Render-FM: Feedforward Model for Real-time Photorealistic Volumetric Rendering

Zhongpai Gao*, Benjamin Planche, Meng Zheng, Anwesa Choudhuri, Van Nguyen Nguyen, Terrence Chen, and Ziyan Wu

United Imaging Intelligence, Boston, MA, USA
{zhongpai.gao, benjamin.planche, meng.zheng, anwesa.choudhuri, vannguyen.nguyen, terrence.chen, ziyan.wu}@uii-ai.com

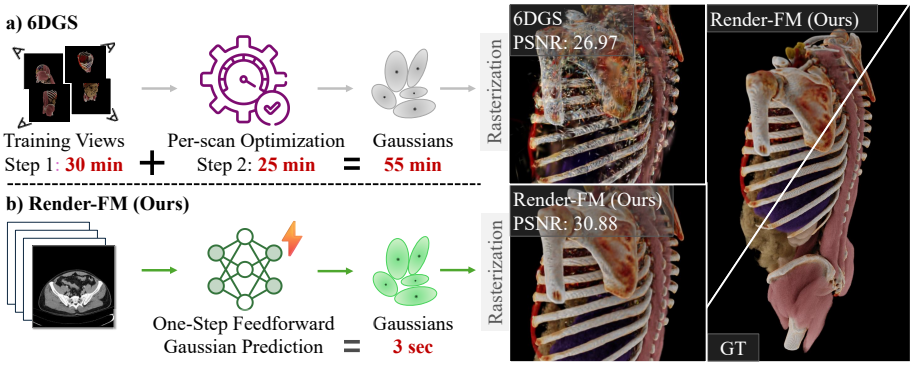

Fig. 1: Comparison of CT scan volumetric rendering pipelines. **Top:** 6DGS [12] relies on physically-based rendering (PBR) of training views followed by per-scan optimization, and exhibits severe artifacts under limited training views. **Bottom:** Our Render-FM produces high-quality renderings via a single feedforward pass.

Abstract. Photorealistic volumetric rendering of CT scans greatly benefits clinical workflows, yet neural approaches such as Neural Radiance Fields (NeRF) and 3D Gaussian Splatting (3DGS) require prohibitive per-scan optimization (hours for NeRF, about 30 minutes for 3DGS), making them impractical in clinical settings. We propose Render-FM, a feedforward model that eliminates this bottleneck by directly regressing 6D Gaussian Splatting (6DGS) parameters from a CT volume in a single 2.8-second forward pass, a 500× speedup over per-scan optimization. To bridge the domain gap between natural scene reconstruction and medical volumetric rendering, we introduce Anatomy-Guided Priming (AGP), which incorporates segmentation masks and transfer functions as structural and appearance priors, information that existing Gaussian splatting methods overlook. Built on an nnU-Net-inspired 3D U-Net trained on diverse CT scans, Render-FM predicts per-voxel 6DGS parameters and supports immediate real-time rendering. Unlike per-scan methods, it generalizes to unseen anatomies, novel transfer

* Corresponding author

functions, and enables compositional organ visualization with zero additional preparation time. Optional 89-second fine-tuning further improves quality, surpassing per-scan optimized baselines. Project page: `https://gaozhongpai.github.io/renderfm/`.

Keywords: Gaussian splatting · Feedforward model · Volumetric rendering · Computed tomography

1 Introduction

Medical imaging modalities like Computed Tomography (CT) produce rich volumetric datasets that are conventionally reviewed as sequences of 2D slices. While fundamental to clinical practice, this slice-by-slice approach often fails to convey the intricate spatial relationships of 3D anatomical structures and pathologies, particularly in complex cases involving multi-organ interactions or vascular networks [6]. Volumetric rendering addresses this limitation by synthesizing comprehensive 3D views that enable intuitive and interactive exploration, significantly enhancing diagnostic assessment, surgical planning, and patient communication [1, 7]. The ability to dynamically inspect patient anatomy from arbitrary perspectives fundamentally transforms how clinicians interact with medical imaging data [4].

Despite technological advancements, achieving photorealistic volumetric rendering faces significant barriers to routine clinical adoption. While conventional Direct Volume Rendering (DVR) methods [17] enable real-time interaction through GPU-accelerated ray-casting, they employ simplified local illumination models that produce visually limited results. In contrast, photorealistic rendering techniques like Cinematic Rendering [6, 7, 26, 31] and physically-based path tracing achieve remarkable visual quality through global illumination, soft shadows, and subsurface scattering, effects critical for realistic tissue appearance and enhanced spatial perception in surgical planning and patient communication. However, these methods require computationally expensive light transport simulation. More recent approaches, such as Neural Radiance Fields (NeRF) [20] and 3D Gaussian Splatting (3DGS) [18], can achieve photorealistic quality but impose prohibitive per-scan optimization requirements that fundamentally conflict with clinical time constraints.

The core bottleneck lies in per-scene optimization: 3DGS typically requires around 30 minutes, while NeRF can extend to 10+ hours. This is compounded by the prerequisite generation of physically-rendered training views, consuming approximately 18 seconds per view, bringing the full preparation pipeline to nearly an hour per case for 3DGS, and far longer for NeRF. Beyond speed, optimized parameters cannot transfer between scans due to variations in anatomy, pathology, and acquisition protocols, nor can they generalize to new transfer functions (*i.e.*, color and opacity definitions) or support compositional organ visualization, capabilities essential for comprehensive diagnostic workflows.

Recent advances in large-scale feedforward models present a transformative paradigm for addressing these limitations. Inspired by foundation model

pre-training principles [2, 21, 22], Large Gaussian Models (LGMs) [27, 32] have demonstrated that feedforward networks trained on large datasets can directly predict 3D Gaussian parameters from sparse image inputs, bypassing per-scene optimization entirely. While LGMs target natural scene reconstruction from 2D views, their core insight transfers naturally to medical volumetric rendering: CT scans provide dense, structured 3D input that makes the feedforward mapping more tractable, and large-scale CT datasets spanning diverse institutions and pathologies offer the breadth needed to learn generalizable anatomical priors across patients.

We introduce Render-FM, a feedforward model for photorealistic volumetric rendering through direct prediction of 6D Gaussian Splatting (6DGS) parameters from CT volumes. Unlike optimization-based methods requiring extensive per-scan training, Render-FM performs parameter regression in a single 2.8-second forward pass, a 500× speedup, while maintaining superior visual quality. We adopt 6DGS [12] over standard 3DGS for its explicit modeling of complex view-dependent optical effects in a 6D spatio-angular space, better capturing scattering and reflectance phenomena at tissue interfaces that are critical for photorealistic medical visualization. The model employs an encoder-decoder architecture inspired by nnU-Net's medical imaging principles [14, 16], combined with our novel Anatomy-Guided Priming (AGP) that incorporates anatomical segmentation masks and transfer functions as structured priors, contextual information that existing Gaussian Splatting methods entirely overlook.

Trained on diverse CT datasets spanning multiple institutions and pathologies [30], Render-FM learns robust generalization across clinical imaging conditions. The resulting model enables immediate real-time rendering at 328+ FPS while supporting dynamic transfer function modification and compositional organ visualization, capabilities impossible with optimization-based approaches. Optional fine-tuning can further enhance quality to 31.67 dB PSNR within 89 seconds, providing flexibility while maintaining practical deployment timelines. Our key contributions include:

- **Feedforward Model Architecture:** A novel feedforward model integrating nnU-Net's medical imaging principles with 6DGS's view-dependent rendering capabilities to directly regress 6DGS parameters from CT volumes, eliminating per-scan optimization and reducing preparation time from ∼1 hour to 2.8 seconds.
- **Anatomy-Guided Priming (AGP):** A novel initialization strategy that leverages segmentation masks and transfer functions to provide anatomically-informed structural and appearance priors for 6D Gaussian primitives, bridging the domain gap between natural scene reconstruction and medical volumetric rendering.
- **End-to-End Training & Evaluation:** A comprehensive training methodology using differentiable 6DGS rendering on large-scale CT datasets, with extensive evaluation demonstrating superior rendering quality, 500× speedup over per-scan optimization, and robust generalizability to unseen anatomies, novel transfer functions, and compositional organ visualization.

2 Related Work

Volumetric Rendering in Medical Imaging Direct Volume Rendering (DVR) synthesizes images by casting rays through volumetric data using transfer functions to map voxel intensities to optical properties [13, 17]. Standard GPU-accelerated DVR implementations enable real-time interaction in clinical viewers like 3D Slicer and OsiriX [1], but employ simplified local illumination (*e.g.*, Phong shading) producing visually limited results. Cinematic Rendering [6–8, 26] achieves photorealism through physically-based rendering with global illumination, soft shadows, and subsurface scattering. However, such photorealistic methods require expensive light transport simulation (seconds per view) and expert transfer function tuning, limiting routine clinical adoption despite their superior visual quality.

Neural Radiance Fields (NeRF) and 3D Gaussian Splatting (3DGS) NeRF [20] implicitly represents scenes using MLPs that map 5D position-and-direction coordinates to density and color, achieving high-quality novel view synthesis at the cost of extensive per-scene optimization (hours to days) and slow rendering. 3DGS [18] addresses rendering speed by explicitly modeling scenes with 3D Gaussian primitives and a differentiable tile-based rasterizer, enabling real-time rendering, but still requires around 30 minutes of per-scene optimization. Both have been adapted for medical imaging, including sparse-view CT reconstruction [3] and X-ray visualization [11]. However, these adaptations retain the per-scan optimization requirement, and optimized parameters cannot transfer across scans or generalize to new transfer functions, which are fundamental limitations for clinical deployment.

6D Gaussian Splatting 6D Gaussian Splatting (6DGS) [12] extends 3DGS by representing primitives within a 6D spatio-angular space, characterized by a 6D covariance matrix that captures variance in both 3D position and 3D direction. This formulation enables explicit modeling of complex view-dependent effects, such as anisotropic reflections and subsurface scattering, that are frequently observed at tissue interfaces in medical data and are difficult to capture with standard 3DGS spherical harmonics. During rendering, the 6D covariance is dynamically sliced based on the viewing direction to yield an effective 3D Gaussian for rasterization, achieving higher fidelity with potentially fewer primitives [12]. These properties make 6DGS particularly well-suited for photorealistic medical visualization.

Feedforward Models and Large Gaussian Models Foundation models pre-trained on large-scale datasets demonstrate robust generalization to downstream tasks without task-specific optimization [2,21,22]. Large Gaussian Models (LGMs) [27, 32] bring this paradigm to 3D reconstruction, training feedforward networks to directly predict 3D Gaussian parameters from sparse 2D image inputs without per-scene optimization. However, LGMs are designed for natural scene reconstruction from RGB images, a setting fundamentally different from medical volumetric rendering, which involves structured 3D CT input, domain-specific appearance priors (transfer functions), and the need for anatomical generalization across patients. Render-FM adapts the feedforward paradigm to this

medical setting by leveraging dense 3D CT volumes as input and incorporating anatomy-guided priors, addressing the gaps that make LGMs ill-suited for direct clinical application.

nnU-Net Framework The nnU-Net framework [14,16] excels in medical image segmentation through self-configuring pipeline adaptation, automatically adjusting patch size, normalization, and architecture to handle the wide variation in CT resolutions, spacings, and fields-of-view across institutions. Its 3D U-Net backbone consistently achieves state-of-the-art results across diverse benchmarks, and its principles have been extended beyond segmentation to tasks such as landmark detection [9] and interactive segmentation [15]. Render-FM leverages these same architectural principles for CT-to-6DGS parameter regression, benefiting from nnU-Net's robustness to clinical CT variability.

To our knowledge, Render-FM is the first method to combine feedforward prediction, 6D Gaussian Splatting, and anatomy-guided priming for photorealistic CT volumetric rendering, eliminating per-scan optimization while achieving rendering quality comparable to or better than per-scan optimized baselines.

3 Methodology

Render-FM learns a direct mapping from a CT volume to 6D Gaussian Splatting (6DGS) parameters via a single feedforward pass, enabling real-time photorealistic rendering without per-scan optimization. An overview of the pipeline is shown in Fig. 2.

Problem Formulation Given a 3D CT scan $V \in \mathbb{R}^{C\times D\times H\times W}$ with $C = 6$ input channels, our goal is to predict parameters Θ defining a 6DGS representation that enables high-quality, view-dependent rendering from arbitrary viewpoints. Formally, we learn a function $f_\phi : V \mapsto \Psi$, where $\Psi \in \mathbb{R}^{37\times\lfloor D/2\rfloor\times\lfloor H/2\rfloor\times\lfloor W/2\rfloor}$ represents voxel-wise 6DGS parameters across the volume, parameterized by network weights ϕ. During rendering, a foreground subset $\Theta \subset \Psi$ determined by the segmentation mask is instantiated as 6D Gaussian primitives.

3.1 Model Architecture

Input Representation The 6-channel input V provides complementary anatomical and appearance information: 1) normalized CT intensity (Hounsfield Units), capturing tissue density; 2) a segmentation mask identifying foreground structures; and 3) four RGBA channels from pre-defined transfer functions, encoding base color and opacity per anatomy class. This multi-channel design enriches the network with structural and appearance context, including tissue density, semantic identity, and base appearance, reducing the burden of learning these priors from scratch and enabling more accurate Gaussian parameter prediction.

Encoder-Decoder Backbone Render-FM employs a 3D U-Net architecture inspired by nnU-Net [14, 16]. The encoder applies successive blocks of two

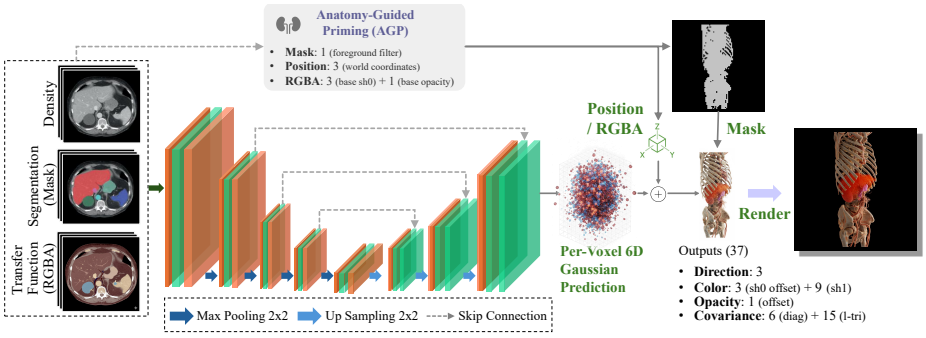

Fig. 2: Overview of the Render-FM pipeline. A 3D U-Net encoder-decoder network processes a 6-channel input volume and regresses 37-channel 6DGS parameters per voxel. Foreground voxels instantiate 6D Gaussians, which are rendered via differentiable view-dependent slicing and tile-based rasterization. End-to-end training is supervised by a rendering loss against physically-based ground-truth views.

$3 \times 3 \times 3$ convolutions with Instance Normalization and Leaky ReLU activation, interleaved with $2 \times 2 \times 2$ max-pooling for downsampling. The decoder mirrors this structure with transposed convolutions for $2 \times 2 \times 2$ upsampling and skip connections that concatenate encoder features at corresponding resolutions (excluding the final stage, which outputs at half the input resolution). The final $1 \times 1 \times 1$ convolution produces 37 output channels per voxel, yielding $\Psi \in \mathbb{R}^{37 \times \lfloor D/2 \rfloor \times \lfloor H/2 \rfloor \times \lfloor W/2 \rfloor}$. To satisfy the spatial divisibility requirements of the multi-scale pooling operations, input volumes are zero-padded to the nearest multiple of 32 along each spatial dimension prior to the forward pass.

6DGS Parameter Prediction Each spatial location in Ψ encodes the attributes of one potential 6D Gaussian primitive. The 3D position μ_p is fixed to the voxel's world-space coordinates and is not predicted by the network. The 37 predicted channels at each location encode: mean direction μ_d (3 channels), view-dependent color via Spherical Harmonic coefficients at degrees L=0 and L=1 (12 channels total), an opacity offset (1 channel), and the 21 upper-triangular elements of the 6×6 covariance matrix Σ. To guarantee a valid positive semi-definite covariance, the 6 diagonal elements are predicted in log-space, while the 15 off-diagonal elements are constrained via tanh activation before Cholesky reconstruction.

3.2 Anatomy-Guided Priming

Standard 3DGS initialization relies on SfM-derived sparse point clouds [25] or random placement, both unsuitable for volumetric medical data as they fail to leverage domain-specific anatomical information. Recent CT-based methods such as DDGS-CT [11] and 6DGS [12] improve upon this by using marching-cubes on CT radiodensity, but still ignore segmentation labels, transfer function

colors, and opacity, information that is readily available in clinical pipelines. We introduce Anatomy-Guided Priming (AGP) as a comprehensive initialization strategy for 6D Gaussian primitives. Since Render-FM predicts a 6D Gaussian for each voxel, Gaussian positions are set directly to voxel world coordinates; base colors and opacities are derived from pre-defined transfer functions; and semantic labels are assigned from the segmentation mask. This anatomically-informed starting point guides the network to predict residual refinements on top of meaningful priors, rather than learning geometry and appearance entirely from scratch, leading to more stable training and better generalization.

3.3 Differentiable Rendering

Gaussian Instantiation Since the network outputs Ψ at half the input resolution, each foreground voxel index in Ψ is first upsampled by a factor of 2 to recover full-resolution voxel coordinates, which are then mapped to physical world-space positions via the volume's affine transformation. Direction $\mu_{d,i}$ and covariance components are extracted from Ψ at the corresponding half-resolution location, and the 6×6 covariance matrix Σ_i is reconstructed from its predicted Cholesky factors. The appearance attributes are: color c_i, computed by combining AGP base RGB values with predicted spherical harmonic offsets; opacity $\alpha_i = \sigma(\alpha_{\text{base}} + \Delta\alpha_i)$, which blends the AGP prior with the predicted residual to ensure $\alpha_i \in [0, 1]$; and class label l_i, assigned from the segmentation mask.

Covariance Slicing for View-Dependence For rendering from a camera at position $\mathbf{p}$, we apply view-dependent covariance slicing to each Gaussian $i \in \Theta$ as described in [12]. The view direction is $\mathbf{v}_i = (\mu_{p,i} - \mathbf{p})/\|\mu_{p,i} - \mathbf{p}\|$. The 6×6 covariance Σ_i is partitioned into spatial (pp), directional (dd), and cross-term (pd, dp) blocks:

$$\Sigma_i = \begin{pmatrix} \Sigma_{pp} & \Sigma_{pd} \\ \Sigma_{dp} & \Sigma_{dd} \end{pmatrix}. \tag{1}$$

The view-conditioned spatial mean and covariance are then:

$$\mu'_{p,i} = \mu_{p,i} + \Sigma_{pd}\Sigma_{dd}^{-1}(\mathbf{v}_i - \mu_{d,i}), \tag{2}$$

$$\Sigma'_{pp,i} = \Sigma_{pp} - \Sigma_{pd}\Sigma_{dd}^{-1}\Sigma_{dp}, \tag{3}$$

with an opacity modulation factor $w_i = \mathcal{N}(\mathbf{v}_i \mid \mu_{d,i}, \Sigma_{dd})$ that scales visibility based on the alignment between the viewing direction and the Gaussian's directional distribution. This slicing mechanism adapts each Gaussian's shape, position, and opacity per viewpoint, capturing view-dependent optical effects such as subsurface scattering and specular reflectance at tissue interfaces.

Tile-Based Rasterization The view-conditioned Gaussians (defined by $\mu'_{p,i}$, $\Sigma'_{pp,i}$, modulated opacity $w_i\alpha_i$, and color from c_i) are rendered using a differentiable tile-based rasterizer adapted from 3DGS [18]. Gaussians are projected onto the image plane, sorted by depth, culled, and assigned to screen tiles. Alpha compositing is performed efficiently within each tile in parallel on the GPU. This fully differentiable process enables end-to-end training through the renderer and achieves real-time rendering speeds at inference.

3.4 Training

A central design choice of Render-FM is the unification of the nnU-Net [14,16] volumetric prediction backbone with the differentiable 6DGS rendering pipeline into a single end-to-end trainable system. Rather than treating parameter regression and rendering as separate stages, rendering gradients flow directly back through the rasterizer into the network weights, allowing the network to learn 6DGS parameters that are optimal for visual quality rather than for any intermediate proxy loss.

Concretely, at each training step, predicted Gaussians are rendered from 4 randomly sampled viewpoints $\mathbf{p}$ via differentiable rasterization $\hat{I} = R(\Theta, \mathbf{p})$ and compared against physically-based ground-truth images I_{gt} produced offline via PBRT [23]. We supervise training with a combination of pixel-level and perceptual losses:

$$\mathcal{L} = \lambda_{L1}\mathcal{L}_{L1} + \lambda_{\text{SSIM}}\mathcal{L}_{\text{SSIM}}, \tag{4}$$

where $\mathcal{L}_{L1} = \|\hat{I} - I_{gt}\|_1$ penalizes pixel-level error and $\mathcal{L}_{\text{SSIM}} = 1 - \text{MS-SSIM}(\hat{I}, I_{gt})$ [28] enforces structural fidelity, with weights $\lambda_{L1} = 0.8$ and $\lambda_{\text{SSIM}} = 0.2$. View losses are averaged before backpropagation across all training scans.

3.5 Inference Pipeline

The Render-FM inference pipeline first preprocesses the CT volume: it is resampled to isotropic 1.5 mm spacing, a segmentation mask is either provided or generated automatically (*e.g.*, via TotalSegmentator [30]), and per-voxel RGBA values are computed from pre-defined transfer functions. The normalized CT intensity, segmentation mask, and RGBA volume are assembled into the 6-channel input V, zero-padded to the nearest multiple of 32.

The assembled input is passed through f_ϕ in a single forward pass, producing the dense parameter volume Ψ. Foreground voxels identified by the segmentation mask are immediately instantiated as 6D Gaussians with geometry and appearance attributes derived from Ψ and the AGP priors, as described above. These 6D Gaussians are ready for real-time interactive rendering without any further optimization, enabling the clinician to inspect the anatomy from arbitrary viewpoints at 328+ FPS. The entire process from raw CT to interactive visualization completes in under 3 seconds on GPU, compared to approximately one hour required by per-scan optimization methods.

For cases requiring higher rendering fidelity, the instantiated 6DGS model can optionally be fine-tuned for less than 2 minutes. This fine-tuning optimizes the predicted Gaussian parameters directly on the test scan using the same differentiable renderer and loss, providing a quality boost while remaining far more practical than full per-scan optimization. The two-stage design, fast feedforward prediction followed by optional lightweight refinement, offers a flexible trade-off between speed and quality to suit different clinical demands.

4 Experiments

We conduct extensive experiments to evaluate the performance of Render-FM against the 6DGS baseline [12], focusing on rendering quality, computational efficiency, and generalizability. We first ablate the role of anatomy-guided priming (AGP) as both an initialization strategy and a necessary condition for feedforward learning, then present a full quantitative and qualitative comparison across in-domain and out-of-domain settings, including seen and unseen transfer functions, fine-tuning, and compositional organ visualization.

4.1 Experimental Protocol

Datasets We utilized two publicly available CT datasets: 1) **TotalSegmentator [30]:** This dataset comprises 1,228 CT scans from clinical routines, covering 117 anatomical classes. It includes a wide range of pathologies, scanners, acquisition protocols, and institutions, making it representative of real-world clinical variability. After filtering scans exceeding 48 million voxels (due to GPU memory constraints) or lacking orthonormal directionality, we used 991 scans for training and 46 for in-domain (*ID*) testing; 2) **CT-ORG [24]:** This dataset includes 140 CT scans from diverse sources, featuring both large organs (*e.g.*, lungs) and small, challenging structures (*e.g.*, bladder). We selected 10 scans (volumes 2–11) for out-of-domain (*OOD*) testing to evaluate Render-FM's generalizability to unseen data distributions.

Data Preparation To prepare the training data, we applied a standardized preprocessing pipeline: 1) *Resampling:* Volumes were resampled to isotropic spacing of 1.5 mm in all dimensions to ensure consistency; 2) *Normalization:* CT intensities (Hounsfield Units) were normalized following the nnU-Net pipeline [14] to standardize intensity ranges; 3) *Segmentation:* Segmentation masks were generated using TotalSegmentator [30], grouping 117 anatomical classes into 11 semantic categories (*e.g.*, skeleton, muscle, cardioVascular; see Supplementary Material). Note that masks can also be manually annotated or automatically generated by other methods, ensuring flexibility for clinical use; 4) *Transfer Functions:* We defined RGBA transfer functions for the 11 semantic groups to provide base appearance cues, enhancing the model's ability to predict Gaussian parameters; and 5) *Ground-Truth Rendering:* For each training scan, we rendered 60 views of all classes, 30 views of skeleton group, and 30 views of organ groups (except the skeleton and muscle groups) with the resolution of 1600×1600 using physically-based rendering via PBRT [23] to serve as ground truth, which takes 18.8 seconds per view on average. To enhance model robustness, we applied data augmentations, including random intensity shifts, Gaussian noise, and simulated acquisition artifacts, mimicking variations in clinical CT imaging.

For evaluation, we applied identical preprocessing to both test sets: 46 TotalSegmentator scans (*ID*) and 10 CT-ORG scans (*OOD*). Our experiments evaluated performance under three conditions: 1) *Seen TF*: using transfer functions identical to training for testing generalization to new scans; 2) *Unseen TF*:

using novel transfer functions not seen during training to test appearance generalizability; and 3) *Skeleton group*: visualizing only skeletal structures to evaluate compositional capabilities. For each test scan, we rendered 40 views for computing evaluation metrics, while 20 views were used for training 6DGS baseline or fine-tuning Render-FM.

For semantic labeling of the 6DGS baseline, the original implementation does not support per-Gaussian class assignment. We therefore extended it using k-nearest neighbors (with k=1) to assign anatomical classes from the segmentation mask to each Gaussian primitive during initialization. In contrast, our anatomy-guided priming approach (6DGS + AGP) integrates semantic classification directly during the initialization process, providing more consistent anatomical structure representation.

Implementation Details Render-FM was implemented in PyTorch using the Adam optimizer [19] ($\beta_1 = 0.9$, $\beta_2 = 0.999$) with an initial learning rate of 1×10^{-3} and PolyLR scheduling [5]. Due to GPU memory constraints, we used a batch size of 3 volumes, leveraging automatic mixed precision (AMP) and gradient accumulation for efficiency. Training was performed on a single NVIDIA A100 80GB GPU, requiring approximately 3 days. At inference, we additionally adopt FlashGS [10] to accelerate rasterization throughput. The number of instantiated Gaussians varied by scan complexity, ranging from 50,000 to 800,000, depending on the anatomical structures present.

The 6DGS experiments, including the original 6DGS and 6DGS with our AGP initialization (*i.e.*, 6DGS + AGP), were trained following the official implementation [12] for 30,000 iterations, with evaluations every 500 iterations to mitigate potential overfitting when training with limited views (20 per scan). We reported the best results for 6DGS experiments to ensure fair comparison. For Render-FM fine-tuning (FT), we conducted 300 optimization iterations, representing a substantial reduction in computational requirements, and reported the final results.

Evaluation Metrics To assess rendering quality, we employed three standard metrics: Peak Signal-to-Noise Ratio (PSNR), which quantifies pixel-level accuracy with higher values indicating better fidelity; Structural Similarity Index (SSIM) [29], which evaluates structural and perceptual similarity; and Learned Perceptual Image Patch Similarity (LPIPS) [33], which measures perceptual similarity with lower values indicating closer alignment to human visual perception. For efficiency, we measured preparation time (in seconds), defined as the duration from raw CT input to 6DGS interactive rendering readiness; the number of Gaussian points, which reflects model complexity; and frames per second (FPS), which quantifies real-time rendering performance.

4.2 Ablation: Anatomy-Guided Priming

We evaluate the contribution of Anatomy-Guided Priming (AGP) from two perspectives: its effect as an initialization strategy for per-scan optimization (6DGS

Table 1: Quantitative comparison. *ID*: in-domain (TotalSegmentator); *OOD*: out-of-domain (CT-ORG); *Seen/Unseen TF*: transfer functions used/not used for training; *AGP*: anatomy-guided priming; *Skeleton group*: compositional visualization requiring 0.0s additional preparation, enabled by per-Gaussian labels assigned during AGP.

Dataset	Type	Method	SSIM	PSNR	LPIPS	Time (s)	# points	FPS
TotalSeg	*ID* *Seen TF*	6DGS	0.912	26.63	0.096	1463.9	68,785	697.5
		6DGS + AGP (**Ours**)	0.925	28.92	0.093	1786.5	135,827	575.6
		Render-FM (**Ours**)	0.919	27.30	0.097	**2.8**	343,058	328.6
		Render-FM (**Ours**) + FT	**0.937**	**31.67**	**0.088**	89.4	227,925	423.8
CT-ORG	*OOD* *Seen TF*	6DGS	0.903	25.97	0.105	1528.7	75,956	679.1
		6DGS + AGP (**Ours**)	0.926	29.36	0.091	2261.9	239,609	411.0
		Render-FM (**Ours**)	0.918	26.21	0.092	**2.6**	586,225	245.2
		Render-FM (**Ours**) + FT	**0.940**	**32.48**	**0.082**	136.2	469,969	275.1
CT-ORG	*OOD* *Unseen TF*	6DGS	0.908	26.59	0.103	1546.4	81,802	684.3
		6DGS + AGP (**Ours**)	0.924	29.66	0.090	2129.3	246,017	385.5
		Render-FM (**Ours**)	0.913	26.77	0.092	**2.8**	586,225	245.5
		Render-FM (**Ours**) + FT	**0.936**	**31.91**	**0.083**	133.4	462,219	277.5
CT-ORG	*OOD* *Seen TF* *Skeleton group*	6DGS	0.935	26.78	0.066	5146.0	76,848	602.0
		6DGS + AGP (**Ours**)	0.938	28.95	0.064	7545.3	286,308	425.1
		Render-FM (**Ours**)	0.925	26.10	0.070	**0.0**	586,225	295.7
		Render-FM (**Ours**) + FT	**0.944**	**30.74**	**0.061**	140.1	466,638	337.3

+ AGP vs. 6DGS), and its role as a necessary condition for feedforward learning in Render-FM.

AGP as initialization for 6DGS As shown in Table 1, applying AGP to the 6DGS baseline consistently improves rendering quality across all conditions. On TotalSegmentator (ID, Seen TF), 6DGS + AGP achieves SSIM of 0.925, PSNR of 28.92, and LPIPS of 0.093, compared to 0.912, 26.63, and 0.096 for vanilla 6DGS, with similar gains on CT-ORG under both seen (SSIM: 0.926 vs. 0.903) and unseen transfer functions (SSIM: 0.924 vs. 0.908). These improvements confirm that initializing Gaussians with anatomically-grounded positions, colors, and opacities from transfer functions provides a stronger starting point than marching-cubes. As a natural consequence of this richer initialization, fewer Gaussians are pruned during optimization, yielding a higher final point count (135,827 vs. 68,785 on TotalSegmentator) that better covers anatomical structures, which is a desired outcome reflecting improved scene representation. This also explains the longer optimization time (1786.5s vs. 1463.9s). Nonetheless, 6DGS + AGP still requires per-scan optimization for every new case.

AGP as a necessary component of Render-FM For our Render-FM feedforward model, AGP is not merely beneficial but essential. Without AGP, Render-FM fails to converge during training. This is because the feedforward network must predict meaningful 6DGS parameters for hundreds of thousands of voxels simultaneously, a highly underconstrained problem without a structured starting point. AGP resolves this by providing each Gaussian with a physically grounded initialization: positions anchored to voxel world coordinates, colors

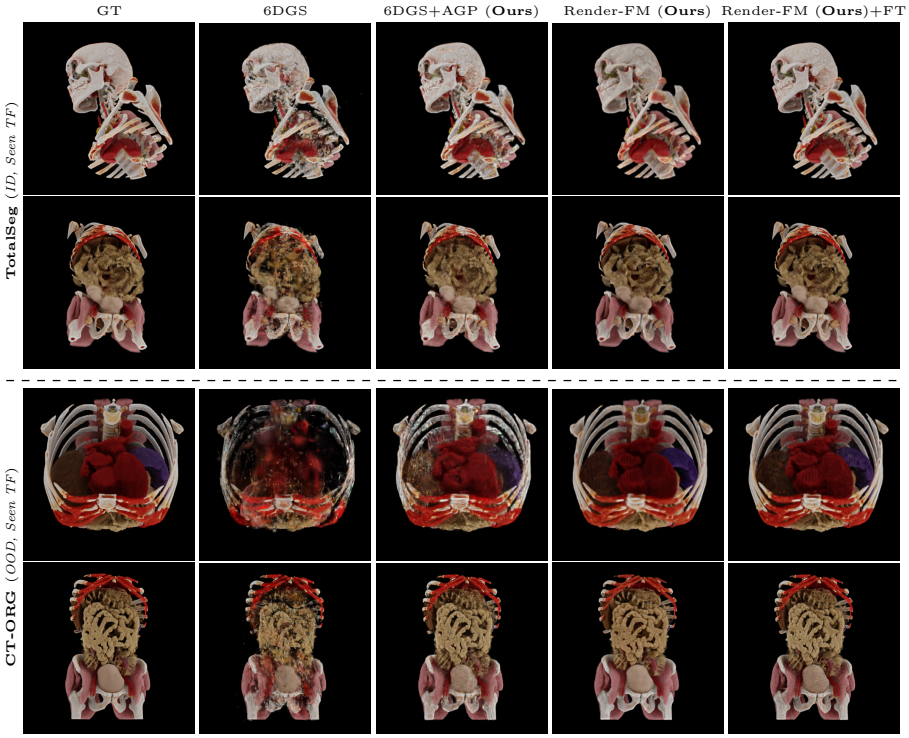

Fig. 3: Qualitative comparison on two datasets: in-domain (*ID*) TotalSegmentator [30] and out-of-domain (*OOD*) CT-ORG [24] for seen transfer functions (*Seen TF*).

and opacities derived from transfer functions, and semantic labels from the segmentation mask. This transforms the learning objective from predicting absolute parameters from scratch into predicting residual corrections on top of meaningful priors, making the optimization landscape tractable. The convergence failure without AGP thus confirms that anatomy-guided initialization is a fundamental design requirement of Render-FM, not an optional enhancement.

4.3 Comparison with Baseline

Table 1 summarizes the performance of Render-FM, the 6DGS baseline [12], and Render-FM with fine-tuning (*FT*) across the TotalSegmentator (*ID*) [30] and CT-ORG (*OOD*) [24] datasets, under *Seen TF*, *Unseen TF*, and *Skeleton group* conditions.

In-Domain (*TotalSeg, Seen TF*) On the TotalSegmentator test set with seen transfer functions, Render-FM surpasses the 6DGS baseline across all quality metrics (SSIM: 0.919 vs. 0.912, PSNR: 27.30 vs. 26.63, LPIPS: 0.097 vs. 0.096) without any per-scan optimization. Preparation time is reduced from 1463.9 seconds to just 2.8 seconds, a 500-fold improvement, while sustaining

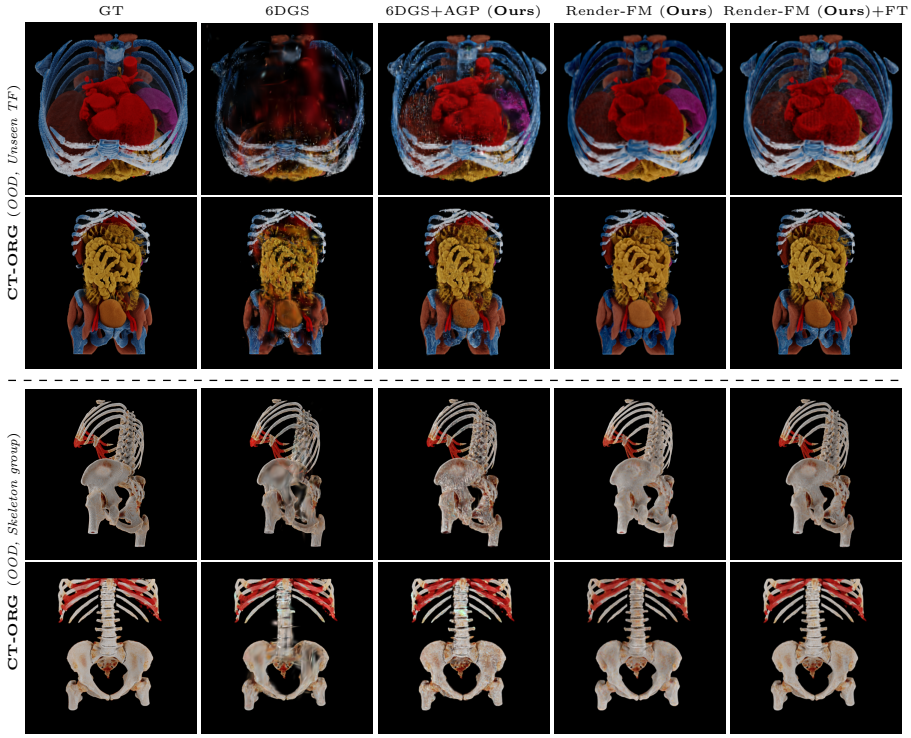

Fig. 4: Qualitative comparison on out-of-domain (*OOD*) CT-ORG [24] for unseen transfer functions (*Unseen TF*) and composability (*Skeleton group*).

real-time rendering at 328.6 FPS. Fine-tuning further elevates performance substantially (SSIM: 0.937, PSNR: 31.67, LPIPS: 0.088), outperforming all methods at a preparation time of only 89.4 seconds.

Out-of-Domain (*CT-ORG, Seen TF*) For out-of-domain testing on CT-ORG with seen transfer functions, Render-FM demonstrates robust generalization, outperforming 6DGS on all metrics (SSIM: 0.918 vs. 0.903, PSNR: 26.21 vs. 25.97, LPIPS: 0.092 vs. 0.105) with a preparation time of just 2.6 seconds at a rendering speed of 245.2 FPS. Fine-tuning significantly enhances performance further (SSIM: 0.940, PSNR: 32.48, LPIPS: 0.082), achieving the best results across all metrics with a preparation time of 136.2 seconds.

Unseen Transfer Functions (*CT-ORG, Unseen TF*) When evaluated with novel transfer functions unseen during training, Render-FM maintains strong performance (SSIM: 0.913, PSNR: 26.77, LPIPS: 0.092), outperforming the 6DGS baseline (SSIM: 0.908, PSNR: 26.59, LPIPS: 0.103) and demonstrating that the learned representation generalizes to unseen appearance mappings without retraining. Fine-tuning further improves results (SSIM: 0.936, PSNR: 31.91, LPIPS: 0.083) with a preparation time of 133.4 seconds.

Composability (*CT-ORG, Skeleton group*) We further evaluated Render-FM's capability for compositional organ visualization. By selectively rendering Gaussians corresponding to specific anatomical classes (*e.g.*, lungs, liver, bones) based on the segmentation mask, Render-FM enables interactive exploration of individual or combined structures with zero additional preparation time (0.0s). This is a fundamental advantage over 6DGS: incorporating semantic labels into 6DGS requires a full separate per-scan optimization pass, inflating preparation time from 1528.7s to 5146.0s. Render-FM achieves SSIM of 0.925, PSNR of 26.10, and LPIPS of 0.070 without any reprocessing, and fine-tuning further improves results to SSIM of 0.944, PSNR of 30.74, and LPIPS of 0.061 with only 140.1 seconds of preparation.

Across all conditions, Render-FM consistently delivers rendering quality comparable to or better than per-scan optimized 6DGS, with preparation times reduced by two to three orders of magnitude. The dense Gaussian prediction strategy results in a higher point count and consequently lower FPS for Render-FM (245.2–328.6 without fine-tuning; up to 423.8 with fine-tuning) compared to 6DGS (602.0–697.5), yet all values **far exceed** the threshold for real-time clinical interaction. Fine-tuning leverages Render-FM's robust initialization to achieve state-of-the-art quality with minimal additional computation.

Figures 3 and 4 provide a qualitative comparison across all methods. The 6DGS baseline, trained on 20 sparse views per scan, often overfits, producing floating noise artifacts in novel views. AGP-initialized 6DGS reduces these artifacts but does not fully eliminate them. In contrast, Render-FM leverages its learned generalizable priors to largely mitigate floating noise, though it may appear slightly blurry in some cases. After brief fine-tuning, Render-FM significantly enhances visual quality, capturing intricate details such as vasculature and bone interfaces with superior clarity, which are critical for clinical interpretation and diagnostic accuracy.

5 Conclusion

We presented Render-FM, a feedforward model for real-time, high-fidelity volumetric rendering of CT scans using 6D Gaussian Splatting. By employing an nnU-Net inspired encoder-decoder architecture trained end-to-end, Render-FM directly regresses 6DGS parameters from a multi-channel CT input volume, without the need for time-consuming per-scan optimization. The model leverages large-scale pre-training to learn generalizable mappings from CT data to expressive, view-dependent 3D representations. Our experiments demonstrate that Render-FM achieves rendering quality comparable or better than that of per-scan optimized methods while reducing preparation time from hours to seconds, enabling interactive frame rates suitable for clinical use. By integrating the robustness of nnU-Net with the expressive power and view-dependent rendering capabilities of 6DGS, Render-FM bridges the gap between advanced neural rendering quality and clinical practicality.

Future Work. While this work focuses on CT, the Render-FM pipeline is designed to be general: given modality-appropriate input channels and transfer functions, the same feedforward framework extends to other volumetric modalities such as MRI, which we leave as a promising direction for future work.

References

1. Beyer, J., Hadwiger, M., Wolfsberger, S., Bühler, K.: High-quality multimodal volume rendering for preoperative planning of neurosurgical interventions. IEEE Transactions on Visualization and Computer Graphics **13**(6), 1696–1703 (2007)
2. Bommasani, R., Hudson, D.A., Adeli, E., Altman, R., et al.: On the Opportunities and Risks of Foundation Models. arXiv preprint arXiv:2108.07258 (2021)
3. Cai, Y., Wang, J., Yuille, A., Zhou, Z., Wang, A.: Structure-aware sparse-view x-ray 3d reconstruction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11174–11183 (2024)
4. Caton Jr, M.T., Wiggins, W.F., Nunez, D.: Three-dimensional cinematic rendering to optimize visualization of cerebrovascular anatomy and disease in ct angiography. Journal of Neuroimaging **30**(3), 286–296 (2020)
5. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence **40**(4), 834–848 (2017)
6. Dappa, E., Higashigaito, K., Fornaro, J., Leschka, S., Wildermuth, S., Alkadhi, H.: Cinematic rendering–an alternative to volume rendering for 3d computed tomography imaging. Insights into imaging **7**, 849–856 (2016)
7. Eid, M., De Cecco, C.N., Nance Jr, J.W., Caruso, D., Albrecht, M.H., Spandorfer, A.J., De Santis, D., Varga-Szemes, A., Schoepf, U.J.: Cinematic rendering in ct: a novel, lifelike 3d visualization technique. American Journal of Roentgenology **209**(2), 370–379 (2017)
8. Elshafei, M., Binder, J., Baecker, J., Brunner, M., Uder, M., Weber, G.F., Grützmann, R., Krautz, C.: Comparison of cinematic rendering and computed tomography for speed and comprehension of surgical anatomy. JAMA surgery **154**(8), 738–744 (2019)
9. Ertl, A., Xiao, S., Denner, S., Peretzke, R., Zimmerer, D., Neher, P., Isensee, F., Maier-Hein, K.H.: nnLandmark: A Self-Configuring Method for 3D Medical Landmark Detection. arXiv preprint arXiv:2504.06742 (2025)
10. Feng, G., Chen, S., Fu, R., Liao, Z., Wang, Y., Liu, T., Pei, Z., Li, H., Zhang, X., Dai, B.: Flashgs: Efficient 3d gaussian splatting for large-scale and high-resolution rendering. arXiv preprint arXiv:2408.07967 (2024)
11. Gao, Z., Planche, B., Zheng, M., Chen, X., Chen, T., Wu, Z.: Ddgs-ct: Direction-disentangled gaussian splatting for realistic volume rendering. In: The Thirty-eighth Annual Conference on Neural Information Processing Systems (2024)
12. Gao, Z., Planche, B., Zheng, M., Choudhuri, A., Chen, T., Wu, Z.: 6DGS: Enhanced Direction-Aware Gaussian Splatting for Volumetric Rendering. arXiv preprint arXiv:2410.04974 (2024)
13. Heng, Y., Gu, L.: Gpu-based volume rendering for medical image visualization. In: 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference. pp. 5145–5148. IEEE (2006)

14. Isensee, F., Jäger, P.F., Kohl, S.A.A., Petersen, J., Maier-Hein, K.H.: nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. Nature Methods **18**, 203–211 (2021)
15. Isensee, F., Rokuss, M., Krämer, L., Dinkelacker, S., Ravindran, A., Stritzke, F., Hamm, B., Wald, T., Langenberg, M., Ulrich, C., et al.: nninteractive: Redefining 3d promptable segmentation. arXiv preprint arXiv:2503.08373 (2025)
16. Isensee, F., Wald, T., Ulrich, C., Baumgartner, M., Roy, S., Maier-Hein, K., Jaeger, P.F.: nnu-net revisited: A call for rigorous validation in 3d medical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 488–498. Springer (2024)
17. Jung, Y., Kim, J., Bi, L., Kumar, A., Feng, D.D., Fulham, M.: A direct volume rendering visualization approach for serial pet–ct scans that preserves anatomical consistency. International Journal of Computer Assisted Radiology and Surgery **14**, 733–744 (2019)
18. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. **42**(4), 139–1 (2023)
19. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
20. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)
21. Moor, M., Banerjee, O., Abad, Z.S.H., Krumholz, H.M., Leskovec, J., Topol, E.J., Rajpurkar, P.: Foundation models for generalist medical artificial intelligence. Nature **616**(7956), 259–265 (2023)
22. Paschali, M., Chen, Z., Blankemeier, L., Varma, M., Youssef, A., Bluethgen, C., Langlotz, C., Gatidis, S., Chaudhari, A.: Foundation models in radiology: What, how, why, and why not. Radiology **314**(2), e240597 (2025)
23. Pharr, M., Jakob, W., Humphreys, G.: Physically based rendering: From theory to implementation. MIT Press (2023)
24. Rister, B., Yi, D., Shivakumar, K., Nobashi, T., Rubin, D.L.: Ct-org, a new dataset for multiple organ segmentation in computed tomography. Scientific Data **7**(1), 381 (2020)
25. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
26. Siemens Healthineers: Cinematic rendering for medical imaging. White paper, Siemens Healthineers (2026), `https://cdn0.scrvt.com/39b415fb07de4d9656c7b516d8e2d907/1800000004349360/6658a1860496/DI_SY_CinematicRendering_1800000004349360.pdf`, powered by syngo.via. Accessed: 2026-03-05
27. Tang, J., Chen, Z., Chen, X., Wang, T., Zeng, G., Liu, Z.: Lgm: Large multi-view gaussian model for high-resolution 3d content creation. arXiv preprint arXiv:2402.05054 (2024)
28. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing **13**(4), 600–612 (2004)
29. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)
30. Wasserthal, J., Breit, H.C., Meyer, M.T., Pradella, M., Hinck, D., Sauter, A.W., Heye, T., Boll, D.T., Cyriac, J., Yang, S., et al.: Totalsegmentator: robust segmen-

tation of 104 anatomic structures in ct images. Radiology: Artificial Intelligence **5**(5), e230024 (2023)
31. Wollschlaeger, L.M., Boos, J., Jungbluth, P., Grassmann, J.P., Schleich, C., Latz, D., Kroepil, P., Antoch, G., Windolf, J., Schaarschmidt, B.M.: Is ct-based cinematic rendering superior to volume rendering technique in the preoperative evaluation of multifragmentary intraarticular lower extremity fractures? European Journal of Radiology **126**, 108911 (2020)
32. Xu, Y., Shi, Z., Yifan, W., Chen, H., Yang, C., Peng, S., Shen, Y., Wetzstein, G.: Grm: Large gaussian reconstruction model for efficient 3d reconstruction and generation. In: European Conference on Computer Vision. pp. 1–20. Springer (2024)
33. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)