DASAM3D: A Unified Foundation Model Framework for Enhanced 3D Scene Reconstruction and Segmentation

Ronny Velastegui[1], Sezer Karaoglu[1], Max Pfingsthorn[2], and Theo Gevers[1]

[1] University of Amsterdam, Amsterdam, The Netherlands
{r.x.velasteguisandoval,s.karaoglu,th.gevers}@uva.nl
[2] Robert Bosch GmbH, Renningen, Germany
max.pfingsthorn@de.bosch.com

Abstract. We present DASAM3D, a framework that fuses the complementary strengths of SAM 3D and Depth Anything 3 (DA3) for object-aware 3D scene reconstruction. A fundamental *appearance–geometry trade-off* exists: SAM 3D yields geometrically coherent per-object Gaussian primitives but with unrealistic appearance, while DA3 delivers photorealistic 3DGS reconstructions yet lacks object-level geometric reasoning. Our key insight is to use SAM 3D exclusively as a *3D geometric layout provider*, guiding DA3 geometry refinement while preserving its photorealistic textures. DASAM3D proceeds through three differentiable stages—**(1)** Global Alignment Optimization, **(2)** Per-Object Alignment Refinement, and **(3)** Layout-Based Geometry Optimization—and inherits DA3's flexibility for posed or pose-free inputs at any scale. Experiments on DL3DV-10K and MipNeRF-360 show consistent gains in reconstruction accuracy and novel view synthesis; a user-preference study confirms DASAM3D's object renderings are preferred over SAM 3D's in visual fidelity.

Keywords: 3D Gaussian Splatting · Foundation Models · Scene Reconstruction · Object-Aware Reconstruction · Neural Rendering

1 Introduction

High-fidelity 3D scene reconstruction has advanced dramatically, from Neural Radiance Fields [1–3,29,30] to 3D Gaussian Splatting (3DGS) [20], which represents scenes as differentiable anisotropic Gaussian primitives enabling real-time photorealistic rendering. A new generation of *foundation models* operates flexibly with or without known camera poses. DUSt3R [36] and MASt3R [24] predict pairwise 3D point maps from uncalibrated images. Fast3R [41] scales this to thousands of images in a single feed-forward pass. VGGT [35] grounds 3D structure estimation via transformer attention. Pi3 [37] introduces a permutation-equivariant architecture that predicts affine-invariant poses and scale-invariant point maps without a fixed reference view. MapAnything [19] is a unified feed-forward model for universal metric 3D reconstruction from images and optional geometric inputs.

Depth Anything 3 (DA3) [25] reconstructs full 3DGS scenes via metric depth and camera pose prediction, accepting either posed or unposed inputs and scaling to any number of views. Feed-forward methods pixelSplat [6] and MVSplat [10] directly predict 3DGS from a small number of calibrated images. SAM 3D [7] is a generative foundation model that reconstructs per-object 3D geometry, texture, and layout from single images, producing outputs in Gaussian splat format with geometric precision.

Despite these advances, a fundamental *appearance–geometry trade-off* persists. **DA3** produces photorealistic but geometrically inaccurate reconstructions—floating Gaussians, bumpy surfaces, inconsistent normals—because its objective is purely photometric with no object-level structural prior. **SAM 3D** yields geometrically coherent, object-isolated Gaussian distributions but produces objects with unrealistic appearance that does not faithfully reflect the colors, textures, and visual properties of the corresponding objects in the input images, making the reconstructed objects unsuitable for photorealistic novel view synthesis. This limitation is shared across the broader 3DGS ecosystem, including Mip-Splatting [46], Scaffold-GS [27], 2DGS [17], GaussianPro [11], and InstantSplat [13], all of which improve photometric quality without incorporating object-level 3D geometric structure.

We propose DASAM3D, a three-stage differentiable pipeline that treats SAM 3D's Gaussian clouds as a *3D geometric layout provider* for DA3: **(1) Global Alignment Optimization** — all SAM 3D objects are merged and registered to DA3's coordinate frame using all real training cameras with per-view affine depth normalization; **(2) Per-Object Alignment Refinement** — each object is independently refined using cameras filtered to object-visible views, combined with a loss to the local DA3 cloud and a bounding-box constraint, followed by quality-based filtering; **(3) Layout-Based Geometry Optimization** — the filtered SAM 3D layout defines adaptive volumetric ROIs, and three complementary losses refine DA3 geometry.

Contributions:

- A framework for refining a 3D Gaussian scene produced by a vision foundation model (DA3) by leveraging the Gaussian output of a second vision foundation model (SAM 3D) as a geometric layout prior—operating entirely in Gaussian parameter space without modifying the learned parameters of either model.
- A strategy for combining the complementary strengths of two vision foundation models: the geometric accuracy of SAM 3D's object-level Gaussian layouts with the photorealistic appearance of DA3, yielding a unified reconstruction that benefits from both properties simultaneously.
- A global-to-local alignment pipeline (Global Alignment Optimization followed by Per-Object Alignment Refinement) driven entirely by real training cameras, combining affine-normalized depth supervision with per-object 3D Chamfer matching and a bounding-box constraint.
- A quality scoring and filtering step (Equation (6)) that evaluates 2D overlap, depth residual, and 3D proximity to gate which objects contribute geometry priors to Stage 3.

– Evaluation on MipNeRF-360 scenes (3D reconstruction accuracy and NVS) and DL3DV-10K (NVS); a user-preference study confirming that DASAM3D produces more visually realistic object reconstructions than SAM 3D alone.

2 Related Work

Pose-Free and Multi-View 3D Reconstruction. DUSt3R [36] and MASt3R [24] predict pairwise 3D point maps and lift them to globally consistent reconstructions, establishing the foundation for pose-free 3D. Fast3R [41] scales this paradigm to 1000+ images in a single feed-forward pass. VGGT [35] introduces visual geometry grounding for accurate structural prediction. Pi3 [37] proposes a permutation-equivariant design that eliminates the fixed reference-view bias of prior methods. MapAnything [19] introduces a unified transformer for universal metric 3D reconstruction, supporting over twelve different 3D vision tasks in a single feed-forward pass. MonST3R [48] extends DUSt3R to dynamic video. DA3 [25] produces full 3DGS reconstructions from posed or unposed images, supporting any number of input views. Feed-forward methods pixelSplat [6] and MVSplat [10] directly regress 3DGS from a fixed set of calibrated images. DASAM3D builds on DA3 as its primary backbone.

3D Gaussian Splatting Extensions. 3DGS [20] enables real-time rendering with differentiable anisotropic Gaussian primitives. Extensions improve compactness [8, 14], neural structure [27], anti-aliasing [46], surface geometry [17, 47], progressive propagation [11], and pose-free reconstruction [13]. Hash-grid scene encoding [30] further accelerates volumetric representations. All these methods optimize photometric quality without object-level 3D geometric priors; DASAM3D provides this missing component.

Depth and Normal Priors for 3DGS. DN-Splatter [34] supervises 3DGS with monocular depth and normal predictions. GaussianRoom [40] adds SDF guidance; MonoGS [28] uses monocular cues for SLAM; GSRec [39] introduces 2D depth hints for covariance regularization. Depth Anything V2 [42] provides strong monocular depth priors for downstream tasks. Unlike these 2D-prior methods, DASAM3D leverages a full volumetric 3D layout from SAM 3D, enabling volumetric ROI definition and 3D Chamfer supervision that are complicated from a depth image alone.

Segment Anything in 3D. SAM [23] and SAM 2 [33] provide promptable 2D and video segmentation. Gaussian Grouping [43], Segment Any 3D Gaussians [5], and OmniSeg3D [45] lift SAM masks into 3DGS. LangSplat [32] and LERF [21] embed language into radiance fields. SAM 3D [7] is a generative model for visually grounded 3D object reconstruction from single images, predicting geometry, texture, and layout with outputs decodable as Gaussian splats; however, its per-object outputs exhibit unrealistic appearance that does not reflect the input image faithfully. DASAM3D uses SAM 3D's Gaussian output exclusively

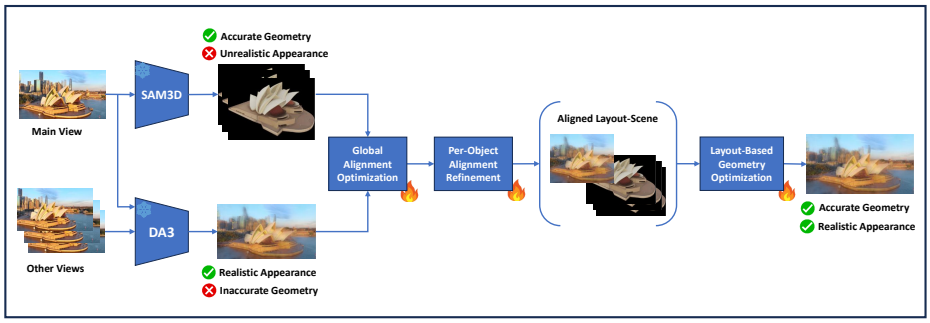

Fig. 1: DASAM3D pipeline overview. Three stages progressively process independently reconstructed DA3 and SAM 3D scenes: **(1)** *Global Alignment Optimization*: affine-normalized depth registration of all SAM 3D objects to DA3; **(2)** *Per-Object Alignment Refinement*: mixed 2D–3D supervision with quality-based filtering; **(3)** *Layout-Based Geometry Optimization*: ROI-masked DA3 geometry refinement. The output combines DA3's photorealistic appearance with SAM 3D's geometric layout.

as a volumetric 3D layout prior, preserving DA3's photorealistic appearance throughout.

Point Cloud Registration and 3D Matching. Classical ICP [4] and feature-metric registration [18] align point clouds via the Chamfer distance [12]. DASAM3D applies a one-way Chamfer term in Gaussian parameter space for cross-model 3DGS alignment under unknown relative pose, scale, and object count.

Scene Composition and Editing. SuGaR [16], GaussianEditor [9], and ConceptGraphs [15] operate on object-level 3DGS; DASAM3D provides geometry-accurate object regions as input to such pipelines.

3 Method: DASAM3D

Given an RGB image sequence, DA3 produces a full-scene Gaussian splat $\mathcal{G}_{\text{DA}}$ and SAM 3D produces per-object clouds $\mathcal{G}_{\text{SAM}} = \{O_k\}_{k=1}^{K}$. DASAM3D registers $\mathcal{G}_{\text{SAM}}$ into DA3's coordinate frame (Stages 1–2) and then refines $\mathcal{G}_{\text{DA}}$ geometry within SAM 3D-defined volumetric regions (Stage 3), without modifying the learned parameters of either foundation model. The full pipeline is illustrated in Figure 1.

3.1 Preliminaries: 3DGS and Rotation Parameterization

A 3DGS scene has N primitives $\mathcal{G} = \{(\boldsymbol{\mu}_i, \mathbf{q}_i, \mathbf{s}_i, o_i, \mathbf{c}_i)\}$ with center $\boldsymbol{\mu}_i \in \mathbb{R}^3$, unit quaternion $\mathbf{q}_i$, per-axis scales $\mathbf{s}_i \in \mathbb{R}^3_{>0}$, opacity o_i, and color $\mathbf{c}_i$. Pixels are rendered via depth-sorted alpha-compositing: $\mathbf{I}(p) = \sum_i \mathbf{c}_i \alpha_i \prod_{j<i}(1-\alpha_j)$, fully differentiable w.r.t. all parameters. We parameterize rotations by an axis-angle

vector $\mathbf{w} \in \mathbb{R}^3$ via the exponential map $\mathbf{R}(\mathbf{w}) = \mathbf{I} + \frac{\sin\theta}{\theta}[\mathbf{w}]_\times + \frac{1-\cos\theta}{\theta^2}[\mathbf{w}]_\times^2$ with $\theta = \|\mathbf{w}\|_2$, which avoids gimbal lock and enables unconstrained gradient descent. Scale is optimized in log-space as $s = \exp(\ell_s)$ to ensure positivity throughout.

3.2 Stage 1: Global Alignment Optimization

Setup. All K SAM 3D objects are merged into a single cloud $\mathcal{G}_{\text{SAM}}$. We optimize a global similarity transform $T = (\mathbf{t}, \mathbf{w}, \ell_s)$ mapping $\mathcal{G}_{\text{SAM}}$ into DA3's coordinate frame. All real training cameras $\{(W_v, K_v)\}_{v=1}^{V}$ predicted by DA3 are used for supervision; no synthetic views are required here.

Transform. Each primitive of $\mathcal{G}_{\text{SAM}}$ is transformed as:

$$\widetilde{\boldsymbol{\mu}}_i = s\,\mathbf{R}(\mathbf{w})\,\boldsymbol{\mu}_i + \mathbf{t}, \quad \widetilde{\mathbf{s}}_i = s\,\mathbf{s}_i, \quad \widetilde{\mathbf{q}}_i = \mathbf{q}(\mathbf{w}) \otimes \mathbf{q}_i. \tag{1}$$

Affine depth normalization. Because DA3 and SAM 3D reconstruct independently, a global scale–offset mismatch exists between their depth maps. For each training view v we compute an affine pre-alignment $(a^{(v)}, b^{(v)})$ between the rendered SAM 3D depth $D_{\text{SAM}}^{(v)}$ and the DA3 reference depth $D_{\text{DA}}^{(v)}$ over co-visible pixels $\mathcal{P}_v = \{p : \alpha_{\text{SAM}}^{(v)}(p) > \epsilon \wedge \alpha_{\text{DA}}^{(v)}(p) > \epsilon\}$ via ordinary least squares in closed form. The normalized target $\hat{D}_{\text{DA}}^{(v)} = a^{(v)} D_{\text{DA}}^{(v)} + b^{(v)}$ is treated as a constant during back-propagation.

Optimization. The total loss is:

$$\mathcal{L}_{\text{s1}} = \frac{1}{|\mathcal{V}|}\sum_v \frac{1}{|\mathcal{P}_v|}\sum_{p\in\mathcal{P}_v}\left(D_{\text{SAM}}^{(v)}(p) - \hat{D}_{\text{DA}}^{(v)}(p)\right)^2 + \lambda_t\,\|\mathbf{t}\|_2^2 + \lambda_r\,\|\mathbf{w}\|_2^2 + \lambda_s\,\ell_s^2 + \mathcal{L}_{\text{bounds}}, \tag{2}$$

where $\mathcal{L}_{\text{bounds}}$ applies soft quadratic penalties outside the allowed parameter ranges. Adam [22, 31]: 500 iterations, lr $= 10^{-2}$, $\lambda_t = 20$, $\lambda_r = 5$, $\lambda_s = 0.1$, $t_{\max} = 1.0\,\text{m}$, $\theta_{\max} = 30°$, $s \in [0.3, 3.0]$. The result is the globally aligned layout $\mathcal{G}_{\text{SAM}}^{(1)}$.

3.3 Stage 2: Per-Object Alignment Refinement

Motivation. Global alignment corrects the inter-model coordinate frame but cannot resolve per-object depth or scale discrepancies from SAM 3D's independent single-object reconstruction. Stage 2 refines each object O_k independently, combining 2D depth supervision at real training cameras with direct 3D spatial grounding via a Chamfer objective [12].

Object-visible view selection. For each O_k we identify training cameras $\mathcal{V}_k = \{v : \bar{\alpha}_k^{(v)} > \tau_\alpha\}$ where $\bar{\alpha}_k^{(v)}$ is the mean rendered alpha of O_k at view v and $\tau_\alpha = 0.005$. If no camera satisfies this, all training cameras serve as a fallback. Since the object is already in DA3's coordinate frame after Stage 1, absolute (non-affine) depth matching is used.

3D Chamfer loss and bounding-box constraint. We extract DA3 means $\mathcal{M}_k^{\mathrm{DA}}$ within an axis-aligned bounding cube of half-extent ηr_k centred on O_k, where r_k is its maximum radius and $\eta = 1.5$. The one-way Chamfer loss pulls object means toward the local DA3 cloud:

$$\mathcal{L}_{\mathrm{cham},k} = \frac{1}{|\mathcal{S}_k|} \sum_{i \in \mathcal{S}_k} \min_{j \in \mathcal{M}_k^{\mathrm{DA}}} \left\| \widetilde{\boldsymbol{\mu}}_i - \boldsymbol{\mu}_j^{\mathrm{DA}} \right\|_2, \tag{3}$$

where $\mathcal{S}_k$ is a stochastic subsample of up to 1000 object means per iteration. A soft bounding-box constraint prevents large drift:

$$\mathcal{L}_{\mathrm{bbox},k} = \mathrm{mean}\big(\mathrm{relu}(\boldsymbol{\mu}_{\min,k} - \widetilde{\boldsymbol{\mu}}) + \mathrm{relu}(\widetilde{\boldsymbol{\mu}} - \boldsymbol{\mu}_{\max,k})\big). \tag{4}$$

The combined per-object loss is:

$$\mathcal{L}_{\mathrm{s2},k} = \mathcal{L}_{\mathrm{depth},k} + \lambda_{\mathrm{3D}}\, \mathcal{L}_{\mathrm{cham},k} + \lambda_{\mathrm{bbox}}\, \mathcal{L}_{\mathrm{bbox},k} + \mathcal{L}_{\mathrm{reg},k}, \tag{5}$$

with 500 iterations, $\lambda_{\mathrm{3D}} = 0.5$, $\lambda_{\mathrm{bbox}} = 20$, loose angular and scale bounds ($\theta_{\max} = 30°$, $s \in [0.5, 2.0]$). Composed transforms yield $\mathcal{G}_{\mathrm{SAM}}^{(2)}$.

Quality scoring and filtering. Poorly segmented or partially visible objects can inject corrupted priors into Stage 3. We compute a composite quality score per object:

$$\mathrm{score}_k = f_{\mathrm{ov}} \cdot (1 - \bar{r}_k) \cdot \exp(-2\, d_k^{\mathrm{norm}}), \tag{6}$$

where f_{ov} is the fraction of cameras where the object is visible, $\bar{r}_k$ is the normalized mean depth residual at co-visible pixels, and d_k^{norm} is the nearest-neighbor distance from the object centroid to the DA3 cloud normalized by the scene scale. Objects passing $f_{\mathrm{ov}} \geq 0.05$ and $\bar{r}_k \leq 0.50$ contribute to Stage 3.

3.4 Stage 3: Layout-Based Geometry Optimization

Adaptive ROI computation. Only DA3 Gaussians within the volumetric footprint of the filtered SAM 3D layout receive gradient updates. For each SAM 3D Gaussian j with mean scale $\bar{s}_j = \frac{1}{3} \sum_d s_{j,d}$, we form a sphere of radius $r_j = \gamma \bar{s}_j$ ($\gamma = 2.0$):

$$\mathcal{R}^{(0)} = \big\{ i : \exists\, j,\ \left\| \boldsymbol{\mu}_i^{\mathrm{DA}} - \widetilde{\boldsymbol{\mu}}_j \right\|_2 \leq r_j \big\}. \tag{7}$$

In a second adaptive pass, each candidate in $\mathcal{R}^{(0)}$ is retained only if its distance to its nearest SAM 3D Gaussian $j^*(i)$ does not exceed $r_{j^*(i)}$: $\mathcal{R} = \{ i \in \mathcal{R}^{(0)} : d_i^{\mathrm{nn}} \leq r_{j^*(i)} \}$. This two-step approach avoids both false positives and false negatives.

Camera setup. Starting from the first real training camera (W_0, K_0) and the centroid $\bar{\boldsymbol{\mu}}_{\mathrm{SAM}}$ of all filtered SAM 3D Gaussians, we build five base views: the real camera plus four synthesized orthogonal views. We linearly interpolate between consecutive base views ($N_{\mathrm{interp}} = 8$ steps per edge) to create candidates, then select $N_{\mathrm{train}} = 5$ views uniformly from those where the SAM 3D alpha coverage exceeds $\tau_{\mathrm{cov}} = 0.01$.

Loss functions. The three-term composite loss is:

$$\mathcal{L}_{\text{s3}} = \lambda_{\text{geo}}\,\mathcal{L}_{\text{depth}} + \lambda_{\text{norm}}\,\mathcal{L}_{\text{norm}} + \lambda_{\text{rgb}}\,\mathcal{L}_{\text{rgb}}, \tag{8}$$

$$\mathcal{L}_{\text{depth}} = \frac{1}{|\mathcal{T}|}\sum_{v\in\mathcal{T}}\frac{1}{|\mathcal{V}_v|}\sum_{p\in\mathcal{V}_v}\left(D^{(v)}_{\text{DA}}(p) - \hat{D}^{(v)}_{\text{SAM}}(p)\right)^2, \tag{9}$$

$$\mathcal{L}_{\text{norm}} = \frac{1}{|\mathcal{T}|}\sum_{v\in\mathcal{T}}\frac{1}{|\mathcal{N}_v|}\sum_{p\in\mathcal{N}_v}\left(1 - \mathbf{n}^{(v)}_{\text{DA}}(p)\cdot\mathbf{n}^{(v)}_{\text{SAM}}(p)\right), \tag{10}$$

$$\mathcal{L}_{\text{rgb}} = \frac{1}{N_p}\sum_{p}\left|\mathbf{I}^{(v_0)}_{\text{full}}(p) - \mathbf{I}^{(v_0)}_{\text{DA,orig}}(p)\right|, \tag{11}$$

where $\hat{D}_{\text{SAM}}$ is affine-normalized SAM 3D depth, $\mathcal{V}_v$ ($\mathcal{N}_v$) are co-visible depth (normal) pixels, surface normals $\mathbf{n}$ are rendered from each Gaussian's minimum-scale axis, $\mathbf{I}^{(v_0)}_{\text{full}}$ is the full-scene rendering at view v_0 (frozen non-ROI $\oplus$ optimized ROI), and $\mathbf{I}^{(v_0)}_{\text{DA,orig}}$ is the pre-Stage-3 DA3 rendering cached at v_0. $\mathcal{L}_{\text{rgb}}$ is applied every 5 iterations to amortize cost. Weights: $\lambda_{\text{geo}}=5.0$, $\lambda_{\text{norm}}=0.01$, $\lambda_{\text{rgb}}=0.5$.

Geometry-only update. Colors are frozen throughout Stage 3 (inherited directly from DA3), preventing geometry updates from corrupting photorealistic textures. Only $\{\boldsymbol{\mu}_i, \mathbf{q}_i, \mathbf{s}_i, o_i\}_{i\in\mathcal{R}}$ are optimized. Stage 3 runs for 300 iterations with lr$=2\times10^{-3}$.

3.5 Implementation

Algorithm 1 summarizes the complete pipeline. All stages use the GSPLAT [44] rasterizer and Adam [22], implemented in PyTorch [31]. Hardware: NVIDIA A100 (40 GB).

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate novel view synthesis on two benchmarks and 3D reconstruction accuracy on one.

DL3DV-10K [26] comprises 10,000 real-world scenes with SfM calibration; we use a 12-scene subset for NVS with held-out test views.

MipNeRF-360 [2] is the standard unbounded scene benchmark covering both indoor and outdoor scenarios. We evaluate on six scenes (Bicycle, Bonsai, Counter, Garden, Kitchen, Room) using the standard 1/8 holdout test split and report both 3D reconstruction accuracy and NVS metrics. For 3D reconstruction evaluation, we use dense point clouds obtained by running COLMAP MVS on the full training image set as a proxy reference surface, following common practice in the absence of sensor-based ground truth.

Algorithm 1 DASAM3D pipeline

Require: DA3 scene $\mathcal{G}_{\text{DA}}$; SAM 3D objects $\{O_k\}_{k=1}^{K}$; training cameras $\{(W_v, K_v)\}_{v=1}^{V}$
Ensure: Refined scene $\hat{\mathcal{G}}_{\text{DA}}$
Merge all $\{O_k\}$ into $\mathcal{G}_{\text{SAM}}$
Compute per-view affine pre-alignment $(a^{(v)}, b^{(v)})$
Optimize $(\mathbf{t}, \mathbf{w}, \ell_s)$ via $\mathcal{L}_{\text{s1}}$ (Equation (2)) over all V cameras
Apply global transform $\rightarrow \mathcal{G}_{\text{SAM}}^{(1)}$
for $k = 1, \ldots, K$ **do**
    Select visible cameras $\mathcal{V}_k$; extract $\mathcal{M}_k^{\text{DA}}$
    Optimize $(\mathbf{t}_k, \mathbf{w}_k, s_k)$ via $\mathcal{L}_{\text{s2},k}$ (Equation (5)): depth + Chamfer + bbox
    Compute score$_k$ (Equation (6)); filter if $f_{\text{ov}} < 0.05$ or $\bar{r}_k > 0.50$
end for
Compose transforms $\rightarrow \mathcal{G}_{\text{SAM}}^{(2)}$ (filtered objects only)
Compute adaptive ROI $\mathcal{R}$ (Equation (7))
Build base views; select N_{train} reference views
Pre-render SAM3D depth/normals; cache $\mathbf{I}_{\text{DA,orig}}^{(v_0)}$; freeze colors
for each iteration (300) **do**
    Compute $\mathcal{L}_{\text{s3}}$ (Equation (8)); apply gradients to $\mathcal{R}$ only; update $\{\boldsymbol{\mu}_i, \mathbf{q}_i, \mathbf{s}_i, o_i\}_{i \in \mathcal{R}}$
end for
return $\hat{\mathcal{G}}_{\text{DA}}$

Baselines. We compare against six representative methods from the current frontier of 3D foundation models: DUSt3R [36], Fast3R [41], MapAnything [19], Pi3 [37], VGGT [35], and DA3 [25]. For the user-preference study we compare against SAM 3D [7].

Metrics. 3D reconstruction accuracy: Chamfer Distance (CD↓, cm) and F-score at 5 cm (F↑, %). *NVS*: PSNR↑ (dB), SSIM↑ [38], LPIPS↓ [49].

4.2 3D Reconstruction Accuracy

Table 1 compares point-cloud reconstruction accuracy on MipNeRF-360 scenes, using COLMAP MVS dense point clouds as the proxy reference surface (see Sec. 4.1). Point clouds are extracted by sampling Gaussian means with an opacity filter $o > 0.1$. We report Chamfer Distance (CD, in cm) and F-score. DASAM3D demonstrates competitive results on both metrics (CD 10.10 cm, F 62.3%), improving over DA3 by approximately 6% on Chamfer Distance. DA3 ranks second on CD (10.76 cm), serving as the primary reference for direct comparison. VGGT leads on F-score (59.7%) among baselines, while DA3 edges ahead on CD, reflecting a coverage vs. distance trade-off common in COLMAP-proxy evaluations. Methods relying purely on 2D correspondences (DUSt3R, MapAnything) show larger errors on both metrics. Qualitative improvements in depth and surface normals are further illustrated in Figure 2.

Table 1: 3D reconstruction accuracy on MipNeRF-360 scenes. CD: Chamfer Distance (cm). F: F-score. **Bold**: best; underline: second best.

Method	CD↓	F↑
DUSt3R [36]	12.93	50.2
Fast3R [41]	11.94	52.7
MapAnything [19]	11.99	51.4
Pi3 [37]	10.92	57.8
VGGT [35]	10.94	59.7
DA3 [25]	10.76	58.9
DASAM3D (Ours)	**10.10**	**62.3**

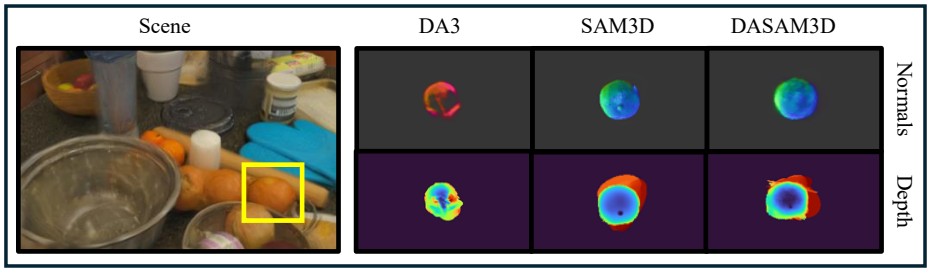

Fig. 2: Depth and surface normal comparison on the Counter scene (MipNeRF-360). Left: the full scene with a bounding box highlighting the onion object used as a representative example. SAM 3D's Gaussian layout provides geometrically accurate normals and depth; DASAM3D's inherits this structural accuracy.

4.3 Novel View Synthesis

Table 2 reports NVS quality on both benchmarks. On DL3DV-10K, DASAM3D achieves 27.08 dB PSNR, 0.817 SSIM, and 0.191 LPIPS—improvements of 0.44 dB, 0.008, and 0.007 over DA3, which is the strongest baseline on this split. On MipNeRF-360 scenes, DASAM3D reaches 24.98 dB PSNR and 0.790 SSIM, with DA3 ranking second (24.71 dB, 0.785 SSIM). VGGT achieves the best LPIPS (0.231) reflecting its diverse pretraining; DASAM3D follows at 0.234 while leading on PSNR and SSIM. Notably, MapAnything achieves higher SSIM than Fast3R despite lower PSNR on both datasets, suggesting its metric constraints favour structural uniformity over peak fidelity. The consistent results across both benchmarks validate that geometry optimization with frozen colors preserves DA3's photorealistic appearance. Figure 3 shows representative qualitative comparisons.

4.4 3D Segmentation: User-Preference Study

Standard photometric metrics do not capture the perceived realism of isolated 3D object reconstructions. We conduct a user-preference study comparing

Table 2: Novel view synthesis on DL3DV-10K and MipNeRF-360. **Bold**: best; underline: second best.

	DL3DV-10K			*MipNeRF-360*		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
DUSt3R [36]	23.41	0.751	0.258	21.83	0.731	0.291
Fast3R [41]	24.87	0.768	0.241	23.12	0.751	0.274
MapAnything [19]	24.13	0.774	0.246	22.54	0.757	0.278
Pi3 [37]	25.34	0.786	0.221	23.97	0.771	0.256
VGGT [35]	25.79	0.797	0.213	24.41	0.780	**0.231**
DA3 [25]	26.64	0.809	0.198	24.71	0.785	0.237
DASAM3D (Ours)	**27.08**	**0.817**	**0.191**	**24.98**	**0.790**	0.234

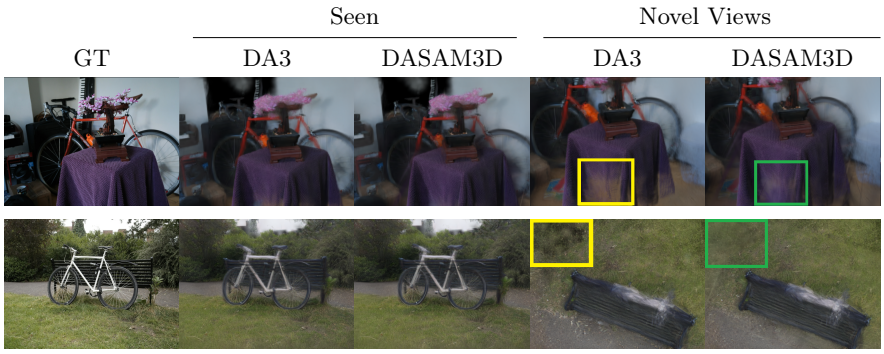

Fig. 3: NVS qualitative comparison on MipNeRF-360. Each row shows a scene with five columns: ground-truth reference (GT), DA3 and DASAM3D renderings on a *seen* view, and DA3 and DASAM3D renderings on a *novel* held-out view. Square markers in the novel-view columns highlight regions where DASAM3D produces sharper detail than DA3, attributable to the layout-based geometry optimization stage.

DASAM3D and SAM 3D on 3D object segmentation quality. We rendered 30 object instances (5 per scene, 6 MipNeRF-360 scenes) from both methods and presented them side-by-side to 20 participants on five criteria: overall preference, appearance realism, geometric accuracy, color fidelity, and texture detail. Results reveal a clear task-dependent split: participants favoured SAM 3D on geometric accuracy (51% vs. 38%), consistent with its role as a dedicated segmentation model that produces clean, object-isolated geometry. However, DASAM3D was preferred on all appearance-related criteria—realism, color fidelity, and texture detail—with margins of 22–35 percentage points, reflecting the effectiveness of the RGB consistency loss $\mathcal{L}_{\text{rgb}}$ (Equation (11)) in anchoring DA3's appearance throughout geometry optimization. Overall preference favoured DASAM3D (52% vs. 38%), indicating that participants weighted appearance quality more

Table 3: User-preference study: DASAM3D vs. SAM 3D on 3D object segmentation quality (20 participants, 30 objects). Values: percentage of trials preferring each option. **Bold**: majority preference.

Criterion	SAM 3D (%)	DASAM3D (%)	Neither (%)
Overall preference	38	**52**	10
Appearance realism	27	**62**	11
Geometric accuracy	**51**	38	11
Color fidelity	29	**61**	10
Texture detail	33	**55**	12

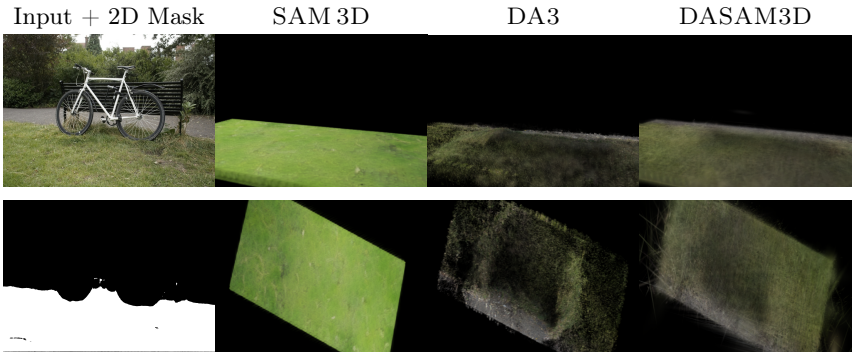

Fig. 4: 3D segmentation comparison. Col. 1: input scene and SAM 2D mask of the *floor* object. Cols. 2–4: the corresponding 3D segment from SAM 3D, DA3, and DASAM3D. SAM 3D produces geometrically precise shapes but with unrealistic appearance differing from the input. DASAM3D inherits SAM 3D's geometric accuracy while preserving DA3's appearance, yielding reconstructions that are both geometrically precise and visually faithful to the input.

heavily than geometric precision in this rendering task. Table 3 and Figure 4 report quantitative and qualitative results.

5 Ablation Studies

All ablations use the Counter scene of MipNeRF-360 unless otherwise stated.

5.1 Alignment Quality vs. Number of Objects

We evaluate how alignment quality scales with the number of SAM 3D objects K. For each $K \in \{1, 2, 4, 6, 8\}$ we select the K highest-confidence objects by mean opacity, run Global Alignment Optimization and Per-Object Alignment Refinement, and measure 3D accuracy on the Counter scene of MipNeRF-360. Table 4 shows that accuracy improves consistently with K (CD drops from 9.41 to

Table 4: Alignment quality vs. object count K (Counter, MipNeRF-360). CD: Chamfer Distance (cm). F: F-score.

K	CD↓	F↑
1	9.41	57.3
2	9.02	59.1
4	8.23	62.8
6	7.79	65.4
8	**7.58**	**67.2**

Table 5: Global vs. Global + Local alignment (three MipNeRF-360 scenes). CD: Chamfer Distance (cm). F: F-score.

Scene	Strategy	CD↓	F↑
Room	Global only	11.41	52.9
	Global + Local	**9.86**	**61.3**
Counter	Global only	10.87	55.2
	Global + Local	**9.52**	**63.4**
Kitchen	Global only	10.53	56.7
	Global + Local	**9.41**	**64.1**

7.58 cm; F-score rises from 57.3% to 67.2%), confirming that a moderate number of well-segmented objects suffices for strong alignment quality.

5.2 Global vs. Global + Local Alignment

We compare Global Alignment Optimization alone against the full two-stage alignment on three MipNeRF-360 scenes. Per-Object Alignment Refinement consistently reduces Chamfer Distance and improves F-score, with the largest benefit in Room (−1.55 cm CD, +8.4% F) and Counter (−1.35 cm, +8.2%), where objects span varying depths and global affine normalization alone leaves residual per-object scale errors. Kitchen benefits slightly less (+7.4% F) as its objects are more uniformly distributed in depth.

5.3 Quality Filtering Effect

We evaluate the quality-based filtering step (Equation (6)) by comparing Stage 3 without filtering against Stage 3 with filtering. The filtering step removes objects whose low overlap fraction or high depth residual indicate poor alignment—most often transparent or heavily occluded instances—that would otherwise inject corrupted geometric priors into Stage 3. Table 6 shows that filtering improves PSNR by 0.36 dB and reduces normal mean angular error (N-MAE) by 1.8°, confirming that even a small fraction of misaligned objects can significantly degrade Stage 3 if left unfiltered.

Table 6: Effect of quality filtering on NVS and geometry (Counter, MipNeRF-360). N-MAE: normal mean angular error (degrees).

Configuration	PSNR↑	SSIM↑	LPIPS↓	N-MAE↓
DA3 baseline	24.63	0.791	0.231	18.4
DASAM3D (no filter)	24.78	0.793	0.229	17.4
DASAM3D (filtered)	**25.14**	**0.801**	**0.221**	**15.6**

Table 7: Stage 3 loss ablation (Counter, MipNeRF-360). Each row removes one component from the full model. AbsRel: absolute relative depth error.

Removed component	PSNR↑	SSIM↑	LPIPS↓	N-MAE↓	Abs Rel↓
None (full DASAM3D)	**25.14**	**0.801**	**0.221**	**15.6**	**0.098**
w/o $\mathcal{L}_{\text{depth}}$ (Equation (9))	24.79	0.791	0.231	19.8	0.142
w/o $\mathcal{L}_{\text{norm}}$ (Equation (10))	24.98	0.797	0.233	21.3	0.112
w/o $\mathcal{L}_{\text{rgb}}$ (Equation (11))	24.23	0.783	0.249	16.2	0.104

5.4 Stage 3 Loss Component Ablation

We ablate each Stage 3 loss term individually on the Counter scene. Removing $\mathcal{L}_{\text{depth}}$ (Equation (9)) causes the largest degradation in absolute depth accuracy (AbsRel increases from 0.098 to 0.142) and also substantially degrades surface normal consistency (N-MAE rises from 15.6° to 19.8°). Removing $\mathcal{L}_{\text{norm}}$ (Equation (10)) has the strongest impact on surface orientation specifically (N-MAE rises to 21.3°, the worst across all ablations) and also worsens LPIPS to 0.233, degrading perceptual quality beyond even the depth ablation variant (0.231), indicating that normal consistency contributes substantially to perceptual quality beyond geometry alone. Removing $\mathcal{L}_{\text{rgb}}$ (Equation (11)) produces the most severe appearance degradation (PSNR drops by 0.91 dB; LPIPS worsens by 0.028), validating that the RGB anchor is essential for preventing geometry updates from corrupting DA3's photorealistic textures.

6 Discussion

Why a volumetric 3D layout yields advantages over 2D priors. Unlike monocular depth maps [34, 39], SAM 3D's Gaussian layouts carry full 3D position, scale, and orientation, enabling volumetric ROI definition (Equation (7)) and the 3D Chamfer supervision in Per-Object Alignment Refinement (Equation (3)). DASAM3D reduces Chamfer Distance by 6.1% over DA3 on MipNeRF-360 (Tab. 1), while purely 2D methods trail by larger margins.

Mixed 2D–3D supervision enables robust per-object placement. Per-Object Alignment Refinement (Equation (5)) reduces Chamfer Distance by 1.35–1.55 cm and improves F-score by 8 pp over Global Alignment Optimization alone (Tab. 5).

The quality scoring (Equation (6)) then gates which objects reach Layout-Based Geometry Optimization, providing robustness to imperfect SAM 3D outputs (Tab. 6).

ROI masking decouples geometry from appearance. Layout-Based Geometry Optimization restricts gradients to adaptive volumetric ROIs with frozen colors and an RGB anchor $\mathcal{L}_{\text{rgb}}$, enabling simultaneous improvements in 3D accuracy and NVS quality (Tabs. 1 and 2). The ablation in Tab. 7 confirms the RGB anchor is the primary guard for appearance (0.91 dB drop when removed) and depth supervision for geometry (AbsRel: 0.098→0.142).

Inherited flexibility from DA3. DASAM3D inherits DA3's support for posed and pose-free inputs at any number of views, making it directly applicable across diverse capture conditions.

Limitations. (i) Stage 2 runtime scales linearly with K; batch parallelization would mitigate this. *(ii)* SAM 3D failures on transparent or heavily occluded objects can produce low-quality priors, partially mitigated by quality filtering. *(iii)* Very large scale differences between DA3 and SAM 3D outputs may require coarser initialization before Stage 1. *(iv)* Extension to outdoor scenes with sparser object structure is an important future direction.

7 Conclusion

We presented DASAM3D, a three-stage differentiable framework that fuses Depth Anything 3 and SAM 3D at the 3D Gaussian parameter level via Global Alignment Optimization, Per-Object Alignment Refinement with quality-based filtering, and Layout-Based Geometry Optimization with depth, normal, and RGB consistency losses. By building on DA3, DASAM3D inherits its ability to operate in both posed and pose-free settings and with any number of input images. Experiments on MipNeRF-360 scenes show consistent geometry improvements in reconstruction accuracy and surface normal quality, and NVS results on both MipNeRF-360 and DL3DV-10K demonstrate favorable performance over strong baselines; a user-preference study confirms that participants perceive DASAM3D's object reconstructions as more visually realistic than those of SAM 3D alone. Future directions include joint end-to-end optimization of alignment and geometry stages; extension to dynamic scenes using SAM 2 [33]; integration with 3DGS editing pipelines [9]; and application to robotic manipulation and semantic navigation that demand both geometric accuracy and photorealistic appearance [15].

References

1. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields.

In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5855–5864 (2021) 1
2. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5470–5479 (2022) 1, 7
3. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Zip-nerf: Anti-aliased grid-based neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 19697–19705 (2023) 1
4. Besl, P.J., McKay, N.D.: Method for registration of 3-d shapes. In: Sensor fusion IV: control paradigms and data structures. vol. 1611, pp. 586–606. Spie (1992) 4
5. Cen, J., Fang, J., Yang, C., Xie, L., Zhang, X., Shen, W., Tian, Q.: Segment any 3d gaussians. In: Proceedings of the AAAI conference on artificial intelligence. vol. 39, pp. 1971–1979 (2025) 3
6. Charatan, D., Li, S.L., Tagliasacchi, A., Sitzmann, V.: pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 19457–19467 (2024) 2, 3
7. Chen, X., Chu, F.J., Gleize, P., Liang, K.J., Sax, A., Tang, H., Wang, W., Guo, M., Hardin, T., Li, X., et al.: Sam 3d: 3dfy anything in images. arXiv preprint arXiv:2511.16624 (2025) 2, 3, 8
8. Chen, Y., Wu, Q., Lin, W., Harandi, M., Cai, J.: Hac: Hash-grid assisted context for 3d gaussian splatting compression. In: European Conference on Computer Vision. pp. 422–438. Springer (2024) 3
9. Chen, Y., Chen, Z., Zhang, C., Wang, F., Yang, X., Wang, Y., Cai, Z., Yang, L., Liu, H., Lin, G.: Gaussianeditor: Swift and controllable 3d editing with gaussian splatting. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 21476–21485 (2024) 4, 14
10. Chen, Y., Xu, H., Zheng, C., Zhuang, B., Pollefeys, M., Geiger, A., Cham, T.J., Cai, J.: Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In: European conference on computer vision. pp. 370–386. Springer (2024) 2, 3
11. Cheng, K., Long, X., Yang, K., Yao, Y., Yin, W., Ma, Y., Wang, W., Chen, X.: Gaussianpro: 3d gaussian splatting with progressive propagation. In: Forty-first International Conference on Machine Learning (2024) 2, 3
12. Fan, H., Su, H., Guibas, L.J.: A point set generation network for 3d object reconstruction from a single image. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 605–613 (2017) 4, 5
13. Fan, Z., Cong, W., Wen, K., Wang, K., Zhang, J., Ding, X., Xu, D., Ivanovic, B., Pavone, M., Pavlakos, G., et al.: Instantsplat: Sparse-view gaussian splatting in seconds. arXiv preprint arXiv:2403.20309 (2024) 2, 3
14. Fang, G., Wang, B.: Mini-splatting: Representing scenes with a constrained number of gaussians. In: European conference on computer vision. pp. 165–181. Springer (2024) 3
15. Gu, Q., Kuwajerwala, A., Morin, S., Jatavallabhula, K.M., Sen, B., Agarwal, A., Rivera, C., Paul, W., Ellis, K., Chellappa, R., et al.: Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In: 2024 IEEE International Conference on Robotics and Automation (ICRA). pp. 5021–5028. IEEE (2024) 4, 14
16. Guédon, A., Lepetit, V.: Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5354–5363 (June 2024) 4

17. Huang, B., Yu, Z., Chen, A., Geiger, A., Gao, S.: 2d gaussian splatting for geometrically accurate radiance fields. In: ACM SIGGRAPH 2024 conference papers. pp. 1–11 (2024) 2, 3
18. Huang, X., Mei, G., Zhang, J.: Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11366–11374 (2020) 4
19. Keetha, N., Müller, N., Schönberger, J., Porzi, L., Zhang, Y., Fischer, T., Knapitsch, A., Zauss, D., Weber, E., Antunes, N., et al.: Mapanything: Universal feed-forward metric 3d reconstruction. arXiv preprint arXiv:2509.13414 (2025) 1, 3, 8, 9, 10
20. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G., et al.: 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. **42**(4), 139–1 (2023) 1, 3
21. Kerr, J., Kim, C.M., Goldberg, K., Kanazawa, A., Tancik, M.: Lerf: Language embedded radiance fields. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 19729–19739 (2023) 3
22. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 5, 7
23. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4015–4026 (2023) 3
24. Leroy, V., Cabon, Y., Revaud, J.: Grounding image matching in 3d with mast3r. In: European conference on computer vision. pp. 71–91. Springer (2024) 1, 3
25. Lin, H., Chen, S., Liew, J., Chen, D.Y., Li, Z., Shi, G., Feng, J., Kang, B.: Depth anything 3: Recovering the visual space from any views. arXiv preprint arXiv:2511.10647 (2025) 2, 3, 8, 9, 10
26. Ling, L., Sheng, Y., Tu, Z., Zhao, W., Xin, C., Wan, K., Yu, L., Guo, Q., Yu, Z., Lu, Y., et al.: Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22160–22169 (2024) 7
27. Lu, T., Yu, M., Xu, L., Xiangli, Y., Wang, L., Lin, D., Dai, B.: 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) 2, 3
28. Matsuki, H., Murai, R., Kelly, P.H., Davison, A.J.: Gaussian splatting slam. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 18039–18048 (2024) 3
29. 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) 1
30. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM transactions on graphics (TOG) **41**(4), 1–15 (2022) 1, 3
31. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019) 5, 7
32. Qin, M., Li, W., Zhou, J., Wang, H., Pfister, H.: Langsplat: 3d language gaussian splatting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20051–20060 (2024) 3

33. Ravi, N., Gabeur, V., Hu, Y.T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., et al.: Sam 2: Segment anything in images and videos. arXiv preprint arXiv:2408.00714 (2024) 3, 14
34. Turkulainen, M., Ren, X., Melekhov, I., Seiskari, O., Rahtu, E., Kannala, J.: Dn-splatter: Depth and normal priors for gaussian splatting and meshing. In: 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 2421–2431. IEEE (2025) 3, 13
35. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 5294–5306 (2025) 1, 3, 8, 9, 10
36. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: Dust3r: Geometric 3d vision made easy. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 20697–20709 (2024) 1, 3, 8, 9, 10
37. Wang, Y., Zhou, J., Zhu, H., Chang, W., Zhou, Y., Li, Z., Chen, J., Pang, J., Shen, C., He, T.: Pi3: Permutation-equivariant visual geometry learning. arXiv preprint arXiv:2507.13347 (2025) 1, 3, 8, 9, 10
38. 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) 8
39. Wu, Q., Zheng, J., Cai, J.: Surface reconstruction from 3d gaussian splatting via local structural hints. In: European Conference on Computer Vision. pp. 441–458. Springer (2024) 3, 13
40. Xiang, H., Li, X., Cheng, K., Lai, X., Zhang, W., Liao, Z., Zeng, L., Liu, X.: Gaussianroom: Improving 3d gaussian splatting with sdf guidance and monocular cues for indoor scene reconstruction. In: 2025 IEEE International Conference on Robotics and Automation (ICRA). pp. 2686–2693. IEEE (2025) 3
41. Yang, J., Sax, A., Liang, K.J., Henaff, M., Tang, H., Cao, A., Chai, J., Meier, F., Feiszli, M.: Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 21924–21935 (2025) 1, 3, 8, 9, 10
42. Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., Zhao, H.: Depth anything v2. Advances in Neural Information Processing Systems **37**, 21875–21911 (2024) 3
43. Ye, M., Danelljan, M., Yu, F., Ke, L.: Gaussian grouping: Segment and edit anything in 3d scenes. In: European conference on computer vision. pp. 162–179. Springer (2024) 3
44. Ye, V., Li, R., Kerr, J., Turkulainen, M., Yi, B., Pan, Z., Seiskari, O., Ye, J., Hu, J., Tancik, M., et al.: gsplat: An open-source library for gaussian splatting. Journal of Machine Learning Research **26**(34), 1–17 (2025) 7
45. Ying, H., Yin, Y., Zhang, J., Wang, F., Yu, T., Huang, R., Fang, L.: Omniseg3d: Omniversal 3d segmentation via hierarchical contrastive learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20612–20622 (2024) 3
46. Yu, Z., Chen, A., Huang, B., Sattler, T., Geiger, A.: Mip-splatting: Alias-free 3d gaussian splatting. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 19447–19456 (2024) 2, 3
47. Yu, Z., Sattler, T., Geiger, A.: Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. ACM Transactions on Graphics (ToG) **43**(6), 1–13 (2024) 3
48. Zhang, J., Herrmann, C., Hur, J., Jampani, V., Darrell, T., Cole, F., Sun, D., Yang, M.H.: Monst3r: A simple approach for estimating geometry in the presence of motion. arXiv preprint arXiv:2410.03825 (2024) 3

49. 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) 8