VOXELTTO: VOXEL-ALIGNED FEED-FORWARD 3D GAUSSIAN SPLATTING WITH TEST-TIME OPTIMIZATION

Anonymous authors
Paper under double-blind review

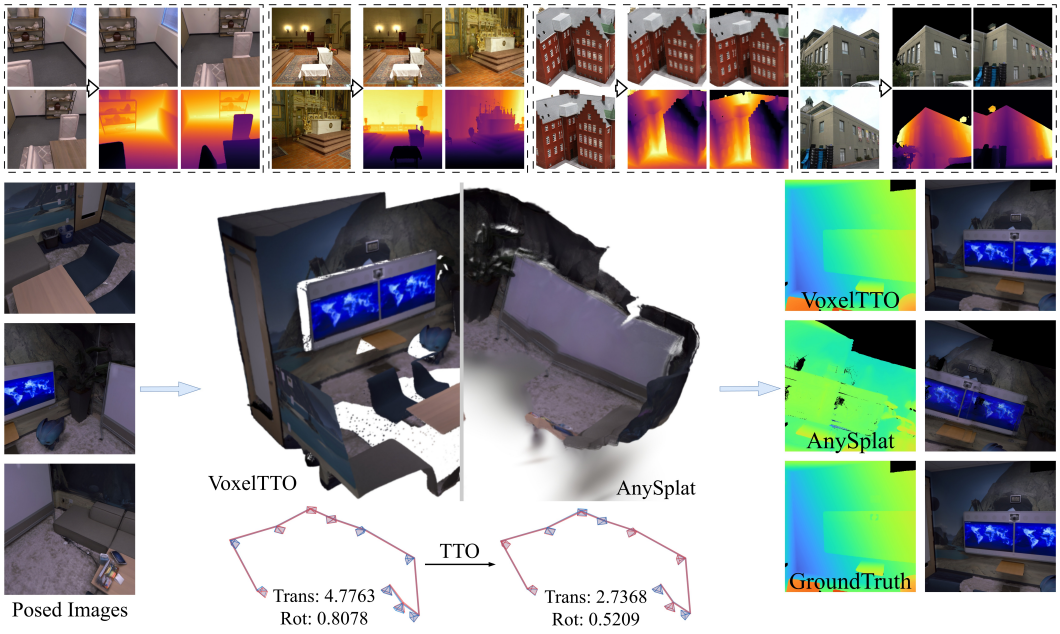

Figure 1: VoxelTTO directly regresses a geometrically accurate Gaussian-splatting scene from posed images. It precisely aligns camera poses and improves RGB and depth rendering in both input and test views across indoor–outdoor and synthetic–realworld scenes.

ABSTRACT

Recent feed-forward 3D Gaussian Splatting (3DGS) methods typically regress pixel-aligned Gaussian primitives, often causing excessive overlap and artifacts, while inaccuracies in predicted camera poses can lead to misalignment in novel-view synthesis (NVS). We present VoxelTTO, a feed-forward framework for reconstructing geometrically accurate 3DGS scenes from an arbitrary number of images and optional camera parameters. VoxelTTO aggregates dense image features into a global voxel representation and decodes Gaussians from voxel features, breaking the pixel-to-Gaussian correspondence. To exploit known camera parameters while keeping the pretrained visual foundation model (VFM) parameters frozen, we introduce test-time optimization (TTO) that adapts lightweight LoRA modules using pose supervision. We further replace vanilla 3DGS rasterization with stochastic solid volume rendering during training and inference, improving geometric fidelity. Training updates only the voxel-aligned Gaussian reconstruction modules, requiring 80 GPU hours. Experiments on Replica, Tanks and Temples, and DTU demonstrate improved RGB-D NVS and camera-pose estimation relative to prior methods.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Conventional 3D reconstruction systems from images rely on multistage pipelines comprising feature matching Sarlin et al. (2020); DeTone et al. (2018) and structure from motion (SfM) Schönberger & Frahm (2016). Recent VFMs Wang et al. (2025a; 2024b) offer an end-to-end alternative, using a large Transformer to directly predict 3D attributes from arbitrary image collections.

Despite their strong geometric priors, point-map and depth-map representations produced by VFMs often fail to capture fine details and photorealistic appearance. By contrast, 3DGS Kerbl et al. (2023), and their extensions Huang et al. (2024); Lu et al. (2024) use richer scene representations to deliver accurate NVS. However, these methods commonly recover camera parameters with SfM before performing costly per-scene optimization. Some approaches directly predict 3DGS scenes Xu et al. (2025); Charatan et al. (2024); Chen et al. (2024), but they typically require accurate camera parameters and can degrade in complex scenes. More recent methods Jiang et al. (2025); Lin et al. (2025); Ye et al. (2025) extend the feed-forward VFMs to 3DGS. They regress pixel-aligned 3DGS scenes directly from unposed images, but their geometric accuracy remains limited. Several VFMs encode camera parameters within the backbone to exploit additional priors and improve prediction accuracy Keetha et al. (2026); Lin et al. (2025); Jiang et al. (2025). Nevertheless, the resulting accuracy remains insufficient, and persists substantial misalignment during test-view synthesis.

In this work, we present VoxelTTO, a voxel-aligned feed-forward 3DGS framework with test-time optimization (TTO). VoxelTTO reconstructs geometrically accurate 3DGS scenes from arbitrary numbers of images and optional camera parameters. When camera priors are available, it supports accurate NVS from test views. To reduce excessive overlap and artifacts, VoxelTTO departs from conventional pixel-aligned feed-forward 3DGS, upsamples patch tokens from a frozen VFM backbone with DPT and fuses them into global voxels. A sparse convolutional U-Net encodes the voxel features, and an appearance-geometry decoupled self-splitting MLP regresses Gaussian primitive parameters. We replace 3DGS rasterization with stochastic solid volume rendering, which provides stronger geometric supervision during optimization and improves geometric fidelity for NVS .

To exploit camera priors, TTO inserts LoRA modules into the VFM backbone. Pose-supervised adaptation of these modules improves alignment between predicted and ground-truth camera parameters. During training, we freeze the VFM backbone and optimize only the Voxel-Aligned Feed-Forward 3DGS module. This design introduces 75.3M trainable parameters, significantly reduces training cost to 80 GPU hours, and is compatible with any VGGT-style VFMs.

Our main contributions are as follows:

- We propose VoxelTTO, a voxel-aligned feed-forward 3DGS reconstruction framework that recovers geometrically accurate scenes from arbitrary image collections and optional camera parameters.
- We introduce voxel-aligned feed-forward 3DGS and stochastic solid volume rendering supervision to reduce excessive overlap and artifacts while strengthening geometry.
- We introduce pose-supervised TTO to exploit camera priors and improve alignment.
- Extensive experiments across multiple datasets demonstrate improved camera-pose estimation and more accurate RGB-D rendering in both input and test views compared to existing feed-forward 3DGS methods.

2 RELATED WORKS

2.1 OPTIMIZATION-BASED NOVEL-VIEW SYNTHESIS

Neural radiance fields (NeRF) Mildenhall et al. (2020) model scene density and view-dependent color with neural fields, but ray-wise volume rendering is computationally expensive. 3DGS Kerbl et al. (2023) represents scenes with anisotropic Gaussian primitives and enables real-time NVS; subsequent work improves its geometric representation Huang et al. (2024); Dai et al. (2024) and optimization Huang et al. (2025); Guédon & Lepetit (2024). Stochastic solid volume rendering is equivalent to 3DGS rasterization under suitable conditions Miller et al. (2023); Zhang et al. (2026); Gomez et al. (2026) and improves depth and surface reconstruction.

These optimization-based methods require accurate camera parameter, which are typically estimated with SfM pipelinesSchönberger & Frahm (2016); Schönberger et al. (2016); Wang et al. (2024a;b). Several methods jointly optimize camera parameters and 3DGS scenes, but generally require sequential images Keetha et al. (2024); Fu et al. (2024) or depth maps Ha et al. (2024). Vanilla 3DGS initializes primitives by KNN distances from a colored point cloud and optimizes each scene independently Kerbl et al. (2023). Subsequent research s directly regress a 3DGS scene from images and camera parameters without optimization Charatan et al. (2024); Chen et al. (2024); Xu et al. (2025).

2.2 Visual Foundation Models

Image-based reconstruction has traditionally relied on separate SfM and dense-reconstruction stages. Recent VFMs directly regress 3D scene information from multiple views. Dust3R Wang et al. (2024b) uses self and cross-attention to infer aligned point maps and camera parameters from image pairs. VGGT alternates single image attention with global attention to regress depth maps, point maps, and camera parameters, from arbitrary views. Later work increases model accuracyLin et al. (2025); Wang et al. (2026) and improves order invariance Wang et al. (2025c). Several VFMs Jiang et al. (2025); Lin et al. (2025); Ye et al. (2025) further support direct regression of 3DGS. Their Gaussian primitives are usually pixel-aligned with depth maps, which inevitably introduces excessive overlap and rendering artifacts. Directly regressing Gaussians from fixed tokens Ren et al. (2026); Li et al. (2026) alleviates this issue but commonly sacrifices geometric accuracy. Recent works begin to explore aligning Gaussian primitives with voxel-based representationsWang et al. (2025b).

VFM geometry is predicted across views and may not strictly satisfy multiview constraints. Test-time optimization (TTO) has shown benefits for Vision Transformer tasks Han et al. (2026) and been applied to VFMs Elflein et al. (2026). Recent TCO inserts LoRA modules into attention layers and constrains them using known camera parameters and depth maps, improving pose estimation and dense reconstruction Zhou et al. (2026). Unlike approaches that encode depth maps or camera parameters into the backbone Keetha et al. (2026); Lin et al. (2025); Jiang et al. (2025), TCO adapts lightweight modules without changing the base VFM architecture.

3 Method

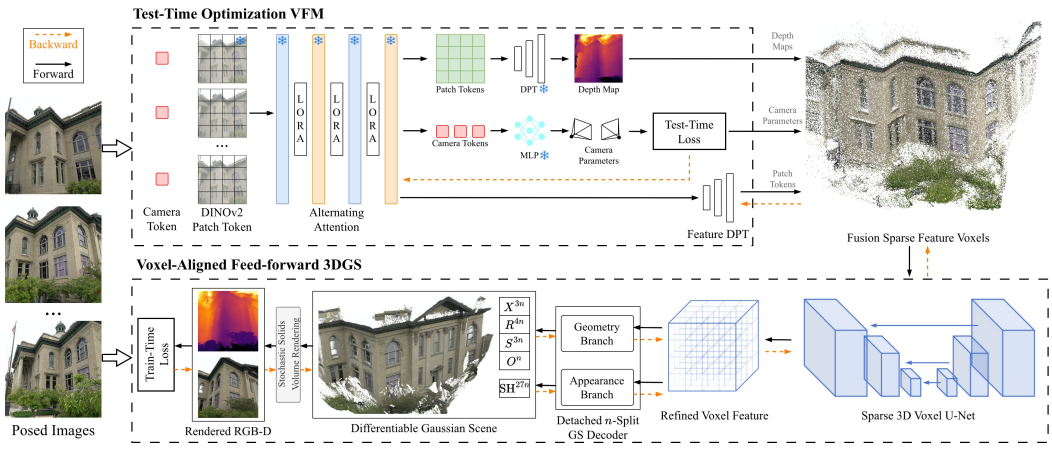

Figure 2: Overview of the VoxelTTO pipeline.

We propose VoxelTTO, a feed-forward method that predicts a 3DGS scene from multiview images and optional camera parameters. By aggregating dense features into voxels and processing them with a sparse-convolutional U-Net, VoxelTTO delivers high geometric fidelity and appearance performance. TTO further addresses the misalignment commonly observed when VGGT-style VFMs predict Gaussian-splatting scenes, improving accuracy in test views.

3.1 PRELIMINARIES AND CHALLENGES

VFMs Wang et al. (2025a; 2024b) use a large Transformer Φ_θ to learn cross-view visual representations from M uncalibrated images $\mathbf{I} = \{\mathbf{I}_m\}_{m=1}^{M}$. Separate prediction heads Γ_ψ recover 3D scene information, including point maps $\mathbf{P}_m \in \mathbb{R}^{H\times W\times 3}$, depth maps $\mathbf{D}_m \in \mathbb{R}^{H\times W}$, and camera parameters $\mathbf{g}_m$, as shown in Equation 1.

$$(\mathbf{P}_m, \mathbf{D}_m, \mathbf{g}_m) = \Gamma_\psi\left(\Phi_\theta\left(\mathbf{I}\right)\right), \qquad m = 1, \ldots, M \tag{1}$$

3DGS VFMs typically upsample patch tokens with DPT to obtain pixel-aligned Gaussian attributes (center μ_q, opacity σ_q, rotation quaternion r_q, scale s_q, and color c_q). Such methods often lack multiview consistency and create **excessive overlap, producing artifacts**. Direct token-to-Gaussian methods commonly have lower geometric accuracy Li et al. (2026).

To exploit camera priors, several VFMs Keetha et al. (2026); Lin et al. (2025); Jiang et al. (2025) encode ground-truth camera parameters $\mathbf{g}_{\text{gt}}$ as tokens before the backbone and incorporate them during inference. Although fusing auxiliary camera parameters with RGB features improves performance, it changes the architecture and prevents direct reuse of unconstrained VFMs. Moreover, the inferred camera parameters $\mathbf{g}_m$ can still differ substantially from $\mathbf{g}_{\text{gt}}$. Even after Umeyama SIM(3) alignment Umeyama (1991), this error causes substantial **caemra misalignment** in subsequent NVS.

3.2 VOXEL-ALIGNED FEED-FORWARD 3DGS

3.2.1 NETWORK ARCHITECTURE

VoxelTTO uses DINOv2 Oquab et al. (2023) to extract multiview patch tokens and employs alternating attention to model the scene. We add LoRA to the alternating-attention layers for TTO, as detailed in subsection 3.3. The frozen VFM backbone first predicts a depth map $\mathbf{D}_m$, camera parameters $\mathbf{g}_m$, and confidence map $\mathbf{K}_m$ for each view. We add a feature DPT that upsamples patch tokens into dense pixel features $\mathbf{F}_m \in \mathbb{R}^{H\times W\times C}$. We then back-project pixels into world coordinates, obtaining a dense feature point cloud $\mathcal{P} = \{(\mathbf{p}_l, \mathbf{a}_l, \mathbf{f}_l, \kappa_l)\}_{l=1}^{L}$, where $\mathbf{p}_l$, $\mathbf{a}_l$, $\mathbf{f}_l$, and κ_l denote point coordinate, color, feature, and confidence, respectively. To reduce memory use and avoid excessive overlap, we aggregate the point cloud into voxel features. Let $\mathcal{P}_v$ denote the points within voxel v. Its feature center is $\mathbf{x}_v = \sum_{l\in\mathcal{P}_v} \kappa_l \mathbf{p}_l / \sum_{l\in\mathcal{P}_v} \kappa_l$, which is not constrained to the regular voxel center and thus preserves geometric accuracy. Voxel color $\mathbf{a}_v$ and feature $\mathbf{f}_v$ are also confidence-weighted averages. We define voxel confidence as $\kappa_v = \min_{l\in\mathcal{P}_v} \kappa_l$ to reduce the impact of unreliable points. A 3D U-Net refines the voxel features for scene understanding. Sparse convolutions Contributors (2022) avoid computation in empty voxels.

To increase the representational capacity of each voxel, we decode every voxel feature $\mathbf{f}_v$ into n Gaussian primitives, allowing one voxel to represent complex local detail. To avoid entangling geometry and appearance during optimization Li et al. (2026), we use separate geometry and appearance branches, $\mathcal{D}_{\text{geo}}$ and $\mathcal{D}_{\text{app}}$. Centers, rotation quaternions, scales, and opacities of n Gaussian primitives are decoded by $\mathcal{D}_{\text{geo}}$. Second-order spherical-harmonic coefficients are decoded by $\mathcal{D}_{\text{app}}$.

3.2.2 STOCHASTIC SOLID VOLUME RENDERING

To further improve the geometric accuracy of NVS, we use stochastic solid volume renderingMiller et al. (2023) instead of vanilla 3DGS rasterization. With an appropriate vacancy function v, volume rendering of the stochastic solid is equivalent to 3DGS rasterization Zhang et al. (2026), as shown in Equation 2, where $\boldsymbol{\omega}$ is the ray direction, $\sigma(\mathbf{x}, \boldsymbol{\omega})$ is the stochastic solid attenuation coefficient at point $\mathbf{x}$ along the ray, $T(t)$ is accumulated transmittance at position t, and $\mathbf{c}$ is the directional color.

$$\begin{gathered}\sigma(\mathbf{x}, \boldsymbol{\omega}) = \left|\boldsymbol{\omega}^\top \nabla \log v(\mathbf{x})\right| = \frac{\left|\boldsymbol{\omega}^\top \nabla v(\mathbf{x})\right|}{v(\mathbf{x})} \qquad T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{x}(s), \boldsymbol{\omega})\, ds\right) \\ \mathbf{C} = \int_{t_n}^{t_f} T(t)\, \sigma(\mathbf{x}(t), \boldsymbol{\omega})\, \mathbf{c}(\mathbf{x}(t), \boldsymbol{\omega})\, dt\end{gathered} \tag{2}$$

An appropriate vacancy v is required to obtain images equivalent to 3DGS rasterization. GGGSZhang et al. (2026) proves that the renderings are equivalent when $v(\mathbf{x}) = \sqrt{1 - G(\mathbf{x})}$.

The accumulated transmittance is then computed as shown in Equation 3, where t_i^* is the location of the Gaussian maximum along the ray.

$$T_i(t) = \begin{cases} \sqrt{1 - G_i(t)}, & t \le t_i^*, \\ \dfrac{1 - G_i(t_i^*)}{\sqrt{1 - G_i(t)}}, & t > t_i^*. \end{cases} \tag{3}$$

For depth rendering, we use the median depth, defined by the point where $T(t_{\text{med}}) = 0.5$ with a fixed number of bisection iterations. During backpropagation, we differentiate the fixed condition $T(t_{\text{med}}) = 0.5$ with respect to Gaussian parameters θ, as shown in Equation 4.

$$\frac{\partial T}{\partial t}\frac{\partial t_{\text{med}}}{\partial \theta} + \frac{\partial T}{\partial \theta} = 0 \tag{4}$$

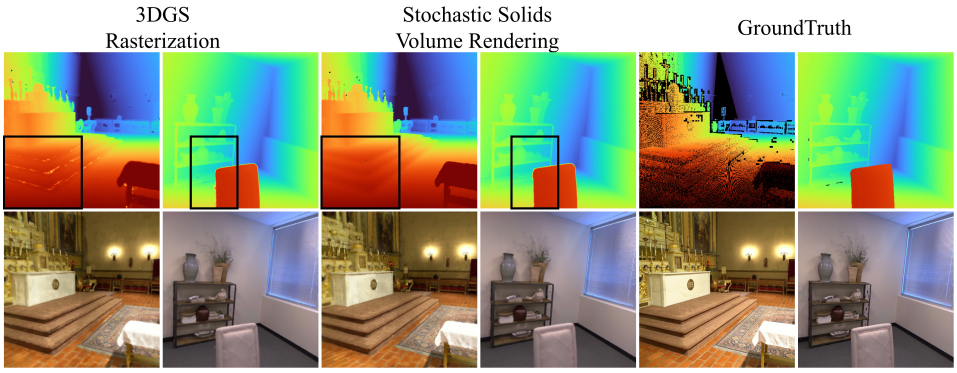

Figure 3: Geometric advantages of stochastic solid volume rendering over 3DGS rasterization.

As shown in Figure 3, stochastic solid volume rendering produces RGB images that are nearly identical to 3DGS rasterization for the same Gaussian scene. For depth rendering, however, it offers a clear geometric advantage by reducing holes and blurred boundaries.

3.3 Test-Time Optimization for VFMs

For reconstruction from posed images, some VFMs encode camera parameters with an MLP and inject them directly into the Transformer. Although this approach improves geometric accuracy, the improvement is insufficient for test views NVS. As shown in Figure 4, AnySplat maintains reasonable input view quality despite artifacts, but produces misalignment in test views.

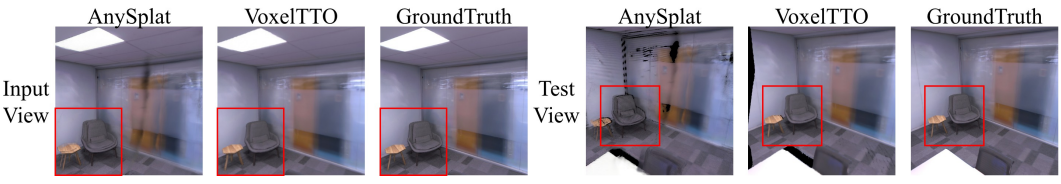

Figure 4: Test-view misalignment after injecting camera parameters.

To address this issue, we introduce LoRA Hu et al. (2021) into the VFM backbone and perform TTO. Following the DA3 architecture Lin et al. (2025), our network has 40 attention layers. Layers 1–12 use single-view attention, whereas layers 13–40 alternate between single-view and global attention. We add LoRA to the $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ projections in layers 13–40. Each layer shares a down-projection matrix $\mathbf{A}$, while their up-projection matrices $\mathbf{B}$ are independent.

When aligned camera parameters are provided, we compute the optional TTO loss after camera decoding and update the LoRA parameters by backpropagation. After fixed iterations, the backbone produces more accurate camera parameters aligned with groundtruth cameras. VGGT-style VFMs usually use the first image as the scene coordinate frame, and image-based reconstruction has scale ambiguity. We therefore align poses before calculating the loss. We fix the first ground-truth camera

pose to the identity matrix. The rotation loss is constructed from the relative rotation between predicted rotations $\mathbf{R}_i^p$ and ground-truth rotations $\mathbf{R}_i^g$, as shown in Equation 5.

$$\mathcal{L}_{\text{rot}} = \frac{1}{2M} \sum_{i=1}^{M} \left[3 - \text{tr}\Big((\mathbf{R}_i^p)^\top \mathbf{R}_i^g \Big) \right]. \tag{5}$$

To address scale ambiguity, we normalize each translation set by mean scale $s = M^{-1} \sum_{i=1}^{M} \|\mathbf{t}_i\|_2$, computed separately for the predicted and ground-truth translations. The scale-normalized translation loss is then calculated by SmoothL1. We also supervise camera intrinsics. The VFM parameterizes intrinsics as $k = (f_x, f_y, c_x, c_y)$, and we directly apply SmoothL1 to the predicted and ground-truth values. The camera-based TTO objective is $L_{\text{TTO}} = \lambda_{\text{pose}} \left(L_{\text{rot}} + w_t L_{\text{trans}} \right) + \lambda_K L_{\text{intrinsics}}$.

3.4 Training

3.4.1 Datasets

We train VoxelTTO on ScanNet Dai et al. (2017), Infinigen Raistrick et al. (2023), ARKitScenes Baruch et al. (2021), and DL3DV-10K Ling et al. (2024). These datasets span indoor and outdoor environments as well as synthetic and real-world scenes. All datasets contain sequential frames. For each dataset, we define a minimum and maximum frame gap, randomly select the interval between the first and last frame within that range, and sample frames from it.

3.4.2 Training Details

Our architecture follows DA3-GIANT, uses $L = 40$ alternating-attention layers as described in subsection 3.3, and is initialized with pretrained weights Lin et al. (2025). The full model has 1.39B parameters, of only 75.3M parameters are trainable We optimize for 30K iterations with Adam, an initial lr of 3×10^{-6}, cosine annealing, 2K warm-up iterations, and a minimum lr of 10^{-8}. Each sample contains 3–16 frames resized to 224×448. Training takes 40 hours on 2 RTX PRO 6000.

As the VFM remains frozen, only the voxel-aligned feed-forward 3DGS module is optimized. The loss contains rendered-depth loss $\mathcal{L}_{\text{depth}}$ and rendered-RGB loss $\mathcal{L}_{\text{rgb}}$, as shown in Equation 6.

$$\mathcal{L} = \mathcal{L}_{\text{depth}} + \mathcal{L}_{\text{rgb}} = \lambda_{\ell_1^{\text{d}}} \mathcal{L}_{\text{depth}}^{\ell_1} + \lambda_{\text{align}} \mathcal{L}_{\text{depth}}^{\text{align}} + \lambda_{\ell_1} \mathcal{L}_{\text{rgb}}^{\ell_1} + \lambda_{\text{ssim}} \mathcal{L}_{\text{rgb}}^{\text{ssim}} \tag{6}$$

Here, $\mathcal{L}_{\text{depth}}^{\ell_1}$ supervises rendered depth with ground-truth depth, whereas $\mathcal{L}_{\text{depth}}^{\text{align}}$ enforces alignment with VFM-predicted depth. $\mathcal{L}_{\text{rgb}}^{\ell_1}$ and $\mathcal{L}_{\text{rgb}}^{\text{ssim}}$ are the ℓ_1 and SSIM Wang et al. (2004) losses between rendered and ground-truth images. We set $\lambda_{\ell_1^{\text{d}}}$, λ_{align}, λ_{ℓ_1}, and λ_{ssim} to 0.25, 0.25, 0.9, and 1.0.

4 Experiments

4.1 Baselines

We compare VoxelTTO with state-of-the-art (SOTA) baselines from two families: VGGT-style VFMs Wang et al. (2025a) and pixelSplat-style Charatan et al. (2024) feed-forward 3DGS. VFM baselines include AnySplat Jiang et al. (2025), YoNoSplat Ye et al. (2025), DepthAnything3 Lin et al. (2025), and MapAnythingKeetha et al. (2026). All support reconstruction from unposed images. AnySplat and DepthAnything3 support camera-parameter encoding. The initialization of vanilla 3DGS is used on MapAnything. Kerbl et al. (2023). pixelSplat-style methods typically require camera parameters; we evaluate VolSplat Wang et al. (2025b) and MVSplat Chen et al. (2024). Experiments are conducted on the Replica Straub et al. (2019), Tanks and Temples Knapitsch et al. (2017), and DTU Aanæ s et al. (2016) datasets, including indoor and outdoor, synthetic and real-world, scene-level and object-level scenarios.

4.2 RGB Novel-View Synthesis

Following prior worksKerbl et al. (2023), we compare RGB NVS performance across VoxelTTO and the baselines. We use PSNR (P), SSIM (S) Wang et al. (2004), and LPIPS (L) Zhang et al. (2018). As shown in Table 1 and Figure 5, VoxelTTO performs strongly in both input and test views. TTO reduces pose-estimation error from the VFM backbone and substantially improves test-view NVS.

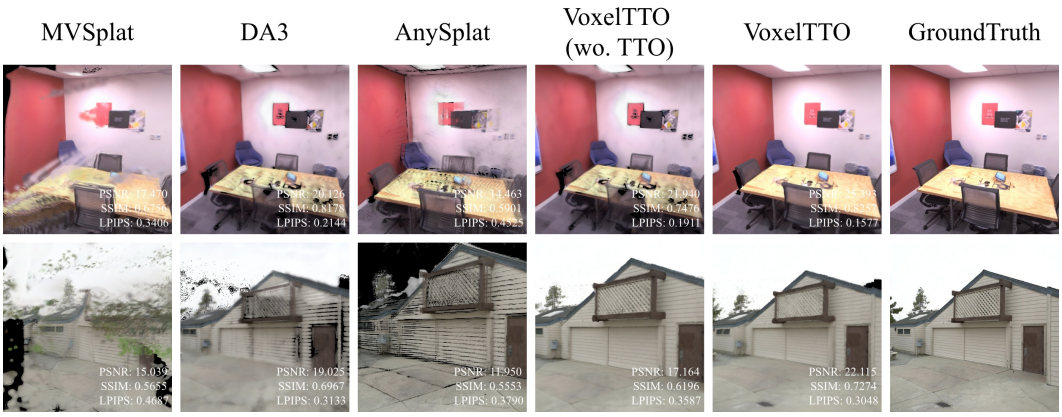

Figure 5: Qualitative RGB novel-view synthesis results for test views.

Table 1: Quantitative RGB novel-view synthesis results for input and test views.

Method	*Replica Dataset*						*TAT Dataset*					
	Input Views			Test Views			Input Views			Test Views		
	P↑	S↑	L↓	P↑	S↑	L↓	P↑	S↑	L↓	P↑	S↑	L↓
VolSplat	16.15	0.667	0.471	15.44	0.636	0.467	15.45	0.595	0.419	16.33	**0.733**	**0.241**
MVSplat	19.97	0.759	0.318	19.41	0.736	0.319	16.48	0.615	0.380	15.05	0.522	0.460
AnySplat	23.25	0.825	0.200	17.78	0.681	0.359	17.34	0.645	0.311	12.71	0.440	0.478
YoNoSplat	24.52	0.809	0.262	11.57	0.635	0.432	13.61	0.477	0.516	13.29	0.660	0.304
DepthAnything3	23.64	0.837	0.208	22.82	0.817	0.243	19.20	0.621	0.348	17.68	0.556	0.413
MapAnything	14.35	0.579	0.533	17.95	0.757	0.227	12.66	0.497	0.570	14.66	0.592	0.381
VoxelTTO (w/o TTO)	29.28	0.890	0.174	25.83	0.827	0.196	**23.74**	**0.767**	**0.247**	14.71	0.485	0.475
Ours	**29.91**	**0.896**	**0.170**	**28.54**	**0.876**	**0.172**	20.89	0.656	0.347	**19.23**	0.565	0.391

4.3 Depth Novel-View Synthesis

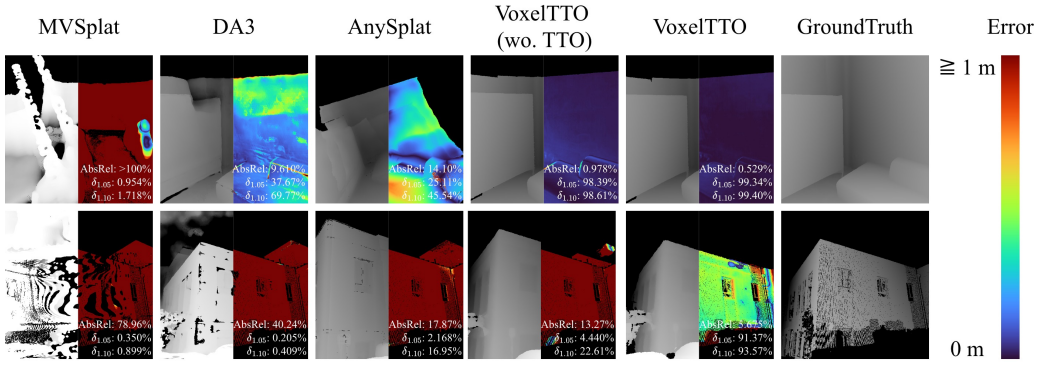

Figure 6: Qualitative depth novel-view synthesis results for test views.

To assess the geometric accuracy, we compare depth NVS performance with each baseline. We report the absolute error RMSE (cm), relative error Rel, and depth accuracy δ_τ at several thresholds. Relative error (Rel, %) is adapted for relative depth error: $\mathrm{Rel} = (1/N)\sum_{i\in\mathcal{P}}(|\hat{d}_i - d_i|/d_i)$. Depth accuracy δ_τ (%) is the proportion of pixels satisfying $\max(\hat{d}_i/d_i, d_i/\hat{d}_i) < \tau$. As shown in Figure 6 and Table 2, VoxelTTO's geometric accuracy is higher than the baselines in both input and test views. TTO mitigates the misalignment caused by VFM and substantially improves test-view geometry.

4.4 Pose Estimation

We use absolute translation error and rotation error to evaluate the poses. Translation error is the Euclidean distance between predicted and ground truth. Rotation error is the geodesic distance on the rotation manifold between prediction and ground truth: $e_R = \arccos\left((\mathrm{tr}\left(\mathbf{R}_{\mathrm{pred}}^{\top}\mathbf{R}_{\mathrm{gt}}\right) - 1)/2\right)$.

Table 2: Quantitative depth novel-view synthesis results for input and test views.

Method	Input Views					Test Views				
	RMSE↓	Rel↓	$\delta_{1.05}$↑	$\delta_{1.10}$↑	$\delta_{1.25}$↑	RMSE↓	Rel↓	$\delta_{1.05}$↑	$\delta_{1.10}$↑	$\delta_{1.25}$↑
Replica Dataset										
VolSplat	102.8	30.42	11.88	22.55	46.78	202.2	63.89	0.812	2.443	6.859
MVSplat	216.0	69.26	1.984	3.894	9.261	1649	446.1	11.84	25.77	42.37
AnySplat	33.72	9.582	59.05	76.88	86.14	35.92	10.78	45.43	67.09	89.86
YoNoSplat	23.47	7.281	51.07	75.58	94.78	242.1	78.73	0.304	0.692	2.643
DepthAnything3	125.5	43.27	28.30	46.40	57.85	172.0	38.69	64.18	71.19	76.29
MapAnything	13.30	3.559	82.69	93.74	98.12	37.96	10.15	44.80	68.37	86.92
VoxelTTO (w/o TTO)	7.394	0.777	98.57	99.22	99.66	9.032	1.115	98.14	98.83	99.38
Ours	**6.167**	**0.688**	**98.66**	**99.29**	**99.72**	**6.945**	**0.866**	**98.86**	**99.24**	**99.58**
TAT Dataset										
VolSplat	490.7	33.73	9.984	19.13	41.25	69.15	31.05	9.058	18.01	42.70
MVSplat	894.3	71.98	0.911	1.809	4.739	143.7	76.57	0.343	0.699	2.203
AnySplat	117.4	4.793	70.01	90.98	96.95	499.1	360.3	5.405	15.37	30.40
YoNoSplat	511.5	37.06	8.018	15.69	33.54	83.57	47.59	7.480	15.38	38.70
DepthAnything3	456.9	41.44	18.76	36.71	60.34	299.4	30.49	3.934	8.181	23.29
MapAnything	139.9	6.560	67.01	81.11	94.77	512.4	399.1	15.28	28.06	31.31
VoxelTTO (w/o TTO)	**82.07**	**2.079**	**96.35**	**97.45**	**98.49**	63.93	7.035	52.39	78.84	95.44
Ours	90.63	2.875	90.12	95.83	98.15	**46.22**	**3.235**	**86.75**	**95.16**	**98.33**

Table 3: Quantitative camera-pose estimation results.

Method	Replica				TAT			
	Rot (°)		Trans (m)		Rot (°)		Trans (m)	
	Mean↓	Max↓	Mean↓	Max↓	Mean↓	Max↓	Mean↓	Max↓
AnySplat	7.320	12.77	0.106	0.470	3.516	4.871	0.135	0.501
YoNoSplat	4.769	6.172	0.063	0.174	5.318	6.593	0.115	0.378
DepthAnything3	0.324	0.549	0.004	0.011	0.682	1.009	0.030	0.084
MapAnything	4.438	5.544	0.056	0.106	3.232	3.649	0.065	0.117
VoxelTTO (w/o TTO)	0.301	0.494	0.006	0.014	1.636	2.040	0.047	0.145
Ours	**0.220**	**0.316**	**0.002**	**0.006**	**0.262**	**0.356**	**0.020**	**0.057**

Table 4: Chamfer distances for surface alignment on DTU.

Method	**Acc.↓**	**Com.↓**	**Over.↓**
VoxelTTO (w/o TTO)	5.556	9.159	7.357
VoxelTTO (Camera Enc.)	3.719	4.853	4.286
VoxelTTO	**2.952**	**3.880**	**3.416**

As shown in Table 3, VoxelTTO freezes both the VFM backbone and pose decoder. Therefore, removing TTO yields pose-estimation accuracy comparable to DA3. The DA3 result in Table 3 uses encoded camera parameters and is consequently slightly improved. Adding TTO substantially improves alignment between predicted and ground-truth poses.

4.5 Surface Alignment

As coupled errors in VFM poses and depth maps can displace the predicted point cloud from the ground-truth surface, further validation is carried on DTU. We compare VoxelTTO without TTO, VoxelTTO with DA3 camera-parameter encoding Lin et al. (2025) instead of TTO, and the full VoxelTTO model. Following DTUAanæ s et al. (2016), we report accuracy (Acc.), completeness (Com.), and their mean (Over.), which are all computed with Chamfer distance. As shown in Table 4, camera-parameter encoding suppresses misalignment to some extent. VoxelTTO further reduces accuracy, completeness, and overall distance.

4.6 Ablation Studies

All the ablation studies are shown in Table 5. For the self-splitting Gaussian decoder, we evaluate the **number of Gaussian splits** per voxel. Increasing the number of splits improves geometry and appearance, but the improvement from 2 to 4 splits is no longer substantial. To evaluate the effect of **stochastic solid volume rendering** on geometry, we compare it with 3DGS rasterization during both training and inference. Using 3DGS rasterization substantially reduces appearance and geometric accuracy under both stages. Here, $\mathbb{V}$-$\mathbb{R}$ denotes stochastic solid volume-rendering supervision ($\mathbb{V}$) during training and 3DGS rasterization ($\mathbb{R}$) for NVS at inference. We compare different voxel sizes in voxel-aligned 3DGS. Both appearance and geometric performance decline as **voxel size** increases. Balancing performance and computation, and following prior work Jiang et al. (2025), we set the size to 0.002m. An RGB loss $\mathcal{L}_{\text{rgb}}$ is required to regress color. We therefore ablate the **depth loss**

Table 5: Ablation studies on test-view RGB-D NVS under Replica dataset.

Experiment	Setting	PSNR↑	SSIM↑	LPIPS↓	RMSE (cm)↓	Rel↓	$\delta_{1.05}$↑	$\delta_{1.10}$↑	$\delta_{1.25}$↑
Number of Gaussian Splits	1	27.82	0.866	**0.172**	8.09	0.94	98.71	99.10	99.47
	2 (Ours)	**28.54**	0.876	0.172	**6.95**	0.87	98.87	99.24	**99.59**
	4	28.35	**0.876**	0.179	7.21	**0.86**	**98.93**	**99.24**	99.54
Stochastic Solid Volume Rendering	$\mathbb{V}$-$\mathbb{V}$ (Ours)	**28.54**	**0.876**	**0.172**	**6.95**	**0.87**	**98.87**	**99.24**	**99.59**
	$\mathbb{V}$-$\mathbb{R}$	27.11	0.865	0.188	12.01	1.24	97.96	98.62	99.15
	$\mathbb{R}$-$\mathbb{R}$	21.52	0.774	0.269	27.58	4.05	87.76	95.11	97.10
	$\mathbb{R}$-$\mathbb{V}$	22.37	0.798	0.260	15.82	3.04	89.32	97.03	98.66
Voxel Size	0.0015	**29.34**	**0.891**	**0.142**	**6.62**	**0.83**	**99.04**	**99.36**	**99.64**
	0.002 (Ours)	28.54	0.876	0.172	6.95	0.87	98.87	99.24	99.59
	0.003	26.78	0.856	0.211	9.45	1.18	98.32	98.88	99.32
	0.005	23.44	0.798	0.290	15.22	2.45	93.17	97.15	98.47
Depth Loss	Ours	**28.54**	**0.876**	0.172	**6.95**	**0.87**	**98.87**	**99.24**	**99.59**
	No $\mathcal{L}_{\text{depth}}$	28.25	0.873	**0.171**	7.49	0.94	98.67	99.10	99.53

separately. Removing the depth loss slightly reduces geometric accuracy. We further evaluate the effect of **TTO iterations**. As shown in Figure 7, test-view appearance performance (LPIPS) and geometric performance (Rel) improve substantially as the number of TTO iterations increases. We select 20 iterations by balancing accuracy and efficiency.

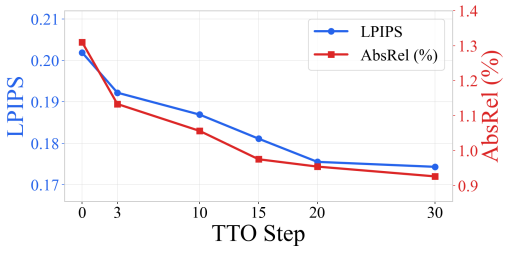

Figure 7: Test-view RGB-D synthesis performance at different TTO iteration counts.

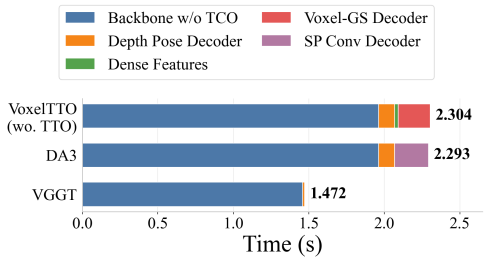

Figure 8: Runtime of individual modules in VoxelTTO without TTO and in the baselines.

4.7 Runtime

For ten 518×518 images, VoxelTTO without TTO takes 2.304 s, compared with 2.293 s for DA3, indicating negligible overhead from voxel decoding (Figure 8). We retain frozen attention computations in BF16 and LoRA computations in Float32, while baselines use their official precision. BF16 reduces backbone runtime from 1.959 s to 0.1956 s. Each TTO step requires 0.2662 s forward and 0.075 s backward, so 20 steps complete in under 7 s and total runtime is under 7.4 s.

5 Conclusion

We presented VoxelTTO, a voxel-aligned feed-forward 3DGS framework with test-time optimization. First, it reduces the excessive overlap and artifacts of pixel-aligned feed-forward 3DGS by aggregating dense pixel features into global voxel features. A sparse 3D voxel U-Net and self-splitting decoder regress Gaussian attributes. Stochastic solid volume rendering replaces 3DGS rasterization during training and inference, improving the geometric accuracy of the Gaussian scene. Second, pose-supervised TTO adapts lightweight LoRA modules in the VFM backbone to align predicted camera parameters with the provided camera priors. Extensive experiments on DTU, Replica, and T&T show that VoxelTTO reconstructs geometrically accurate 3DGS scenes and achieves SOTA test-view RGB and depth rendering accuracy.

AI USE STATEMENT

In this work, we did not use generative AI tools for any tasks requiring disclosure under the ICLR 2027 AI Policy for Authors. We used generative AI tools solely for language editing and improving the readability of the manuscript; create or edit software code. We have reviewed all AI-assisted work. All AI-assisted edits were carefully reviewed by the authors to ensure correctness and to preserve the intended technical meaning. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The train dataset can be found in:

ScanNetDai et al. (2017)https://scannetpp.mlsg.cit.tum.de/scannetpp; InfinigenRaistrick et al. (2023)https://github.com/princeton-vl/infinigen; ARKitScenesBaruch et al. (2021)https://github.com/apple-aiml-research/ARKitScenes; DL3DVLing et al. (2024)https://dl3dv-10k.github.io/DL3DV-10K/

The eval dataset can be found in

ReplicaStraub et al. (2019)https://github.com/facebookresearch/replica-dataset; Tanks and TemplesKnapitsch et al. (2017)https://www.tanksandtemples.org/; DTUAanæ s et al. (2016)http://roboimagedata.compute.dtu.dk/?page_id=36

The code can be found in

https://anonymous.4open.science/r/VoxelTTO-E9D8 and https://anonymous.4open.science/r/Geometry-Grounded-Gaussian-Splatting-1B46

REFERENCES

Henrik Aanæ s, Rasmus R. Jensen, George Vogiatzis, Engin Tola, and Anders B. Dahl. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*, 120(2):153–168, 2016. doi: 10.1007/s11263-016-0902-9.

Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, et al. Arkitscenes–a diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. *arXiv preprint arXiv:2111.08897*, 2021.

David Charatan, Sizhe Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *CVPR*, 2024.

Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. *arXiv preprint arXiv:2403.14627*, 2024.

Spconv Contributors. Spconv: Spatially sparse convolution library. https://github.com/traveller59/spconv, 2022.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

Pinxuan Dai, Jiamin Xu, Wenxiang Xie, Xinguo Liu, Huamin Wang, and Weiwei Xu. High-quality surface reconstruction using gaussian surfels. In *ACM SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024.

Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 224–236, 2018.

Sven Elflein, Ruilong Li, Sérgio Agostinho, Zan Gojcic, Laura Leal-Taixé, Qunjie Zhou, and Aljosa Osep. Vgg-t^3: Offline feed-forward 3d reconstruction at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Yang Fu, Sifei Liu, Amey Kulkarni, Jan Kautz, Alexei A. Efros, and Xiaolong Wang. Colmap-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20796–20805, June 2024.

Diego Gomez, Antoine Guédon, Nissim Maruani, Bingchen Gong, and Maks Ovsjanikov. From blobs to spokes: High-fidelity surface reconstruction via oriented gaussians, 2026. URL https://arxiv.org/abs/2604.07337.

Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. *CVPR*, 2024.

Seongbo Ha, Jiung Yeon, and Hyeonwoo Yu. Rgbd gs-icp slam. In *European conference on computer vision*, pp. 180–197. Springer, 2024.

Dongchen Han, Yining Li, Tianyu Li, Zixuan Cao, Ziming Wang, Jun Song, Yu Cheng, Bo Zheng, and Gao Huang. Vit^3: Unlocking test-time training in vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

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

Han Huang, Yulun Wu, Chao Deng, Ge Gao, Ming Gu, and Yu-Shen Liu. Fatesgs: Fast and accurate sparse-view surface reconstruction using gaussian splatting with depth-feature consistency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, et al. Anysplat: Feed-forward 3d gaussian splatting from unconstrained views. *ACM Transactions on Graphics (TOG)*, 44(6):1–16, 2025.

Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. Splatam: Splat track & map 3d gaussians for dense rgb-d slam. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 21357–21366, 2024.

Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, Jonathon Luiten, Manuel Lopez-Antequera, Samuel Rota Bulò, Christian Richardt, Deva Ramanan, Sebastian Scherer, and Peter Kontschieder. MapAnything: Universal feed-forward metric 3D reconstruction. In *International Conference on 3D Vision (3DV)*. IEEE, 2026.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. URL https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/.

Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017.

Yinglong Li, Donghui Shen, Xiaoyu Zhang, Zhichao Ye, Hongyu Wu, Aimin Hao, Guofeng Zhang, and Haomin Liu. Querysplat: Decoupling geometry and appearance representations in 3dgs prediction. *arXiv preprint arXiv:2608.01186*, 2026.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22160–22169. IEEE, 2024.

Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20654–20664, 2024.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

Bailey Miller, Hanyu Chen, Alice Lai, and Ioannis Gkioulekas. Objects as volumes: A stochastic geometry view of opaque solids. *IEEE*, 2023.

Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

Alexander Raistrick, Lahav Lipson, Zeyu Ma, Lingjie Mei, Mingzhe Wang, Yiming Zuo, Karhan Kayan, Hongyu Wen, Beining Han, Yihan Wang, Alejandro Newell, Hei Law, Ankit Goyal, Kaiyu Yang, and Jia Deng. Infinite photorealistic worlds using procedural generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12630–12641, 2023.

Jiawei Ren, Michal Tyszkiewicz, Jiahui Huang, and Zan Gojcic. Tokengs: Decoupling 3d gaussian prediction from pixels with learnable tokens. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4938–4947, 2020.

Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 13(4):376–380, 1991.

Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. Vggsfm: Visual geometry grounded deep structure from motion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 21686–21697, 2024a.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025a.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. Vggt-ω, 2026. URL https://arxiv.org/abs/2605.15195.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, 2024b.

Weijie Wang, Yeqing Chen, Zeyu Zhang, Hengyu Liu, Haoxiao Wang, Zhiyuan Feng, Wenkang Qin, Feng Chen, Zheng Zhu, Donny Y. Chen, and Bohan Zhuang. Volsplat: Rethinking feed-forward 3d gaussian splatting with voxel-aligned prediction. *arXiv preprint arXiv:2509.19297*, 2025b.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-Equivariant Visual Geometry Learning. *arXiv preprint arXiv:2507.13347*, 2025c.

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.

Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Depthsplat: Connecting gaussian splatting and depth. In *CVPR*, 2025.

Botao Ye, Boqi Chen, Haofei Xu, Daniel Barath, and Marc Pollefeys. Yonosplat: You only need one model for feedforward 3d gaussian splatting, 2025.

Baowen Zhang, Chenxing Jiang, Heng Li, Shaojie Shen, and Ping Tan. Geometry-grounded gaussian splatting. *arXiv preprint arXiv:2601.17835*, 2026.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

Lei Zhou, Haoyu Wu, Akshat Dave, and Dimitris Samaras. Learning 3d reconstruction with priors in test time. *arXiv preprint arXiv:2604.03878*, 2026.