Volumetric Gaussian Splatting: Direct Optimization in Voxel Space for Efficient Medical Data Visualization

Valentin Kraft[1][0000−0003−4057−2452], Andrea Schenk[1][0000−0003−1806−8380], and Gabriel Zachmann[2][0000−0001−8155−1127]

[1] Fraunhofer Institute for Digital Medicine MEVIS, 28359 Bremen, Germany
[2] Computer Graphics and Virtual Reality Department, University of Bremen, 28359 Bremen, Germany

Abstract. We present a novel formulation of 3D Gaussian Splatting for volumetric data that departs from the conventional paradigm of multi-view image supervision. Instead of optimizing against rendered RGB images, our approach directly operates in voxel space using target volumes and region-of-interest constraints, enabling a volume-native optimization process. To this end, we introduce a differentiable splat-to-volume rasterization that projects anisotropic Gaussians into a voxel grid and allows direct optimization of spatial and radiometric parameters.
Each Gaussian explicitly encodes a physical intensity value, such as Hounsfield units in computed tomography, enabling a volume-rendering-like representation with post hoc transfer function control and flexible visualization without retraining.
Our volume-informed initialization and tailored optimization enable fast convergence within minutes (e.g., 20–50 minutes in our setup), achieving accurate reconstructions (PSNR $\approx$ 23–33) that are visually comparable to classical volume rendering, while reaching substantially higher frame rates—making the method particularly suitable for resource-constrained environments such as mixed reality devices.
Overall, our work reframes Gaussian Splatting as a direct volumetric optimization problem, opening new directions for efficient and controllable visualization of medical data.
The implementation is available at https://github.com/valentinkraft/vgs.

Keywords: Gaussian Splatting · Medical Visualization · Volume Rendering · Differentiable Rendering · Volumetric Optimization

1 Introduction

Recent advances in medical image analysis have led to highly accurate and increasingly automated segmentation of anatomical structures, driven by methods such as TotalSegmentator [21] and foundation models like SAM [2]. While these developments enable rich semantic understanding of volumetric data, efficient and high-quality visualization of such data—particularly in immersive environments—remains a significant challenge.

Volume rendering is the de facto standard for visualizing medical data, preserving full volumetric information and enabling flexible exploration via transfer functions. However, high visual fidelity demands substantial computational resources, limiting applicability on resource-constrained devices such as mixed reality (MR) headsets. Streaming-based approaches—transmitting rendered images from powerful servers to lightweight clients—mitigate this but introduce latency, system complexity, and failure modes undesirable in clinical settings.

3D Gaussian Splatting [10] has emerged as a highly efficient rendering paradigm for real-time scene visualization, yet remains fundamentally image-based: scenes are learned from multi-view RGB observations under known camera parameters, requiring pose estimation, feature extraction, and image-domain supervision. This is poorly aligned with medical imaging, where ground truth is directly available in voxel space as physically meaningful intensity values (e.g., Hounsfield units). Optimizing through rendered images introduces unnecessary indirection, projection errors, and pipeline complexity.

We therefore reformulate 3D Gaussian Splatting as a direct volumetric optimization problem, optimizing anisotropic Gaussians against a target volume guided by region-of-interest (ROI) constraints—eliminating camera models, pose estimation, and intermediate rendering. Each splat explicitly encodes a physical intensity value, enabling post-hoc transfer function control without retraining, supported by a differentiable splat-to-volume rasterization for end-to-end optimization in voxel space. While not aiming to replace high-quality volume rendering in terms of visual fidelity, the method offers a compelling efficiency trade-off for interactive and immersive applications.

Our main contributions are as follows:

- **Volume-supervised 3D Gaussian Splatting.** We replace multi-view RGB supervision with direct voxel-space optimization via a differentiable splat-to-volume rasterization, enabling end-to-end optimization against a target volume and ROI mask.
- **Intensity-aware volumetric Gaussian representation.** Each splat encodes a physical intensity value (e.g., Hounsfield units), enabling post-hoc transfer function modification without retraining.
- **Efficient real-time volumetric rendering.** The compact Gaussian representation provides real-time, volume-rendering-like visualization suitable for resource-constrained environments such as mixed reality devices.

2 Related Work

Gaussian Splatting and Neural Rendering: 3D Gaussian Splatting [10] represents scenes as sets of anisotropic Gaussians optimized from multi-view images, enabling real-time rendering with high visual fidelity. Extensions have demonstrated near-photorealistic anatomical visualization via path tracing [17], relightable representations [7], and direction-dependent volumetric effects [8, 9]. Volume-rendering interaction concepts such as clipping [14] and transfer-function-like behavior [19] are beginning to migrate to Gaussian Splatting. Despite these advances, all existing approaches remain tied to image-based supervision—even volumetric extensions rely on rendering-based optimization rather than directly leveraging voxel-space ground truth. Most closely related is the concurrent work on Volume Encoding Gaussians (VEG) [6], which similarly stores per-Gaussian scalar values to enable transfer-function-agnostic rendering. However, VEG trains from pre-rendered multi-view images using an opacity-guided multi-transfer-function strategy, whereas we optimize directly against the CT volume in voxel space—eliminating the need for rendered training images entirely and encoding physically meaningful Hounsfield unit values per Gaussian. Kleinbeck et al. [11] propose a layered Gaussian Splatting representation for immersive anatomy visualization, pre-rendering CT data via path tracing to train separate Gaussian layers for different anatomical structures. While effective for layer-switching at render time, color and opacity are baked into the Gaussians, precluding post-hoc transfer function adjustments—a limitation our voxel-native, HU-aware formulation avoids.

Volume Rendering and Splatting Techniques: Volume rendering [5] is the established standard for medical data visualization, with photorealistic extensions via path tracing [13]. Early splatting-based approaches [22] share conceptual similarities with modern methods by approximating volume rendering using Gaussian kernels. Despite significant progress, volume rendering remains computationally demanding for large-scale medical datasets [3].

Efficient Medical Visualization and Mixed Reality: NeRF-based methods such as MedNeRF [4] have adapted generative radiance fields to reconstruct CT projections from sparse X-ray views. However, these approaches train from rendered image projections rather than volumetric data directly, and support neither real-time rendering nor interactive transfer function control—properties central to our method. Surveys [16, 20] highlight both the potential and limitations of mixed reality in clinical workflows. While optimized volume rendering techniques [12] can improve clinical decision-making, they still struggle on resource-constrained hardware. Our work bridges the gap by reformulating Gaussian Splatting as a direct volumetric optimization problem, enabling efficient visualization without intermediate rendering.

3 Method

3.1 From Image-Space to Volume-Space

We reformulate 3D Gaussian Splatting as a direct volumetric optimization problem. Instead of learning a scene from multi-view RGB images, we optimize a set of anisotropic 3D Gaussians directly against a target volume V and a region-of-interest (ROI) mask M. Supervision, initialization, and topology adaptation are all performed in voxel space, naturally aligning the representation with medical imaging. This eliminates the need for camera models, feature extraction, feature matching, and image-based supervision.

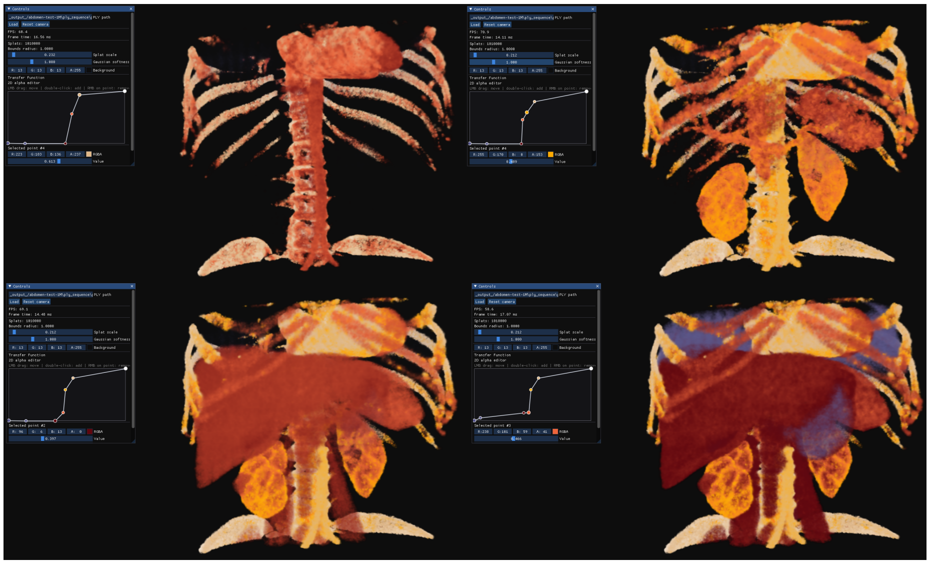

Fig. 1. Our novel viewer application exploits the new per-splat intensity values (effectively storing HU values) to dynamically change the transfer function during rendering, similar to a volume rendering.

3.2 Gaussian Representation

Similar to the original 3DGS approach [10], we parameterize each Gaussian i by its center position μ_i, its covariance matrix Σ_i and opacity α_i. In addition to that, each splat explicitly encodes a physical intensity value c_i (e.g., Hounsfield units in CT), sampled directly from the input volume at each Gaussian position using trilinear interpolation in normalized volume space. For large splats, the method can instead use the mean of the voxels covered by the splat footprint.

This allows transfer functions to be applied or modified after training (see Fig 1), similar to classical volume rendering, without requiring re-optimization. In contrast to the original method, we restrict the spherical harmonic representation to degree 0, resulting in view-independent intensities that better reflect the physically meaningful and direction-independent nature of medical imaging data. For compatibility with standard Gaussian Splatting tools, exported PLY files include a grayscale SH color parameter derived from the intensity values via a standard black-and-white transfer function.

3.3 Differentiable Splat-to-Volume Formulation

To enable direct supervision in voxel space, we rasterize the Gaussian representation into a dense voxel grid that is aligned with the target volume. This allows a direct comparison between the reconstructed volume $\hat{V}$ and the ground-truth volume V.

Each Gaussian i, defined by its center μ_i, covariance Σ_i, opacity α_i, and intensity c_i, contributes locally to nearby voxels via a spatial weighting function w_i. Specifically, the contribution of Gaussian i to a voxel at position $\mathbf{x}$ is given by

$$w_i(\mathbf{x}) = \alpha_i \, \mathcal{N}(\mathbf{x} \mid \mu_i, \Sigma_i), \tag{1}$$

where $\mathcal{N}(\mathbf{x} \mid \mu_i, \Sigma_i)$ denotes the anisotropic Gaussian kernel evaluated at $\mathbf{x}$. Intuitively, $w_i(\mathbf{x})$ measures how strongly Gaussian i influences voxel $\mathbf{x}$, modulated by its opacity.

To reconstruct the voxel intensity, we aggregate contributions from all Gaussians using a normalized weighted average:

$$\hat{V}(\mathbf{x}) = \frac{\sum_i w_i(\mathbf{x}) \, c_i}{\sum_i w_i(\mathbf{x}) + \varepsilon}, \tag{2}$$

where c_i is the intensity associated with Gaussian i, and ε is a small constant to ensure numerical stability. This formulation can be interpreted as a smooth interpolation of Gaussian intensities in voxel space.

A naive evaluation of all Gaussians for every voxel would be computationally prohibitive. Therefore, we restrict computation to local neighborhoods by evaluating each Gaussian only within a finite spatial support derived from its covariance.

3.4 Smart Initialization

For most clinical tasks, e.g. intervention planning, only specific anatomical structures are relevant. To focus model capacity and computation on these regions, we assume the availability of a region-of-interest (ROI) mask. The mask can be obtained from segmentation methods such as TotalSegmentator and may be binary or soft (float). Gaussian centers are sampled within the ROI, where soft masks are interpreted as sampling probabilities, allowing uncertain but potentially relevant regions to be retained.

In addition, we incorporate simple structural priors derived from the input volume. Local gradients or second-order cues (e.g., structure tensor or Hessian-based estimates) can be used to infer preferred orientations, enabling anisotropic Gaussians to align with elongated structures such as vessels or bones. This improves convergence and reduces the need for extensive topology adaptation during optimization.

The initial Gaussian scales are sampled uniformly within a predefined range in voxel units and converted to world space using the mean voxel spacing. This ensures scale consistency across datasets with different resolutions while providing a diverse set of initial splat sizes. Due to this volume-informed initialization strategy, already very early training iterations yield good results (see Fig. 2).

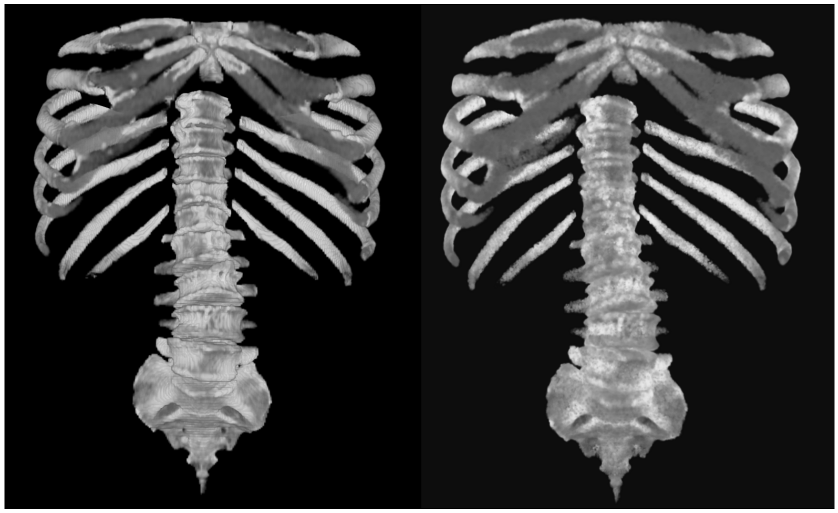

Fig. 2. "Case 4" dataset with the TotalSegmentator bones mask. Left: Volume rendering (reference). Right: Our method (500k splats, optimization iteration 100). Despite using only very few iterations of optimization, the Gaussian representation already closely approximates the reference, due to the smart initialization strategy.

3.5 ROI-Aware Optimization

Optimization is performed directly in voxel space using losses restricted to the ROI. This prevents large empty background regions from dominating optimization and improves robustness for sparse anatomical structures.

The framework supports both structure-focused optimization (e.g., mask fitting) and intensity-based optimization, as well as their combination. By default, the loss for the structure-focused optimization is Dice and the standard loss for the intensity / CT data values is mean squared error (MSE). An additional penalty discourages density leakage outside the ROI, which stabilizes training.

$$\mathcal{L} = \lambda_{\mathrm{mask}}\mathcal{L}_{\mathrm{mask}} + \lambda_{\mathrm{ct}}\mathcal{L}_{\mathrm{ct}} + \lambda_{\mathrm{leak}}\mathcal{L}_{\mathrm{leak}}. \tag{3}$$

3.6 Adaptive Topology Optimization

To maintain an efficient representation, the Gaussian set is dynamically adapted during optimization. Densification and pruning are performed periodically during a predefined training window. At each densification step, Gaussians with large accumulated position gradients (measured via the norm of voxel-space position updates) are refined: small splats are cloned, larger splats are split, and low-opacity splats are removed. The selection threshold is set adaptively from the current gradient distribution, and an additional hole-filling step can spawn new splats in under-covered regions (e.g., regions with low accumulated weights in the voxel grid). Unlike standard Gaussian Splatting, these topology updates are driven entirely by voxel-space optimization signals rather than by view-dependent visibility.

This enables effective modeling of thin and complex anatomical structures while keeping the representation compact.

3.7 Implementation Details

Our implementation extends the original 3DGS pipeline [10] with a dedicated volumetric training path. Data loading, initialization, supervision, rasterization, optimization, and export operate directly on NIfTI volumes and ROI masks in a single unified workflow. To handle large medical volumes, we employ ROI-restricted computation, optional working-grid downscaling, chunked Gaussian accumulation, and mixed-precision storage. Task-specific presets (e.g., for vessels) and intermediate PLY checkpoints support downstream rendering.

By default, optimization uses Adam with learning rates of 1×10^{-4} for positions, 5×10^{-3} for scales, 2.5×10^{-2} for learnable opacities, and 2.5×10^{-3} for learnable intensity/color coefficients. The position learning rate is exponentially decayed to 1.6×10^{-6} over 20,000 iterations; other learning rates remain fixed. The splat-to-volume normalization uses $\varepsilon = 10^{-6}$. The default loss combines Dice mask supervision and CT MSE with $\lambda_{\text{mask}} = 1.0$, $\lambda_{\text{ct}} = 1.0$, and an outside-mask leak penalty $\lambda_{\text{leak}} = 0.1$. Densification runs every 100 iterations from iteration 100 to 2000, using an adaptive threshold based on the 60th percentile of accumulated per-splat position gradients; splats with opacity below 10^{-4} are pruned. Initial Gaussian scales are sampled from 1.5–4.0 voxels and clamped to 1.5–10.0 voxels. Initial orientations are derived from intensity-volume gradients smoothed with $\sigma = 0.8$ voxels; Hessian-based structure cues are available but disabled by default. All of these hyperparameters are exposed as CLI parameters and documented in the repository.

Since standard Gaussian Splatting viewers do not support per-splat intensity values, we developed a novel viewer application enabling dynamic transfer function changes during rendering, analogous to classical volume rendering workflows (see Fig. 1). The implementation is publicly available in our GitHub repository.

4 Experiments

4.1 Experimental Setup

We evaluate our method on medical volumetric datasets, focusing on anatomically relevant structures such as vessels and organs. The input data consists of CT images together with corresponding segmentation masks.

Table 1. Datasets from the AbdomenCT-1k [15] project used in the evaluation.

Name	Dimensions	Voxel size [mm]
Case 1	$292 \times 292 \times 180$	$1.5 \times 1.5 \times 1.5$ (resampled)
Case 4	$512 \times 512 \times 157$	$0.742 \times 0.742 \times 3.0$
Case 6	$512 \times 512 \times 227$	$0.816 \times 0.816 \times 3.0$

We used several datasets from the AbdomenCT-1k project [15] (see Table 1). Segmentations were obtained using TotalSegmentator v2 [21] and saved as binary masks. In addition, a soft (float) liver segmentation mask was created using an internal segmentation model.

All experiments are conducted on a Windows 11 system with an Intel i7 Ultra 155H, 32GB RAM, and an NVIDIA RTX 4060 Mobile GPU (8GB VRAM).

We compare against two baselines:

- **Volume Rendering (VR):** Standard GPU-based volume rendering using the GigaVoxel renderer in MeVisLab 4.2.70 [1]. This serves as a reference for visual fidelity.
- **3D Gaussian Splatting (3DGS):** Standard image-based 3DGS as implemented in Nerfstudio [18]. Training is performed on 500 rendered images with a resolution of 1024×1024 and from random spherical camera positions. The renderings were generated with the same GigaVoxel renderer to ensure consistency of input data.

4.2 Evaluation Protocol

Our method and standard 3DGS differ fundamentally in supervision regime, input modality, coordinate system, and optimization objective—making direct quantitative comparison ill-defined. We therefore adopt a two-part protocol: (i) quantitative evaluation in the volume domain, and (ii) qualitative comparison in image space.

Quantitative evaluation: Our primary metric is the masked mean squared error (MMSE) in the native volume domain. After training, the Gaussian representation is rasterized onto the original voxel grid and compared to the ground-truth volume within the ROI. Formally, let V denote the normalized ground-truth volume, $\hat{V}$ the reconstructed volume, and $M \in \{0, 1\}$ the ROI mask:

$$\mathcal{L}_{\text{MMSE}} = \frac{1}{\sum_i M_i} \sum_i M_i \, (\hat{V}_i - V_i)^2. \tag{4}$$

All Gaussians are used during evaluation, irrespective of subset-based updates during training. MMSE values for standard 3DGS cannot be computed meaningfully without an additional registration step, as image-space optimization does not produce models aligned with the input voxel grid.

Qualitative evaluation: We visually compare our method against direct volume rendering and standard 3DGS (Fig. 3). Since the methods operate in different coordinate systems, renderings are generated from approximately matched viewpoints for visual inspection of structural fidelity and anatomical plausibility.

5 Results

5.1 Quantitative Results

Table 2 reports MMSE and PSNR values for our method across datasets, splat counts, and mask types. For a simple baseline comparison, we also report MMSE and mask-related PSNR values for downsampled versions of the input dataset (see Table 3).

Table 2. Best and worst MMSE and mask-based PSNR for our method within the first 1000 iterations. The Gaussian model is rasterized onto the native voxel grid and evaluated against the ground truth input volume.

Dataset	Mask	Splat Count	MMSE ↓		PSNR ↑ [dB]	
			Worst	Best	Worst	Best
Case 1	TS (binary)	100k	0.00336	0.00242	24.92	26.16
Case 1	TS (binary)	250k	0.00276	0.00235	25.58	26.29
Case 1	TS (binary)	500k	0.00273	0.00213	25.65	26.71
Case 1	TS (binary)	1000k	0.00241	0.00192	26.17	27.18
Case 1	Liver (float)	250k	0.00066	0.00059	31.80	32.27
Case 1	Liver (float)	500k	0.00066	0.00056	31.81	32.53
Case 4	TS-Bones (binary)	100k	0.008410	0.004617	20.75	23.36
Case 4	TS-Bones (binary)	250k	0.007950	0.004173	21.00	23.80
Case 4	TS-Bones (binary)	500k	0.007106	0.003935	21.48	24.05
Case 4	TS-Bones (binary)	1000k	0.005787	0.003633	22.38	24.40
Case 6	TS (binary)	250k	0.001172	0.000766	29.31	31.16
Case 6	TS (binary)	500k	0.001043	0.00068	29.82	31.67
Case 6	TS (binary)	1000k	0.000836	0.000614	30.78	32.12

5.2 Qualitative Results

Figure 3 shows representative renderings from approximately matched viewpoints comparing our method against volume rendering and standard 3DGS.

5.3 Efficiency Analysis

Rendering performance: Table 5 reports frames per second (FPS) at 1024×1024 render resolution. Volume rendering was measured using the GigaVoxelRenderer in MeVisLab; Gaussian methods used the SIBR Gaussian Viewer App.

Table 3. Baseline comparison: MMSE and Mask-based PSNR for a downsampled version of the dataset, evaluated against its full-sized version.

Dataset	Mask	Downsample factor	MMSE ↓	PSNR ↑ [dB]
Case 1	TS (binary)	2	0.001680	27.75
Case 1	TS (binary)	4	0.047450	13.24
Case 4	TS-Bones (binary)	2	0.000925	30.34
Case 4	TS-Bones (binary)	4	0.002945	25.31
Case 6	TS (binary)	2	0.000150	38.25
Case 6	TS (binary)	4	0.000557	32.54

Table 4. Mini-Ablation: PSNR values in [dB] for different loss terms in the first 1000 iterations. In all cases, the leak loss $\mathcal{L}_{\text{leak}}$ was fixed to its default weight: $\lambda_{\text{leak}} = 0.1$.

Dataset	Splat Count	$\lambda_{\text{mask}} = \lambda_{\text{ct}} = 1$		$\lambda_{\text{mask}} = 0, \lambda_{\text{ct}} = 1$		$\lambda_{\text{mask}} = 1, \lambda_{\text{ct}} = 0$	
		Worst	Best	Worst	Best	Worst	Best
Case 6	500k	29.82	31.67	21.26	31.77	28.56	31.64

Table 5. Rendering performance of the different methods at 1024×1024 image resolution using the SIBR viewer application [10].

Method	Dataset	Mask	Splat count	Framerate [FPS]
Volume Rendering	Case 1	TS (binary)	-	107
3DGS	Case 1	TS (binary)	250k	272
Ours	Case 1	TS (binary)	250k	337
Ours	Case 1	TS (binary)	500k	236
Ours	Case 1	TS (binary)	1000k	147
Volume Rendering	Case 4	TS-bones (binary)	-	125
3DGS	Case 4	TS-bones (binary)	250k	281
Ours	Case 4	TS-bones (binary)	250k	341
Ours	Case 4	TS-bones (binary)	500k	238
Ours	Case 4	TS-bones (binary)	1000k	152
Volume Rendering	Case 6	TS (binary)	-	104
3DGS	Case 6	TS (binary)	250k	265
Ours	Case 6	TS (binary)	250k	342
Ours	Case 6	TS (binary)	500k	239
Ours	Case 6	TS (binary)	1000k	143

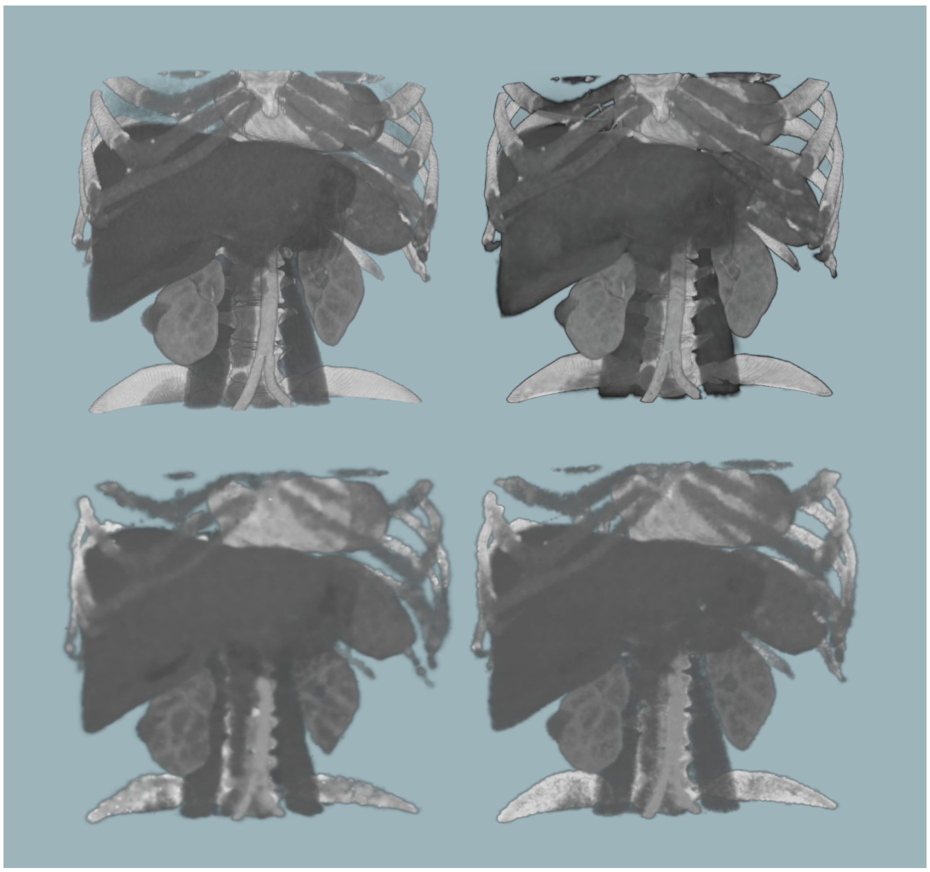

Fig. 3. The "Case 1" dataset with TotalSegmentator binary masks rendered using different modalities. Top left: Volume Rendering (reference), top right: Standard 3DGS (250k splats; 32 mins train time), bottom left: our method (250k splats; 39 mins train time), bottom right: our method (1 million splats; 76 mins train time).

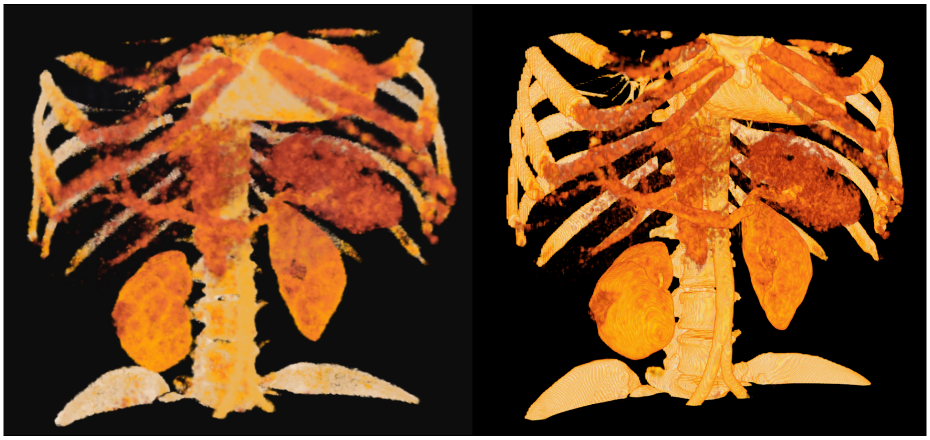

Fig. 4. The "Case 1" dataset rendered with our method using 1 million splats (left) and rendered with volume rendering (right). The transfer functions were matched manually.

Training performance: Table 6 reports optimization timings for 1000 iterations. Training time depends on splat count, subset size, voxel resolution, and ROI extent; values should be interpreted as indicative.

6 Discussion

By optimizing directly in voxel space against a target volume and ROI mask, our method avoids multi-view rendering, structure-from-motion preprocessing, and camera-based supervision. Supervision is grounded in the native data representation, eliminating error sources from pose estimation, projection geometry, and view-dependent artifacts. This is reflected in our quantitative results: low MMSE values indicate good reconstruction fidelity within the ROI (Table 2), and strong PSNR is achieved within the first few hundred iterations (Table 6), demonstrating rapid convergence from the smart initialization. The PSNR values achieved by our method are in comparable ranges to a low downsampling of the image (see Tables 2 and 3). Qualitatively, reconstructed volumes preserve overall anatomical structure well; however, our method appears a bit blurrier (see Fig. 4). Standard 3DGS often appears sharper but can exhibit intensity inaccuracies, particularly in semi-transparent regions (see Fig. 3).

A mini-ablation on loss term contributions (Table 4) confirms that both the mask supervision term (λ_{mask}) and the CT intensity term (λ_{ct}) contribute to reconstruction quality: omitting either term reduces worst-case PSNR by up to 8 dB, while their combination yields the most stable performance across iterations.

A distinctive advantage of our formulation is the per-splat intensity encoding (e.g., Hounsfield units). This enables interactive post-hoc transfer function adjustment—similar to classical volume rendering—without retraining, in contrast

Table 6. Training performance of our method for the first 1000 optimization iterations.

Dataset	Mask	Optimized subset size	Splat count		Time [min]	PSNR [dB]	VRAM [GB]
			Start	End			
Case 1	TS (binary)	2k	100k	100k	23	24.77	2.16
Case 1	TS (binary)	4k	250k	250k	39	25.61	3.27
Case 1	TS (binary)	8k	200k	375k	77	25.62	5.37
Case 1	TS (binary)	8k	500k	518k	77	25.65	5.46
Case 1	TS (binary)	8k	1000k	1036k	76	26.17	5.66
Case 1	Liver (float)	8k	250k	250k	85	31.86	5.42
Case 1	Liver (float)	8k	500k	500k	95	31.81	5.50
Case 4	TS-bones (binary)	4k	250k	250k	54	21.00	5.02
Case 4	TS-bones (binary)	4k	500k	500k	55	21.48	5.11
Case 4	TS-bones (binary)	4k	1000k	1000k	57	22.38	5.19
Case 6	TS (binary)	4k	250k	250k	67	29.31	6.27
Case 6	TS (binary)	4k	500k	500k	67	29.82	6.36
Case 6	TS (binary)	4k	1000k	1000k	92	30.78	6.55

to standard 3DGS where appearance is largely fixed after optimization. Regarding efficiency, our method matches standard 3DGS rendering performance while significantly outperforming classical ray-marched volume rendering, making it well suited for resource-constrained environments such as mixed reality devices.

Current limitations include memory and compute demands of volumetric rasterization at large splat counts. The implementation addresses this through sparse mini-batch optimization, where only a subset of splats is actively updated per iteration, reducing peak GPU memory consumption at the cost of slower effective convergence per splat. On our test hardware (8 GB VRAM), training with up to 1 million splats remained feasible across all test configurations when using mini-batch optimization with maximum 8k subsets (see Table 6). Further challenges include sensitivity to initialization quality and mask accuracy, as well as a tendency toward uniform splat distributions that can limit fidelity in fine-structured regions such as thin vessels. Claims about suitability for mixed reality devices are currently supported by efficiency measurements on a mobile GPU; dedicated MR hardware evaluation and end-to-end latency benchmarks are left for future work.

7 Conclusion

We presented Volumetric Gaussian Splatting, a reformulation of 3D Gaussian Splatting that operates directly in voxel space. By replacing image-based supervision with voxel-space optimization against a target volume and ROI mask, the approach eliminates camera models, pose estimation, and multi-view reconstruction—producing a representation inherently aligned with medical imaging data.

The intensity-aware Gaussian representation, where each splat encodes a physically meaningful value (e.g., Hounsfield units), enables flexible post-hoc transfer function design without retraining. This combines the efficiency of Gaussian-based rendering with the controllability of classical volume rendering, achieving real-time frame rates on resource-constrained hardware.

Our experiments demonstrate rapid convergence from smart initialization, good reconstruction fidelity within the ROI, and rendering performance substantially exceeding classical volume rendering. Limitations include increased training memory demands, sensitivity to ROI mask quality, and room for improvement in adaptive densification.

Future work could target improved reconstruction accuracy, further training efficiency, more extensive evaluation on MR hardware and physically-based extensions such as ambient occlusion to enable more advanced rendering effects. We believe volume-centric Gaussian representations open new directions beyond traditional image-based Gaussian Splatting.

Acknowledgments. This study was partly funded by the Federal Ministry of Research, Technology and Space (BMFTR) within the AHrEZ project (grant number 16SV9075).

References

1. Mevislab website, https://www.mevislab.de/, last accessed 2026/04/17
2. Bui, N.T., Hoang, D.H., Tran, M.T., Doretto, G., Adjeroh, D., Patel, B., Choudhary, A., Le, N.: Sam3d: Segment anything model in volumetric medical images. arXiv:2309.03493 (2023)
3. Celarek, A., Kopanas, G., Drettakis, G., Wimmer, M., Kerbl, B.: Does 3d gaussian splatting need accurate volumetric rendering? Computer Graphics Forum **44**(2), e70032 (2025). https://doi.org/10.1111/cgf.70032
4. Corona-Figueroa, A., Frawley, J., Bond-Taylor, S., Bethapudi, S., Shum, H.P.H., Willcocks, C.G.: Mednerf: Medical neural radiance fields for reconstructing 3d-aware ct-projections from a single x-ray (2022)
5. Drebin, R.A., Carpenter, L., Hanrahan, P.: Volume rendering. In: Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques. p. 65–74. SIGGRAPH '88, Association for Computing Machinery, New York, NY, USA (1988). https://doi.org/10.1145/54852.378484
6. Dyken, L., Sewell, A., Usher, W., Debardeleben, N., Petruzza, S., Kumar, S.: Volume encoding gaussians: Transfer function-agnostic 3d gaussians for volume rendering (2026), https://arxiv.org/abs/2504.13339
7. Gao, J., Gu, C., Lin, Y., Li, Z., Zhu, H., Cao, X., Zhang, L., Yao, Y.: Relightable 3d gaussians: Realistic point cloud relighting with brdf decomposition and ray tracing (2024), https://arxiv.org/abs/2311.16043
8. Gao, Z., Planche, B., Zheng, M., Chen, X., Chen, T., Wu, Z.: Ddgs-ct: Direction-disentangled gaussian splatting for realistic volume rendering. In: Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., Zhang, C. (eds.) Advances in Neural Information Processing Systems. vol. 37, pp. 39281–39302. Curran Associates, Inc. (2024). https://doi.org/10.52202/079017-1240

9. Gao, Z., Planche, B., Zheng, M., Choudhuri, A., Chen, T., Wu, Z.: 6dgs: Enhanced direction-aware gaussian splatting for volumetric rendering (2025), https://arxiv.org/abs/2410.04974
10. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (July 2023), https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/
11. Kleinbeck, C., Schieber, H., Engel, K., Gutjahr, R., Roth, D.: Multi-layer gaussian splatting for immersive anatomy visualization. IEEE Transactions on Visualization and Computer Graphics **31**(5), 2353–2363 (2025). https://doi.org/10.1109/TVCG.2025.3549882
12. Kraft, V., Schumann, C., Salzmann, D., Weyhe, D., Zachmann, G., Schenk, A.: A clinical user study investigating the benefits of adaptive volumetric illumination sampling. IEEE Transactions on Visualization and Computer Graphics **30**(12), 7571–7578 (2024). https://doi.org/10.1109/TVCG.2024.3353926
13. Kroes, T., Post, F.H., Botha, C.P.: Exposure render: An interactive photo-realistic volume rendering framework. PLOS ONE **7**(7), 1–10 (07 2012). https://doi.org/10.1371/journal.pone.0038586, https://doi.org/10.1371/journal.pone.0038586
14. Li, D., Jia, D., Rajeh, Y., Engel, D., Viola, I.: Rara clipper: A clipper for gaussian splatting based on ray tracer and rasterizer. In: Proceedings of the SIGGRAPH Asia 2025 Conference Papers. SA Conference Papers '25, Association for Computing Machinery, New York, NY, USA (2025). https://doi.org/10.1145/3757377.3763982
15. Ma, J., Zhang, Y., Gu, S., Zhu, C., Ge, C., Zhang, Y., An, X., Wang, C., Wang, Q., Liu, X., et al.: Abdomenct-1k: Is abdominal organ segmentation a solved problem? IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(10), 6695–6714 (2021)
16. Magalhães, R., Oliveira, A., Terroso, D., Vilaça, A., Veloso, R., Marques, A., Pereira, J., Coelho, L.: Mixed reality in the operating room: A systematic review. Journal of medical systems **48**(1), 76 (August 2024). https://doi.org/10.1007/s10916-024-02095-7
17. Niedermayr, S., Neuhauser, C., Petkov, K., Engel, K., Westermann, R.: Application of 3d gaussian splatting for cinematic anatomy on consumer class devices (2024), https://arxiv.org/abs/2404.11285
18. Tancik, M., Weber, E., Ng, E., Li, R., Yi, B., Kerr, J., Wang, T., Kristoffersen, A., Austin, J., Salahi, K., Ahuja, A., McAllister, D., Kanazawa, A.: Nerfstudio: A modular framework for neural radiance field development. In: ACM SIGGRAPH 2023 Conference Proceedings. SIGGRAPH '23 (2023)
19. Tang, K., Yao, S., Wang, C.: ivr-gs: Inverse volume rendering for explorable visualization via editable 3d gaussian splatting (2025), https://arxiv.org/abs/2504.17954
20. Vervoorn, M., Wulfse, M., Doormaal, T., Ruurda, J., van der Kaaij, N., Heer, L.: Mixed reality in modern surgical and interventional practice: a narrative review of the literature (preprint). JMIR Serious Games **11** (07 2022). https://doi.org/10.2196/41297
21. Wasserthal, J., Breit, H.C., Meyer, M.T., Pradella, M., Hinck, D., Sauter, A.W., Heye, T., Boll, D.T., Cyriac, J., Yang, S., Bach, M., Segeroth, M.: Totalsegmentator: Robust segmentation of 104 anatomic structures in ct images. Radiology: Artificial Intelligence **5**(5) (Sep 2023). https://doi.org/10.1148/ryai.230024
22. Zwicker, M., Pfister, H., van Baar, J., Gross, M.: Ewa volume splatting. In: Proceedings of the Conference on Visualization '01. p. 29–36. VIS '01, IEEE Computer Society, USA (2001)