
- Section C describes the evaluation metrics used in our study.
- Section D discusses the application of different loss functions, types of Gaussians, and voxelizers.
- Section E elaborates on the Fast Volume Reconstruction method and analyzes its computational complexity.
- Section F details the implementation of adaptive density control in this work.
- Section G presents additional visualizations of DGR reconstruction.

B. Experiment Details

B.1. Code and Reproducibility

Our code is publicly available at https://github.com/wskingdom/DGR and is meticulously organized for readability. Experiments are categorized into three groups, each with corresponding code: (1) Cone-Beam Sparse-View CT, (2) Fan-Beam Sparse-View CT, and (3) Fan-Beam Limited-Angle CT. To reproduce our results, please refer to the **Readme.md** file in our codebase.

B.2. Organization of Experiments

To accommodate diverse experimental settings in related research (e.g., varying projection numbers and scanning geometry), our experiments are divided into three groups:

Cone-Beam Sparse-View CT: Experiments with 75, 50, and 25 views were conducted on the FIPS dataset [9], aligning with the settings of R^2-Gaussian [12].

Fan-Beam Sparse-View CT: Experiments with 180, 120, 90, and 60 views were performed on the AAPM-Mayo LDCT dataset [8] and the FUMPE dataset [7], following the methodology of the advanced Deep Learning Reconstruction (DLR) method, SWORD [11].

Fan-Beam Limited-Angle CT: Experiments on 90° Limited-Angle CT were conducted on the AAPM-Mayo LDCT dataset [8], consistent with DiffusionMBIR [3].

C. Evaluation Metrics

C.1. Peak Signal-to-Noise Ratio (PSNR)

PSNR [4] is a widely used metric to evaluate the quality of the reconstructed images. It is defined as:

$$\text{PSNR}(x, y) = 10 \cdot \log_{10} \left(\frac{\text{MAX}^2}{\text{MSE}(x, y)} \right), \quad (15)$$

Table 6. Usage of Different Loss Functions

Cone-Beam Sparse-View CT (50 view, 300 iter)				
$\mathcal{L}_1$ Loss	SSIM Loss	TV Loss	PSNR	SSIM
✓			38.66	0.929
✓	✓		38.94	0.933
✓	✓	✓	**39.65**	**0.939**

Table 7. Usage of Different Loss Functions

Fan-Beam Sparse-View CT (60 view, 300 iter)				
$\mathcal{L}_1$ Loss	SSIM Loss	TV Loss	PSNR	SSIM
✓			38.82	0.931
✓	✓		38.88	0.933
✓	✓	✓	**39.15**	**0.936**

Table 8. Usage of Different Loss Functions

Fan-Beam Limited-Angle CT (90°, 300 iter)				
$\mathcal{L}_1$ Loss	SSIM Loss	TV Loss	PSNR	SSIM
✓			37.69	0.932
✓	✓		37.81	0.934
✓	✓	✓	**38.02**	**0.936**

where MAX is the maximum possible pixel value of the image and $\text{MSE}(x, y)$ is the mean squared error between the original and reconstructed images.

C.2. Structural Similarity Index (SSIM)

SSIM [10] is a metric that measures the similarity between two images. It is defined as:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \quad (16)$$

where μ_x and μ_y are the mean values of the images x and y, σ_x^2 and σ_y^2 are the variances of the images, σ_{xy} is the covariance of the images, and C_1 and C_2 are constants to stabilize the division when the denominator is small.

C.3. Evaluation Details

For the Cone-Beam 75/50/25-view Sparse-View CT experiments, we assess performance on the FIPS dataset [9] and maintain its original data split for consistency.

In the Fan-Beam 180/120/90/60-view Sparse-View CT experiments, we evaluate the performance of the reconstruction methods over the entire 3D volume, as established in SWORD [11]. In this context, x and y in the PSNR and SSIM metrics refer to the reconstructed 3D volume and the ground truth 3D volume, respectively.

For the 90° Limited-Angle CT experiments, we evaluate performance across axial, coronal, and sagittal slices fol-

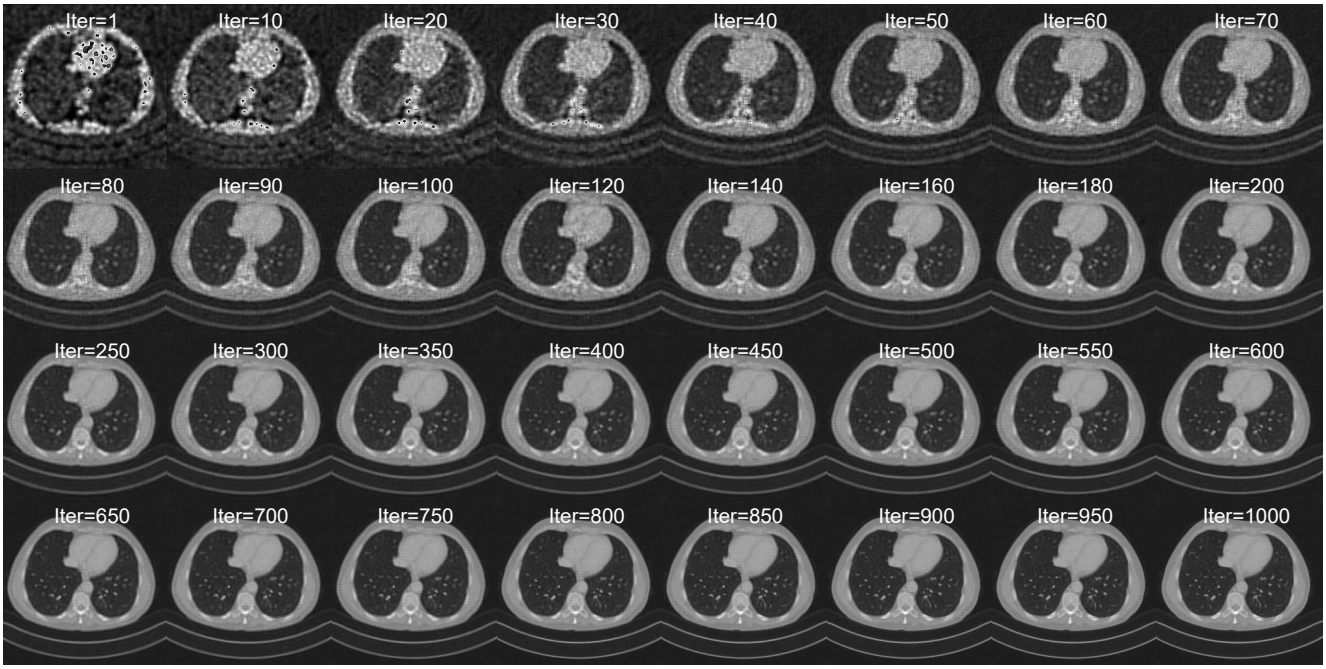

Figure 4. This figure illustrates the reconstruction process by iterations of Fan-Beam 60-view CT. The positions of Gaussians are initialized using Filtered Back Projection, as described in the experimental settings.

lowing DiffusionMBIR [3]. Here, x and y in the PSNR and SSIM metrics denote the reconstructed 2D slice and the ground truth 2D slice, respectively. The mean PSNR and SSIM values are then computed across all slices.

D. Discussions

D.1. Ablation on Loss Functions

In the main text, we use different loss functions together to enhance reconstruction quality. Specifically, the SSIM loss is computed in the projection domain to preserve structural information, while the TV loss is computed in the volume domain to promote sparsity in the reconstructed volume.

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_1(\hat{P}, P) + \lambda_2 \mathcal{L}_{\text{SSIM}}(\hat{P}, P),$$

where $\lambda_1 = 0.8$ and $\lambda_2 = 0.2$ are the weights of the $\mathcal{L}_1$ Loss and SSIM loss, respectively.

A combined use of $\mathcal{L}_1$ Loss, SSIM loss, and TV loss is as follows:

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_1(\hat{P}, P) + \lambda_2 \mathcal{L}_{\text{SSIM}}(\hat{P}, P) + \lambda_3 \mathcal{L}_{\text{TV}}(V),$$

where $\lambda_1 = 0.6$, $\lambda_2 = 0.2$, and $\lambda_3 = 1$ are the weights of the $\mathcal{L}_1$ Loss, SSIM loss, and TV loss, respectively.

The results for these loss combinations are presented in Table 6, Table 7, and Table 8, corresponding to Cone-Beam Sparse-View CT, Fan-Beam Sparse-View CT, and Fan-Beam Limited-Angle CT experiments on the real-world FIPS dataset [9]. The results demonstrate that the combined use of $\mathcal{L}_1$ Loss, SSIM loss, and TV loss achieves the best performance in terms of PSNR and SSIM metrics.

D.2. Isotropic Gaussians vs Anisotropic Gaussians

The use of isotropic Gaussians is justified by the generally isotropic nature of tissue attenuation properties in CT, especially for soft tissues, which exhibit minimal directional dependence. As noted in Principles of Computerized Tomographic Imaging [5], small elements of a distributed source can be treated as isotropic, reflecting tissue behavior.

To further validate this point, we conduct ablation studies on 50-view Sparse-View CT of Table 1 (Synthetic dataset), repeating experiments 10 times and reporting Mean $\pm$ SD in Table 10 to evaluate the result of substituting our isotropic Gaussians with anisotropic Gaussians. Results also validate that using isotropic Gaussians can not only achieve better image quality and stability but also reduce training time.

D.3. Comparison with Other Voxelizer

We also conduct ablation studies on 50-view Sparse-View CT of Table 1 (Synthetic dataset) and repeat experiments 10 times and reporting Mean $\pm$ SD in Table 11 to evaluate the result of substituting our Fast Volume Reconstruction with the R^2-Gaussian voxelizer. Results show that our reconstruction module is faster than R^2-Gaussian voxelizer. Besides, our reconstruction module boosts the Gaussian optimization jointly, which achieves better performance.

Table 9. Comparisons of 180/120/90/60-view Sparse-View CT on FUMPE dataset. Best in **Bold.**

Method	Extra Data	180-view		120-view		90-view		60-view	
		PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
FBP [1]	0	33.74	0.859	30.65	0.803	28.52	0.757	25.59	0.592
MCG [2]	8341	36.93	0.898	37.05	0.899	37.03	0.899	35.46	0.900
DiffusionMBIR [3]	8341	36.78	0.914	36.50	0.907	36.79	0.918	34.86	0.918
SWORD [11]	8341	41.50	0.967	39.69	0.959	37.77	0.948	31.64	0.910
DGR	0	**41.78**	**0.978**	**40.13**	**0.969**	**38.13**	**0.961**	**36.48**	**0.923**

Table 10. Replace isotropic Gaussians by Anisotropic Gaussians

	PSNR↑	SSIM↑	Time↓
DGR	**38.74± 0.07**	**0.960± 0.01**	**7m13s**
w/ anisotropic Gaussian	38.71± 0.17	0.960± 0.02	8m05s

Table 11. Replace our voxelizer by R^2-Gaussian Voxelizer

	PSNR↑	SSIM↑	Time↓
DGR	**38.74± 0.07**	**0.960± 0.01**	**7m13s**
w/ R^2-Gaussian Voxelizer	38.02± 0.22	0.956± 0.02	8m37s

E. Fast Volume Reconstruction

E.1. Highly Parallelized Implementation

We show the detailed implementation of our Fast Volume Reconstruction In Algorithm 1 and our code. The algorithm takes the mean μ, covariance C, and intensity I of the Gaussians as input and reconstructs the 3D volume in a fast and efficient manner. Notably, this process is implemented in a highly parallelized manner to accelerate the computation without sacrificing the reconstruction quality. More details can be found in our implementation code.

E.2. Analysis of Complexity

Method	VRAM (GiB)	Time (s)
Direct Reconstruction (estimated)	16662.50	/
FVR w/o Decomposition	16.87	1.05
FVR w/ Decomposition	16.87	0.09

As mentioned in the ablation study, the proposed fast volume reconstruction saves both time and space consumption. This difference in complexity primarily stems from the computation of the Mahalanobis distance, D^2.

Direct Reconstruction In the direct reconstruction approach, the Mahalanobis distance is calculated as $D^2 = (P-\mu)^\top \Sigma^{-1}(P-\mu)$. Here, $P \in \mathbb{R}^{w\times h\times c\times d}$ represents the position of a voxel in the volume, where $(d=3)$ indicates the spatial dimension. $\mu \in \mathbb{R}^{n\times d}$ is the set of means for n Gaussians, and $\Sigma \in \mathbb{R}^{n\times d\times d}$ is the covariance matrix for these Gaussians. The computational complexity for direct reconstruction is $O(n \cdot w \cdot h \cdot c \cdot d^2)$. Evidently, this method is not feasible for a large number of Gaussians due to its prohibitive memory consumption.

Fast w/o Decomposition In this condition, the Mahalanobis distance is calculated as $D^2 = \sum_d B''_{n,w_0,h_0,c_0,d}\ C^{-1}_{n,d,d}\ B''_{n,w_0,h_0,c_0,d}$. Here, w_0, h_0, and c_0 denote the dimensions of the local region. The computational complexity for this approach is $O(n \cdot w_0 \cdot h_0 \cdot c_0 \cdot d^2)$. Compared to direct reconstruction, the local region size $(w_0 \cdot h_0 \cdot c_0)$ is significantly smaller than the full volume size $(w \cdot h \cdot c)$, leading to a substantial reduction in memory consumption.

Fast Volume Reconstruction with Decomposition The decomposed form of Fast Volume Reconstruction yields the Mahalanobis distance as:

$$D^2 = \sum_d \left(B'_{w_0,h_0,c_0,d} - \Delta\mu_{n,1,1,d}\right) C^{-1}_{n,d,d} \left(B'_{w_0,h_0,c_0,d} - \Delta\mu_{n,1,1,d}\right)$$

The space complexity remains equivalent to that of fast volume reconstruction without decomposition. However, the time complexity is significantly reduced to $O(w_0 \cdot h_0 \cdot c_0 \cdot d)$, a benefit derived from the decomposition of the large matrix multiplication.

F. Adaptive Densification

Inspired by the adaptive density control utilized in 3D Gaussian Splatting [6], we implement this mechanism for our DGR, aiming to densify the Gaussians within the discretized 3D volume, employing techniques including cloning, splitting, and pruning. The densification strategy is periodically applied throughout the optimization process, striking a balance between the number of Gaussians and the reconstruction quality.

We clone the Gaussian in under-reconstructed regions into two Gaussians. Specifically, the original and cloned Gaussians share identical positions μ and covariances Σ, but their intensities are halved to preserve the total intensity. This process helps to capture the fine details in the under-reconstructed regions. While the gradients of the original Gaussians remain unchanged, the gradients of the cloned Gaussians are set to zero.

Conversely, in over-reconstructed regions, we split the Gaussian into two smaller Gaussians, keeping their scale σ in proportion to the original one. The positions μ of

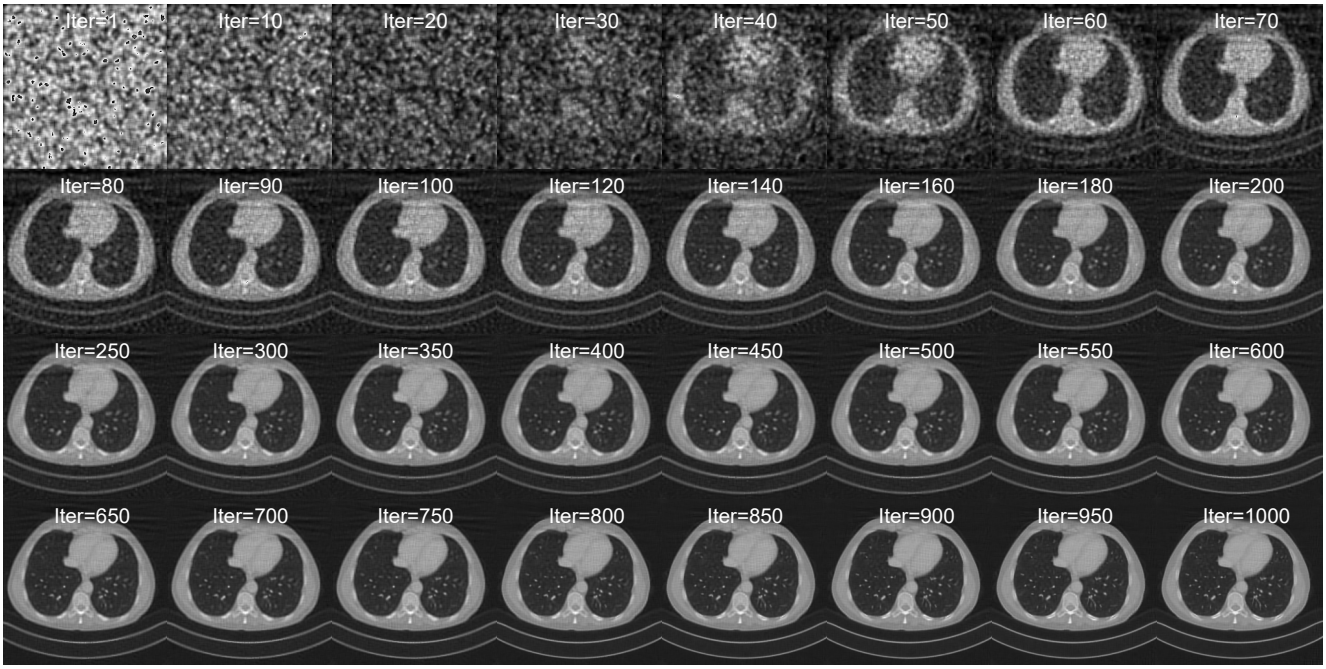

Figure 5. This figure illustrates the reconstruction process by iterations of Fan-Beam 90° Limited-Angle CT. The positions of Gaussians are randomly initialized for visualization comparison.

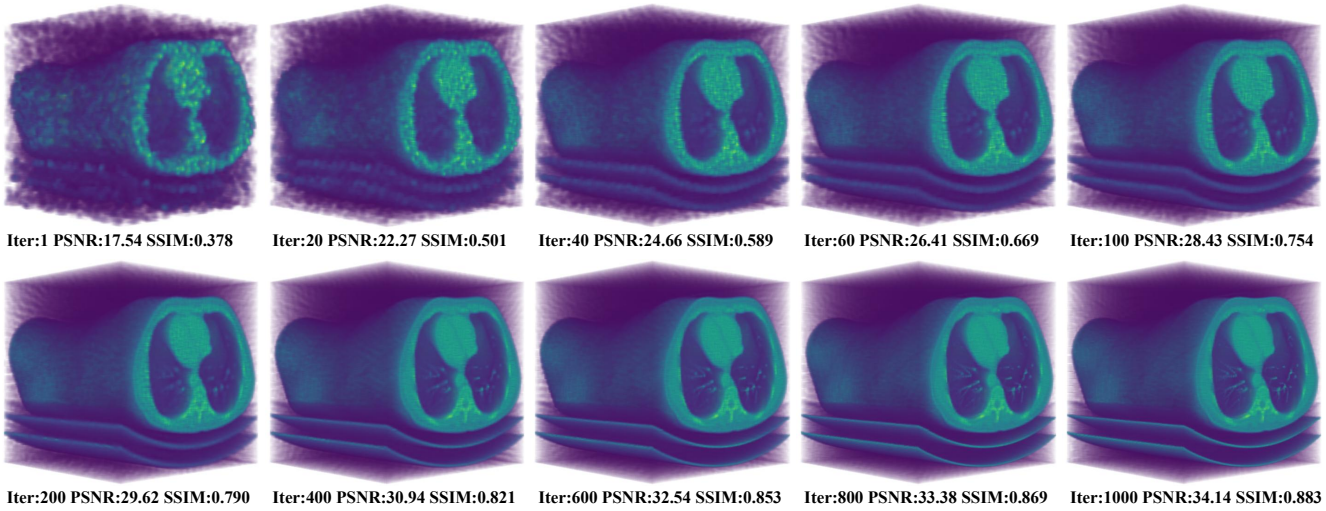

Figure 6. This figure illustrates the reconstruction process by iterations of Fan-Beam 60-view Sparse-View CT by 3D volume visualization.

these new Gaussians are derived from the probability density function (PDF) of the original Gaussian.

Furthermore, we implement a pruning mechanism to eliminate Gaussians that either possess near-zero gradient magnitudes or exhibit a scale σ exceeding three times the local box dimension. For these Gaussians, we simply remove them from the reconstruction volume to reduce the computational cost and prevent overfitting. More details of the algorithm are provided in Algorithm 2 and in our code.

G. Visualization

Figures 4, 5, and 6 visually depict the reconstruction process across various iterations, with the corresponding iteration number indicated for each image. For comparative purposes, Gaussians for Sparse-View CT are initialized as detailed in the main text, whereas those for Limited-Angle CT are initialized randomly. As the iterations progress, the reconstructed volumes consistently show steady improvement, demonstrating the effectiveness of DGR.

Algorithm 1 Fast Volume Reconstruction

Require: $\mu \in \mathbb{R}^{n \times d}$: mean of Gaussians ($d = 3$ for 3D space)
Require: $C \in \mathbb{R}^{n \times d \times d}$: covariance of Gaussians
Require: $I \in \mathbb{R}^{n \times 1}$: intensity of Gaussians
$V \leftarrow \mathbf{O}_{w,h,c}$ ▷ Initialize target volume with zeros
$B'_{w_0,h_0,c_0,d} \leftarrow \texttt{meshgrid}(\{-\frac{w_0-1}{2}, \ldots, \frac{w_0-1}{2}\}, \{-\frac{h_0-1}{2}, \ldots, \frac{h_0-1}{2}\}, \{-\frac{c_0-1}{2}, \ldots, \frac{c_0-1}{2}\})$ ▷ Initialize the shift of local regions
$\Delta\mu \leftarrow \mu - \lfloor \mu \rfloor$ ▷ Align μ to discrete voxel grid
$B'^T C^{-1} B'_{n,w_0,h_0,c_0} \leftarrow \sum_d B'_{w_0,h_0,c_0,d}\, C^{-1}_{n,d,d}\, B'_{w_0,h_0,c_0,d}$ ▷ Decompose large matrix multiplication
$B'^T C^{-1} \Delta\mu_{n,w_0,h_0,c_0} \leftarrow \sum_d B'_{w_0,h_0,c_0,d}\, C^{-1}_{n,d,d}\, \Delta\mu_{n,1,1,d}$
$\Delta\mu^T C^{-1} B'_{n,w_0,h_0,c_0} \leftarrow \sum_d \Delta\mu_{n,1,1,d}\, C^{-1}_{n,d,d}\, B'_{w_0,h_0,c_0,d}$
$\Delta\mu^T C^{-1} \Delta\mu_{n,1,1,1} \leftarrow \sum_d \Delta\mu_{n,1,1,d}\, C^{-1}_{n,d,d}\, \Delta\mu_{n,1,1,d}$
$\Gamma \leftarrow e^{-\frac{1}{2}(B'^T C^{-1} B' + B'^T C^{-1} \Delta\mu + \Delta\mu^T C^{-1} B' + \Delta\mu^T C^{-1} \Delta\mu)} \cdot I$ ▷ Compute the Gaussian contributions
$P_{n,w_0,h_0,c_0,d} \leftarrow \lfloor \mu \rfloor_{n,1,1,1,d} + B'$ ▷ Compute the voxel positions that the Gaussians will impact
$Valid_{n,w_0,h_0,c_0} \leftarrow (P_{n,w_0,h_0,c_0,0} \geq 0) \wedge (P_{n,w_0,h_0,c_0,0} < w) \wedge (P_{n,w_0,h_0,c_0,1} \geq 0) \wedge (P_{n,w_0,h_0,c_0,1} < h) \wedge (P_{n,w_0,h_0,c_0,2} \geq 0) \wedge (P_{n,w_0,h_0,c_0,2} < c)$ ▷ Get valid indices that are within the volume
$V \leftarrow \texttt{scatter\_add}(V, P[Valid], \Gamma[Valid])$ ▷ Accumulate the contributions at the valid indices in parallel
return V ▷ Reconstructed CT volume

Algorithm 2 Adaptive Densification

Require: $\mu \in \mathbb{R}^{n \times d}$: mean of Gaussians
Require: $\sigma \in \mathbb{R}^{n \times 1}$: standard deviation of Gaussians
Require: $I \in \mathbb{R}^{n \times 1}$: intensity of Gaussians
Require: n_{max}: Maximum allowed quantity of Gaussians (500K by default)
Require: τ: Minimum gradient value (2e-4 by default)
Require: θ: Threshold determining Gaussian classification as small or large (0.005 of body diagonal length by default)
Require: $size$: Box size of each Gaussian (17 by default)
for $iteration \leftarrow 100$ **to** max_iter **step** 100 **do**
$avg\_grad \leftarrow \mu.grad / iteration$
$mask_{clone} \leftarrow (avg\_grad \geq \tau) \wedge (\sigma \leq \theta)$
$available\_gaussians \leftarrow n_{max} - n$ ▷ Ensure the total number of Gaussians does not exceed the limit
$n_{clone} \leftarrow \min(available\_gaussians, \sum_{mask_{clone}})$
$sorted\_indices \leftarrow \texttt{argsort}(avg\_grad[mask_{clone}])$
$mask_{clone} \leftarrow mask_{clone} \wedge \texttt{top\_k}(sorted\_indices, n_{clone})$ ▷ Select top n_{clone} Gaussians to clone
$I[mask_{clone}] \leftarrow I[mask_{clone}]/2$ ▷ Halve the intensity of the cloned Gaussians to maintain the total intensity
$\mu_{clone} \leftarrow no\_grad(\mu[mask_{clone}]),\ \sigma_{clone} \leftarrow no\_grad(\sigma[mask_{clone}]),\ I_{clone} \leftarrow no\_grad(I[mask_{clone}])$
$\mu \leftarrow \mu \cup \mu_{clone},\ \sigma \leftarrow \sigma \cup \sigma_{clone},\ I \leftarrow I \cup I_{clone}, n \leftarrow n + n_{clone}$
$mask_{split} \leftarrow (avg\_grad \geq \tau) \wedge (\sigma > \theta)$
$available\_gaussians \leftarrow n_{max} - n$
$n_{split} \leftarrow \min(available\_gaussians, \sum_{mask_{split}})$
$sorted\_indices \leftarrow \texttt{argsort}(avg\_grad[mask_{split}])$
$mask_{split} \leftarrow mask_{split} \wedge \texttt{top\_k}(sorted\_indices, n_{split})$ ▷ Select top n_{split} Gaussians to split
$\mu_{new} \leftarrow PDF(\mu[mask_{split}], \sigma[mask_{split}], 2)$ ▷ Initialize new Gaussians by PDF sampling
$\sigma_{new} \leftarrow \sigma[mask_{split}]/\sqrt[3]{2}, I_{new} \leftarrow I[mask_{split}]$ ▷ Divide the standard deviation by $\sqrt[3]{2}$ to maintain the total volume
$\mu \leftarrow \mu \cup \mu_{new} - \mu_{split},\ \sigma \leftarrow \sigma \cup \sigma_{new} - \sigma_{split},\ I \leftarrow I \cup I_{new} - I_{split}, n \leftarrow n + n_{split}$
$mask_{prune} \leftarrow (avg\_grad) \leq \tau \vee (\sigma > 3 \times size)$
$\mu \leftarrow \mu[\neg mask_{prune}],\ \sigma \leftarrow \sigma[\neg mask_{prune}],\ I \leftarrow I[\neg mask_{prune}]$
end for
return μ, σ, I

References

[1] Ronald Newbold Bracewell and ACf Riddle. Inversion of fan-beam scans in radio astronomy. *Astrophysical Journal, vol. 150, p. 427*, 150:427, 1967. 3

[2] Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. *Advances in Neural Information Processing Systems*, 35:25683–25696, 2022. 3

[3] Hyungjin Chung, Dohoon Ryu, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Solving 3d inverse problems using pre-trained 2d diffusion models. ieee. In *CVF Conference on Computer Vision and Pattern Recognition*, page 6, 2023. 1, 2, 3

[4] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th international conference on pattern recognition*, pages 2366–2369. IEEE, 2010. 1

[5] Avinash C Kak and Malcolm Slaney. *Principles of computerized tomographic imaging*. SIAM, 2001. 2

[6] 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. 3

[7] Mojtaba Masoudi, Hamid-Reza Pourreza, Mahdi Saadatmand-Tarzjan, Noushin Eftekhari, Fateme Shafiee Zargar, and Masoud Pezeshki Rad. A new dataset of computed-tomography angiography images for computer-aided detection of pulmonary embolism. *Scientific data*, 5 (1):1–9, 2018. 1

[8] Taylor R Moen, Baiyu Chen, David R Holmes III, Xinhui Duan, Zhicong Yu, Lifeng Yu, Shuai Leng, Joel G Fletcher, and Cynthia H McCollough. Low-dose ct image and projection dataset. *Medical physics*, 48(2):902–911, 2021. 1

[9] The Finnish Inverse Problems Society. X-ray tomographic datasets, 2024. 1, 2

[10] 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. 1

[11] Kai Xu, Shiyu Lu, Bin Huang, Weiwen Wu, and Qiegen Liu. Stage-by-stage wavelet optimization refinement diffusion model for sparse-view ct reconstruction. *IEEE Transactions on Medical Imaging*, 2024. 1, 3

[12] Ruyi Zha, Tao Jun Lin, Yuanhao Cai, Jiwen Cao, Yanhao Zhang, and Hongdong Li. R^2-gaussian: Rectifying radiative gaussian splatting for tomographic reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 1