

Fus3D: Decoding Consolidated 3D Geometry from Feed-forward Geometry Transformer Latents

Laura Fink¹, Linus Franke^{1,2,3}, George Kopanas⁴, Marc Stamminger¹,
and Peter Hedman[†]

¹ Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

² Inria, Université Côte d’Azur, France

³ Julius-Maximilians-Universität Würzburg, Germany

⁴ Google DeepMind, UK

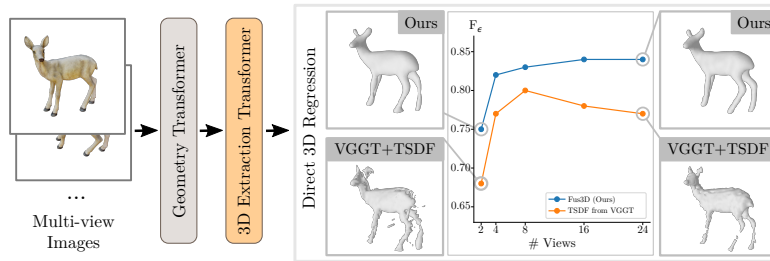


Fig. 1: Fus3D directly regresses a 3D representation from the latent space of a multi-view geometry transformer, bypassing per-view prediction and post-hoc fusion. This, compared to VGGT [44] using TSDF fusion, yields improved surface completeness under sparse views (left, 2 views) and avoids error accumulation reducing details at scale (right, 24 views), also reflected in the F-score curves across view counts (center).

Abstract. We propose a feed-forward method for dense Signed Distance Field (SDF) regression from unstructured image collections in less than three seconds, without camera calibration or post-hoc fusion. Our key insight is that the intermediate feature space of pretrained multi-view feed-forward geometry transformers (FFGT) already encodes a powerful joint world representation; yet, existing pipelines discard it, routing features through per-view prediction heads before assembling 3D geometry post-hoc, which discards valuable completeness information and accumulates inaccuracies. We instead perform 3D extraction directly from FFGT features via learned volumetric extraction: voxelized canonical embeddings that progressively absorb multi-view geometry information through interleaved cross- and self-attention into a structured volumetric latent grid. A simple convolutional decoder then maps this grid to a dense SDF. We additionally propose a scalable, validity-aware supervision scheme directly using SDFs derived from depth maps or 3D assets, tackling practical issues like non-watertight meshes. Our approach yields complete and well-defined distance values across sparse- and dense-view settings and demonstrates geometrically plausible completions.

Keywords: Attention-based Voxel Query · Learned Geometry Fusion

[†] Work done while at Google DeepMind, UK

1 Introduction

3D reconstruction from unstructured image collections and video remains a fundamental challenge in computer vision, with broad implications for downstream tasks in semantic understanding, robotics, and scene interaction. A key requirement in such settings is a complete 3D representation, e.g. when accurate distance information is critical. Distance information, often in the form of Signed Distance Functions (SDFs), is difficult to acquire when input views are sparse or otherwise poorly conditioned. In this work, we provide a novel method for this online and sparse-view setting by leveraging extensively pretrained multi-view geometry transformers and directly regressing 3D SDFs from them.

Recent feed-forward multi-view geometry transformers, such as DUS3R [47], VGGT [44], or DA3 [23], have demonstrated remarkable capabilities in recovering scene geometry directly from images, producing per-image depth or point maps alongside camera poses and intrinsics in a single forward pass. Central to their success is a powerful geometry prior acquired through large-scale training, enabling robust reconstruction even under challenging conditions such as strong symmetry or minimal inter-view overlap – scenarios that confound classical feature-matching pipelines.

Despite these advances, a fundamental bottleneck remains in how 3D scenes are ultimately assembled from the per-image predictions of these transformers. Current pipelines rely on classical post-hoc fusion of per-view outputs, typically via point cloud merging, Truncated SDF (TSDF) fusion or Poisson reconstruction to obtain a coherent 3D representation. This design has two consequential failure modes. First, in sparse-view settings, the transformer’s learned world prior is only leveraged locally, per image, leaving unobserved regions geometrically underspecified and yielding incomplete reconstructions (Fig. 1, bottom left). Second, when processing many views, minor per-view inconsistencies accumulate and propagate into the fused representation, introducing noise that degrades reconstruction fidelity at scale (Fig. 1, bottom right, red curve).

We address both failure modes through a single unifying idea: rather than fusing per-view outputs in 3D space, we directly regress the SDF from the transformer’s intermediate latent features. This allows the full multi-view geometry prior, jointly built up across all input views, to inform the 3D estimate at every spatial location, yielding complete reconstructions in sparse-view regimes and noise-free scaling with more views (see crops in Fig. 1).

The central technical challenge is the dimensionality mismatch between the set of 2D input tokens and the 3D target representation. We resolve this by introducing a learned embedding query that attend into the geometry transformer’s feature space, effectively lifting 2D multi-view features into a 3D structured latent space without requiring any 3D-to-3D autoencoder training regimes. Crucially, we perform repeated extraction during the feature construction process of the multi-view transformer in order to build meaningful 3D latent features. As we demonstrate, the resulting features encode a rich geometry prior including object-level symmetry and shape statistics which enables geometrically complete and well-defined SDFs even for unobserved surfaces. Finally, the intermediate fea-

ture volume reveals emergent semantic structure that could serve as a foundation for downstream perception. In summary, we make the following contributions:

- We identify the per-view prediction bottleneck in feed-forward geometry transformers as the root cause of both sparse-view incompleteness and noise accumulation under many views, and thus propose 3D extraction directly from the transformer’s intermediate feature space.
- We introduce a learned volumetric extraction module: a set of 3D-position-conditioned canonical embeddings that progressively fuse multi-view geometry features into a structured latent grid via attention, enabling projection-free 2D-to-3D lifting.
- We propose a validity-aware SDF supervision scheme, to handle non-watertight meshes and unobserved regions gracefully, enabling scalable training on large-scale real-world datasets.
- Our results indicate a geometric shape prior encoded within the 3D feature space, enabling complete and robust reconstructions in challenging scenarios.

The project page and source code are located at: lorafib.github.io/fus3d/.

2 Related Work

Classical and Online Reconstruction. Classical pipelines using Structure-from-Motion (SfM) and Multi View Stereo (MVS) remain strong baselines for reconstruction under sufficient overlap and texture [38–40], while SLAM systems coupled with TSDF fusion enable real-time incremental reconstruction in RGB-D settings [26, 29, 32, 37]. Both approaches, however, handle camera calibration, depth estimation and geometry fusion as distinct tasks.

Analysis-by-Synthesis Neural rendering methods formulate 3D reconstruction through view synthesis based on various scene representations, with NeRF [28] and Gaussian splatting [17] as canonical examples. Learning-based implicit shape representations such as SDFs and occupancy functions have become popular choices for geometry modeling [27, 34]. NeuS and its follow-ups achieve high-fidelity surface reconstruction by coupling SDF representations with differentiable rendering [21, 45, 50].

Feed-Forward Geometry from Unposed Images. DUST3R jointly recovers dense geometry and camera parameters from unconstrained image collections via point-map regression [47], substantially reducing the separation between matching, pose estimation, and dense reconstruction. VGGT, π^3 , and MapAnything extend this trend to broader geometric outputs, in the form of depth, point maps, cameras, and tracks, with strong generalization across variable numbers of views [16, 44, 49]. While VGGT achieves impressive geometric generalization, its outputs remain per-view predictions that must be consolidated via TSDF fusion to yield a surface, discarding the joint multi-view prior in the process; this predict-then-fuse pipeline serves as a direct baseline for our approach.

A parallel line of work augments feed-forward prediction with persistent spatial memory: Spann3R maintains a global-frame memory for consistent point map prediction [42], while CUT3R develops a stateful formulation for improved temporal consistency [46]. Despite impressive progress, all of these methods produce per-pixel or per-patch 2.5D outputs and require post-hoc fusion to assemble coherent 3D geometry. Their proposed spatial memories support only ray-map or pixel-aligned readouts [46], not direct 3D positional queries. Our work departs from this paradigm entirely, performing 3D extraction directly in the transformer’s latent space without any per-view prediction or post-hoc fusion.

Generalizable Reconstruction via Differentiable Rendering. Generalizable novel view synthesis (NVS) methods reconstruct radiance fields or Gaussian primitives from sparse views, under known [3, 4, 58] or unposed [14, 53] cameras. Generalizable reconstruction methods such as LRM [10] and its follow-ups [9, 61] regress structured 3D representations from sparse images via large transformers, with MeshLRM [51] extending this to end-to-end mesh reconstruction through differentiable mesh rendering. M-LRM [20] proposes 3D aware positional encoding for latent features projected into voxel grids, and GeoLRM [59] fuse voxel-based sparse latent features to decode Gaussians, showcase the promising direction of 3D aware reconstruction in posed scenarios.

Post-hoc consolidation methods address view-aligned primitive redundancy via fusion [14], pruning [57], or projecting latents via depth maps into grids [48], but remain bound to the predict-then-fuse composition. We instead aggregate multi-view features directly into a 3D latent volume, avoiding view-aligned intermediate predictions entirely. Generalizable SDF regression approaches, such as SparseNeuS [24], VolRecon [36], and UFORecon [31] use learned correspondence finding based on cost volumes or frustum features and neural SDF decoding. Hence, VolRecon and UFORecon represent the closest prior work to ours in terms of output representation and serve as our primary geometry quality baselines; however, both require known camera parameters at test time in contrast to our fully feed-forward, pose-free formulation.

3D Generative Methods. Autoencoder-based pipelines compress explicit 3D representations into volumetric latent codes that support generation, interpolation, and image-conditioned reconstruction [2, 5, 8, 30, 52]. These structured latent spaces have proven especially expressive for high-fidelity geometry generation and are increasingly paired with diffusion models for diverse shape synthesis [52]. Our extraction transformer shares this volumetric latent structure but departs fundamentally in how it is trained: autoencoder pipelines require 3D-to-3D training in order to learn a meaningful latent representation, thus relying on high quality 3D data not only as supervision signal but also as input to the encoder which is often omitted for the actual generation task [52]. While direct 3D supervision is replaceable using differential rendering, the reliance on 3D data as input during training creates a bottleneck for dataset generation [13]. By leveraging a pretrained backbone with 2D-to-3D training, we shift the burden of represen-

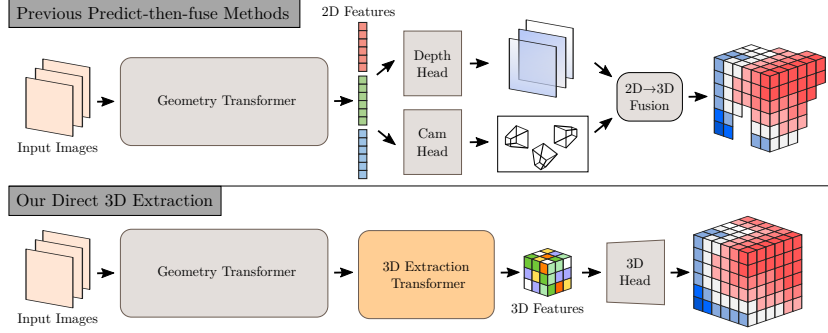


Fig. 2: Juxtaposition of geometric extraction methodologies. Common pipelines like VGGT [44] (top) route transformer features through per-view 2D decoder heads, discarding the joint multi-view representation before 3D assembly. We instead extract dense 3D features directly from the transformer’s intermediate feature space, preserving the full multi-view information.

tation learning to scalable 2D models, using limited 3D data only to lift their powerful priors into a consolidated 3D space.

3 Method

Contemporary geometry transformers $\mathcal{G}$ [44, 47] encode a set of context images $\{I_i\}_{i=1}^N$ into per-patch features, processed by interleaved frame-local self-attention and frame-global cross-attention blocks before being routed to specialized 2D decoder heads $\mathcal{H}_{\text{depth}}, \mathcal{H}_{\text{pts}}, \mathcal{H}_{\text{cam}}$ that produce per-image depth maps, point maps, and camera parameters in one-to-one correspondence with input patches. A coherent 3D scene is then recovered via post-hoc fusion, lifting per-view predictions into a shared coordinate frame via point cloud followed by a surface reconstruction step. This is entirely decoupled from the transformer’s internal representations and misses the opportunity to leverage the rich priors encoded in the model. In this design, the expressive joint multi-view feature space assembled by $\mathcal{G}$ is discarded before the 3D representation is constructed.

In contrast, we replace the per-view decoder heads with a learned 3D extraction module $\mathcal{E}$ that operates directly on the transformer’s intermediate feature space. Rather than projecting back to individual image planes, $\mathcal{E}$ aggregates multi-view geometry features into a dense volumetric feature grid via attention, without explicit camera projection or per-view fusion, allowing the full joint multi-view prior to inform every spatial location simultaneously. A juxtaposition of these two paradigms can be seen in Fig. 2.

3.1 Learned Volumetric Extraction from Multi-View Features

To directly regress a 3D representation from a set of input images $\{I_i\}_{i=1}^N$, we propose a three-stage pipeline: a pretrained geometry transformer backbone $\mathcal{G}$

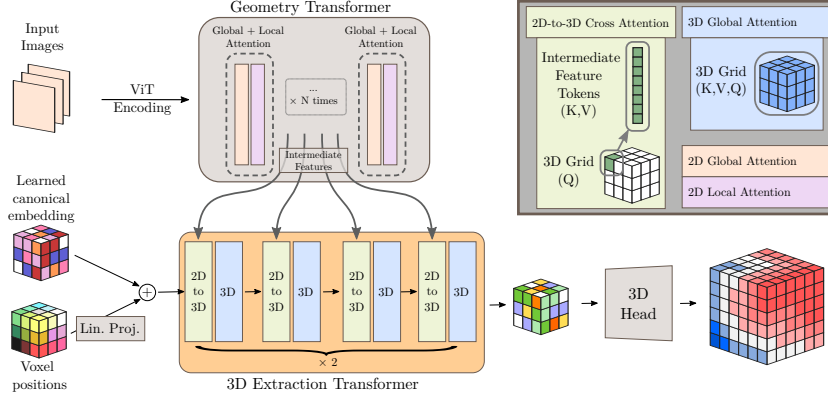


Fig. 3: Architecture of Fus3D: The geometry transformer $\mathcal{G}$ (beige) processes tokenized input images, yielding a list of 2D intermediate features $\{z_{2D}^b\}_{b=1}^B$ extracted from different stages. The extraction transformer $\mathcal{E}$ (orange) leverages 2D-to-3D cross attention (green) to absorb the 3D information into features of the learned canonical embedding z_{3D} , and distributes this information via 3D self-attention (blue) throughout the volumetric latent. The head $\mathcal{H}_{3D}$ decodes the resulting structured latent $\hat{z}_{3D}$ into a dense SDF grid.

that provides powerful multi-view geometry features; a 3D extraction transformer $\mathcal{E}$ that fuses the intermediate features of $\mathcal{G}$ into a dense volumetric latent grid; and a 3D decoder head $\mathcal{H}_{3D}$ that maps these latent features to a dense grid of SDF values $\hat{f}_{X_\Omega}$ in a grid X_Ω over a volume domain Ω . An overview of the architecture is given in Fig. 3.

Backbone. Our design requires that all input views are processed within a unified feature space, such that intermediate representations are directly comparable across frames. This property holds for geometry transformers like VGGT [44], DA3 [23], and π^3 [49], which process all views jointly in a shared token sequence, but not for pair-wise methods such as MAST3R [19] or DUST3R [47], which handle image pairs independently and aggregate predictions post-hoc. We therefore adopt the transformer architecture of VGGT as the geometry backbone $\mathcal{G}$. As a direct consequence, no explicit conditioning on precomputed camera calibration is required. We retain VGGT’s coordinate convention, placing the scene’s coordinate frame at the camera center of the first input view.

Learned Canonical Embedding. The mismatch in dimensionality between per-image feature representations and the target 3D output is a major difficulty for 3D feature extraction from 2D latents from $\mathcal{G}$. To address this problem, we take inspiration from neural memory approaches for continuous reconstruction [42, 46], in which a scene-agnostic learned embedding, initialized independently of any observed image, serves as the initial state of a memory that is subsequently updated through attention layers [41]. We generalize this idea to

3D: rather than using a 2D positional encoding, we condition a set of learned embeddings z_{3D} (we use extents of 16^3 and dimension $d=2048$) on their corresponding 3D voxel positions within the volume Ω . These position-conditioned latent features constitute the initial state of the 3D memory, into which multi-view geometry information is progressively fused and compressed.

The voxel-grid structure implies that the region of interest must be approximately known at inference time. While the bounding box of Ω can, in principle, vary per scene, we adopt a scheme whereby the datasets are considered in a normalized coordinate space, yielding a fixed bounding box relative to the first input view, common in the field [7, 31, 36, 54]. (In Sec. 5, we show preliminary results for queries with varying scale and shift.)

Multi-view Feature Aggregation. Following the design of 2D decoder heads [44], we adopt a hierarchical design to compute volumetric features from our learned 3D embeddings z_{3D} and the intermediate 2D latent features $\{z_{2D}^b\}_{b=1}^B$ of the backbone $\mathcal{G}$. Our 3D extraction transformer $\mathcal{E}$ lifts and computes the updated volumetric 3D features as

$$\hat{z}_{3D} = \mathcal{E}\left(z_{3D}, \{z_{2D}^b\}_{b=1}^B\right), \quad (1)$$

elements of $\{z_{2D}^b\}_{b=1}^B$ are injected at distinct stages of $\mathcal{E}$, as seen in Fig. 3.

Concretely, the embeddings z_{3D} serve as the initial queries for the first cross-attention module of $\mathcal{E}$, whose keys and values are drawn from the first intermediate feature map z_{2D}^1 . The attended volume is then passed through a volumetric self-attention block, which propagates information across spatial locations in the volume Ω . Following the interleaved local-global attention design of Wang et al. [44], these two operations form a single transformer block. The process is repeated for each intermediate feature map in $\{z_{2D}^b\}_{b=1}^B$ ($B=4$) with the updated 3D features $\hat{z}_{3D}$ carried forward between stages; the full four-stage sequence is then applied twice, yielding 8 transformer blocks in total.

3.2 Feed-forward SDF Regression

We define the ideal signed distance function f for the shape $S \subset \mathbb{R}^3$ as

$$f(\mathbf{x}) = \begin{cases} -d(\mathbf{x}, \partial S), & \mathbf{x} \in S, \\ d(\mathbf{x}, \partial S), & \mathbf{x} \notin S, \end{cases} \quad (2)$$

where $\mathbf{x} \in S$ are points inside the shape, $\mathbf{x} \notin S$ are outside and $\mathbf{x} \in \partial S$ are at the boundary of S . The unsigned distance is then $d(\mathbf{x}, \partial S) = \min_{\mathbf{s} \in \partial S} \|\mathbf{x} - \mathbf{s}\|_2$.

Decoding. The volumetric latent features $\hat{z}_{3D}$ produced by $\mathcal{E}$ are decoded by $\hat{f}_{X_\Omega} = \mathcal{H}_{3D}(\hat{z}_{3D})$ with a resolution of 64^3 . $\mathcal{H}_{3D}$ is a simple convolutional up-sampling network. $\hat{f}_{X_\Omega}$ represents the SDF values $\hat{f}(\mathbf{x}_{i,j,k})$ corresponding to the voxel center positions $\mathbf{x}_{i,j,k}$ at index (i, j, k) of grid X_Ω sampling the bounding volume Ω . The voxel size is denoted as ϵ .

Supervision. Following prior SDF works [55, 56], we supervise the network with a combination of a direct value loss $\mathcal{L}_{\text{SDF}}$, a gradient loss $\mathcal{L}_{\nabla}$, and add an Eikonal regularizer $\mathcal{L}_{\text{eik}}$. However, in practice, target SDF values may not be well-defined everywhere within Ω , requiring masking.

Validity Masks. TSDF fusion leaves unobserved regions without valid values, and large-scale 3D asset datasets (Sec. 4) frequently contain non-watertight meshes, for which the interior is ambiguous and spurious sign flips arise in case of open boundaries (see the supplemental material). Rather than relying on mesh pre-processing [22, 56], we instead model these failure modes in our training loss through two binary masks defined on X_{Ω} .

The *validity mask* $M_{\text{val}} \subseteq X_{\Omega}$ identifies positions $\mathbf{x}$ for which the ground-truth SDF value is well-defined and reliable; all loss terms are evaluated exclusively on $X \subseteq M_{\text{val}}$.

The *Eikonal mask* $M_{\text{eik}} \subseteq X_{\Omega}$ identifies positions in which the ground-truth SDF is locally smooth and sign-consistent. It is constructed by evaluating $\mathcal{L}_{\text{eik}}$ on the ground-truth field $f_{X_{\Omega}}$ itself. If the Eikonal term is greater than 1 by a sufficiently large amount, we classify the value and its local neighborhood as unreliable. Consequently, unreliable regions are excluded from sign-sensitive supervision. Using these masks, we adapt the regression losses in the following ways:

$$\mathcal{L}_{\text{SDF}}(X) = \frac{1}{|X|} \sum_{\mathbf{x}_n \in X} \begin{cases} \|\hat{f}(\mathbf{x}_n) - f(\mathbf{x}_n)\|_1, & \mathbf{x}_n \in M_{\text{eik}}, \\ \|\hat{f}(\mathbf{x}_n) - |f(\mathbf{x}_n)|\|_1, & \mathbf{x}_n \notin M_{\text{eik}}, \end{cases} \quad (3)$$

$$\mathcal{L}_{\nabla}(X) = \frac{1}{|X|} \sum_{\mathbf{x}_n \in X} \begin{cases} \|\nabla_{\mathbf{x}} \hat{f}(\mathbf{x}_n) - \nabla_{\mathbf{x}} f(\mathbf{x}_n)\|_2, & \mathbf{x}_n \in M_{\text{eik}}, \\ 0, & \mathbf{x}_n \notin M_{\text{eik}}. \end{cases} \quad (4)$$

Outside M_{eik} , $\mathcal{L}_{\text{SDF}}$ degrades gracefully to an unsigned distance variant, preserving a meaningful signal without imposing incorrect sign supervision.

Additionally, we keep the camera loss $\mathcal{L}_{\text{cam}}$ by Wang et al. [44] to retain the $\mathcal{G}$'s camera estimation functionality.

Total Loss. We combine the above terms as:

$$\mathcal{L}_{\text{tot}} = \lambda_s \mathcal{L}_{\text{SDF}}(X_S) + \lambda_G \mathcal{L}_{\text{SDF}}(X_{\Omega}) + \lambda_{\nabla} \mathcal{L}_{\nabla}(X_{\Omega}) + \lambda_{\text{eik}} \mathcal{L}_{\text{eik}}(X_{\Omega}) + \mathcal{L}_{\text{cam}}, \quad (5)$$

where $\mathcal{L}(X_{\Omega})$ refer to losses evaluated on voxel centers of grid X_{Ω} , and additionally employ $\mathcal{L}_{\text{SDF}}(X_S)$, where X_S refers to sample positions close to the shape boundary ∂S to put extra emphasis on surface extraction. The gradients for $\mathcal{L}_{\nabla}(X_{\Omega})$ and $\mathcal{L}_{\text{eik}}(X_{\Omega})$ are computed by finite differences on the grid.

4 Experimental Setup

Datasets and Sampling

All experiments require scenes annotated with images, camera parameters, and

a surface mesh. Meshes are converted to SDFs or TSDFs [29] via rendered depth maps on the fly (see the supplemental material for details). Unlike VGGT [44], we do not normalize scale relative to the first view’s depth, instead normalizing to a dataset-global scene range. Both datasets feature restricted camera setups, simplifying the regression task (see the supplemental material). Mini-batches of size b are constructed by drawing b scenes, each with 2–8 unique views sampled uniformly without overlap constraints. Images are rescaled to 448 px on the longer side; augmentation crops the shorter side to 228–448 px and applies consistent color jitter on each view set.

Objaverse. We use $\approx 5\%$ of the OBJVERSE [6] subset used by Xiang et al. [52]: 17K object-centric scenes with 24 views each, and 170 held-out scenes for evaluation. Supervision meshes are carved to retain only triangles potentially visible from outside the object. We note that this is a comparably small training set, leaving substantial headroom for improvement at larger scale.

DTU. For forward-facing DTU scenes [12], objects are only partially observed, so the ground-truth mesh is open and SDF signs (inside/outside) are ill-defined. Hence, we adapt the calculation of losses in two ways: we compute $\mathcal{L}_{\text{SDF}}(\mathbf{X}_O)$ from a depth-fused TSDF. For $\mathcal{L}_{\text{SDF}}(\mathbf{X}_S)$, we keep using the GT mesh but resort to its unsigned variant by setting $\mathbf{M}_{\text{eik}}$ to the empty set during its computation. The setup of train and test splits follows VolRecon [36] and UFORcon [31].

Training

Initialization. The backbone $\mathcal{G}$ is initialized with the publicly available VGGT weights [43], which use DINOv2 for patch tokenization [33]. The 3D decoder head $\mathcal{H}_{3D}$ follows the structure generation decoder of Xiang et al. [52] with the final sigmoid activation removed, initialized from their released weights accordingly. The extraction transformer $\mathcal{E}$ and canonical embedding z_{3D} are trained from scratch.

Backbone finetuning. In these experiments, we specialize our back-end to work better for object-centric datasets. Inspired by Ren et al. [35], we achieve this through light-weight LoRA fine-tuning [11] and apply the same fine-tuning strategy consistently across all comparison methods.

Training Curriculum. We employ a two-stage curriculum. In the first stage, the SDF is predicted at low resolution (16^3) using a single linear decoder, supervised with $\mathcal{L}_{\text{SDF}}$ and $\mathcal{L}_{\text{cam}}$ only. In the second stage, $\mathcal{H}_{3D}$ is introduced to predict the full-resolution SDF grid under the complete loss described in Sec. 3.2. For DTU experiments, we fine-tune the resulting weights on the DTU training set.

Following related work [14], we use Adam with decoupled weight decay regularization as optimizer [25] and a cosine learning rate scheduler with warm up for all stages. Training uses up to four NVIDIA A100 80 GB GPUs.

5 Results

We assess the quality of our method’s predictions in two complementary settings: With generalizable implicit methods and with pose-free feed-forward fusion baselines. We further provide analysis to validate our key design decisions.

Comparison with Feed-Forward SDF Baselines

We compare with VolRecon [36] and UFOREcon [31] on the DTU dataset. Given the domain gap between OBJAVERSE, we finetune on their proposed training split (see Sec. 4). Note that this fine-tuning reduces the geometric extrapolation, so that it is limited to potentially visible regions for which a supervision signal is available. Since both methods require known camera parameters, we supply VGGT-predicted poses and intrinsics in place of ground-truth, placing all methods in a fully feed-forward, pose-free setting consistent with our own. We note that this may disadvantage the baselines, which were designed for precise camera input; for completeness, additional comparisons are shown in the supplemental material, supplying VolRecon and UFOREcon with COLMAP-estimated and ground-truth poses, though these settings forfeit the feed-forward assumption and introduce additional computational overhead. Coordinate systems from different modalities are aligned to the ground-truth frame utilizing standard alignment procedures [1, 18, 60]; see the supplemental material for details. In line with UFOREcon [31], we distinguish between sets with substantial view overlap as *favorable* configurations, and *unfavorable* ones, where views are far apart with limited shared visibility, posing a harder problem.

As shown in Tab. 1 and Fig. 4 (and in the supplemental material), we consistently outperform both VolRecon and UFOREcon in the feed-forward setting with unknown poses. We improve Chamfer distance by 29% in both the favorable and unfavorable cases. Qualitatively,

our reconstructions are more complete and less noisy throughout; this is most pronounced in the unfavorable setting. Notably, despite requiring no camera parameters at inference time, we remain competitive with both baselines even when supplied with COLMAP-estimated poses. This suggests that the joint multi-view prior preserved in our volumetric latent space effectively compensates for the absence of explicit camera conditioning.

	favorable (23, 24, 33)			unfavorable (1, 16, 36)		
	↓ CD	D _{GT2P}	D _{P2GT}	CD	D _{GT2P}	D _{P2GT}
VolRecon †	2.905	2.331	3.478	5.781	3.811	7.751
UFOREcon †	2.771	2.101	3.440	3.275	2.252	4.298
VolRecon	3.678	3.369	3.987	8.878	5.869	11.887
UFOREcon	3.415	2.989	3.841	4.942	3.737	6.146
Fus3D	2.432	1.804	3.059	3.525	2.814	4.236

Table 1: Quantitative results with direct SDF prediction baselines in a feed-forward scenario. For reference, † indicated non-feed forward (COLMAP) poses for competing methods.

Comparison with Unposed Predict-Then-Fuse Baselines

We compare our method with the default 2D extraction paradigm of geometry transformers on the OBJAVERSE dataset, isolating the effect of direct volumetric latent extraction versus predict-then-fuse surface reconstruction.

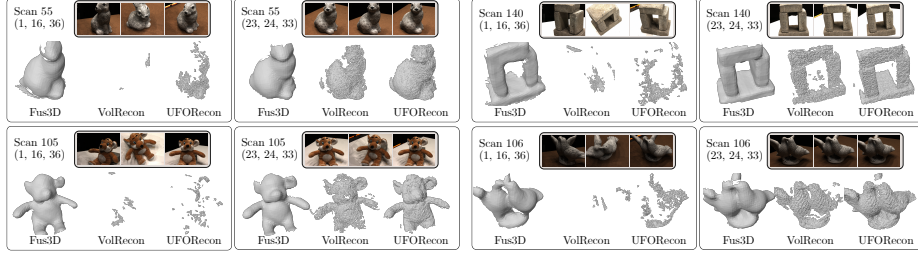


Fig. 4: Qualitative results on DTU. Beige boxes indicate input views. Numbers in brackets correspond to image indices of “favorable” (23, 24, 33) and “unfavorable” (1, 16, 36) view combinations. (Following the baselines’ evaluation protocols, only visible geometry is evaluated.)

For a fair comparison, we fine-tune VGGT on the OBJAVERSE training set (denoted $VGGT_{ft}$), supervising depth maps and camera parameters following our standard training regime, with conventional fine-tuning for the prediction heads and LoRA-based backbone refinement [35]. We compare with two post-hoc fusion strategies for $VGGT_{ft}$ outputs: $VGGT_{ft} + Poisson$, applying Poisson surface reconstruction [15] to predicted and fused point maps, and $VGGT_{ft} + TSDF$, constructing isosurfaces from predicted depth maps and cameras via TSDF fusion [29] followed by marching cubes [60]. For Fus3D, surfaces are likewise extracted via marching cubes. To ensure a fair comparison, all methods are evaluated at matching output resolutions by sampling with the same voxel size ϵ . Reconstructed surfaces are compared against ground truth via F-score and Chamfer distance computed on sampled surface points, following Yariv et al. [56].

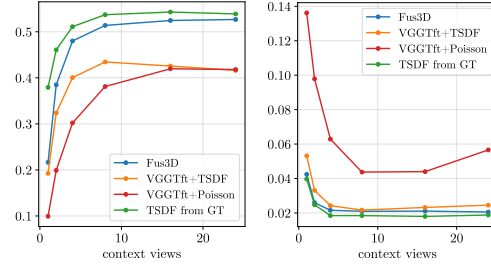


Fig. 5: Comparison vs *ground-truth TSDF isosurfaces* (green) on 170 OBJAVERSE test scenes, w.r.t number of input views. Left: F-scores with threshold 0.5ϵ . Right: Chamfer distances.

Fig. 5 reports these metrics as a function of the number of input views. As OBJAVERSE is a 360° object-centric dataset, large regions remain unobserved in sparse-view settings, directly exposing the limitations of predict-then-fuse approaches: TSDF fusion leaves holes in unobserved regions, and while Poisson reconstruction can close smaller gaps, its extrapolation is limited and not robust. We include a TSDF constructed from ground-truth depth and cameras, *TSDF from GT*, indicating the upper bound achievable without geometric extrapolation. Visual examples are shown in Fig. 6 and 7. Notably, although both Fus3D and $VGGT_{ft}$ are trained on up to 8 views, Fus3D generalizes well to 24 views at inference time, while the predict-then-fuse baselines suffer from accu-

mutating inconsistencies at higher view counts, directly validating the scalability advantage of extracting geometry from the joint latent space rather than fusing per-view predictions.

Tab. 2 reports full metrics at 8 input views across 170 OBJAVERSE test scenes. Fus3D outperforms all baselines on F-score at both thresholds, Chamfer distance, and mean absolute error (MAE) on valid SDF values of X_Ω , with a particularly pronounced advantage on SDF MAE (0.004 vs. 0.023 for $\text{VGGT}_{\text{ft}} + \text{TSDF}$), a direct consequence of regressing the SDF explicitly rather than deriving it from fused depth predictions. $\text{VGGT}_{\text{ft}} + \text{TSDF}$ remains competitive on surface-level metrics but degrades on stricter thresholds ($F_{0.5\epsilon}$: 0.43 vs. 0.51), reflecting residual noise and incompleteness from per-view fusion. $\text{VGGT}_{\text{ft}} + \text{Poisson}$ lags on all metrics, confirming the limited extrapolation capacity of Poisson reconstruction in sparse-view settings. For completeness, we report metrics for non-finetuned baselines in the lower segment which show substantially lower performance compared to the task-adapted baselines. The only metric where Fus3D does not lead is EMD, where $\text{VGGT}_{\text{ft}} + \text{TSDF}$ scores marginally better (0.014 vs. 0.019).

Table 2: Quantitative comparison on the 170 test scenes of the OBJAVERSE dataset for 8 input views. Lower segment shows non-finetuned baselines.

	F_ϵ	$F_{0.5\epsilon}$	CD	EMD	SDF MAE
Fus3D	0.83	0.51	0.021	0.019	0.004
$\text{VGGT}_{\text{ft}} + \text{TSDF}$	0.80	0.43	0.022	0.014	0.023
$\text{VGGT}_{\text{ft}} + \text{Poisson}$	0.70	0.38	0.044	0.052	–
TSDF VGGT	0.35	0.12	0.080	0.082	0.136
Poisson VGGT	0.29	0.10	0.121	0.788	–

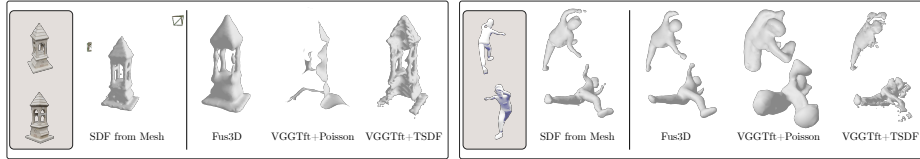


Fig. 6: Visual examples from two input views depicted in the beige boxes.

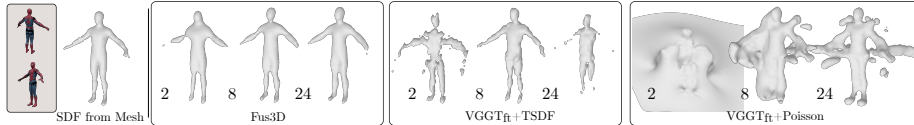


Fig. 7: Visual examples with varying number of input views. Two exemplary views shown on the left.

Ablations and Analysis

Eikonal Validity. Fig. 8 (left) visualizes the Eikonal term on a slice of the predicted SDF $\hat{f}_{X_\Omega}$ and of the TSDF obtained from VGGT_{ft} outputs. TSDFs store fused projective rather than Euclidean distances, causing systematic distance overestimation and Eikonal violations within the truncation band. These artifacts manifest as distorted isosurfaces when extracting at offsets $\neq 0$. Outside the truncation band, the Eikonal condition is recovered by construction through Euclidian redistancing in observed free space [29]. Fus3D, by contrast, predicts

well-behaved SDF values throughout the volume, particularly near and outside the surface. This is reflected quantitatively in Fig. 8 (right): at an isovalue offset of 0.5ϵ , Fus3D achieves an even higher F-score than the TSDF constructed from ground-truth synthetic data, demonstrating that the quality of the predicted distance field, not just surface placement, is a meaningful advantage of direct SDF regression.

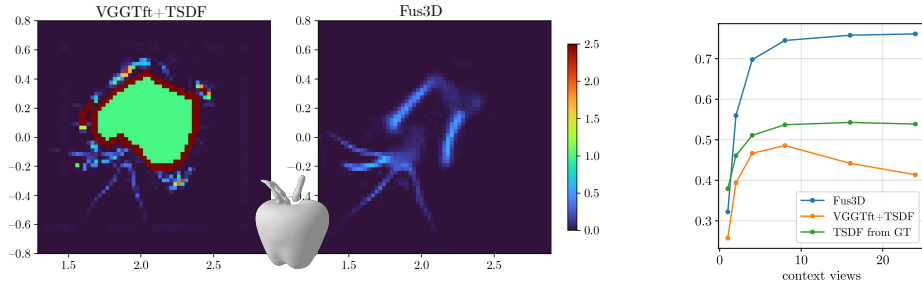


Fig. 8: Left: Visualization of the eikonal term for the center slice of the x-axis. Right: Plot of the $F_{0.5\epsilon}$ -score for surfaces extracted at isovalue $+0.5\epsilon$.

PCA Analysis of $\hat{z}_{3D}$. To probe the structure of the learned volumetric latent space, we do Principal Component Analysis (PCA) on $\hat{z}_{3D}$ across objects from the same category. Fig. 9 shows that the dominant components vary smoothly in space and correlate with the SDF zero-crossing, adapting consistently to changes in object shape and pose, suggesting that the latent volume encodes geometrically meaningful structure rather than unstructured per-voxel features. Fig. 10 further shows that features near the surface tend to align with shared geometric parts across instances of the same category, hinting that the latent space captures some degree of category-level surface structure beyond object-specific occupancy. We note that due to the relatively low resolution of the volumetric grid, these patterns appear at a coarse spatial scale and finer correspondences are not resolved. Nevertheless, the consistent PCA colorings across shapes suggest that the latent volume implicitly organizes geometry in a structured way, which may be beneficial for downstream tasks such as shape recognition or generation.

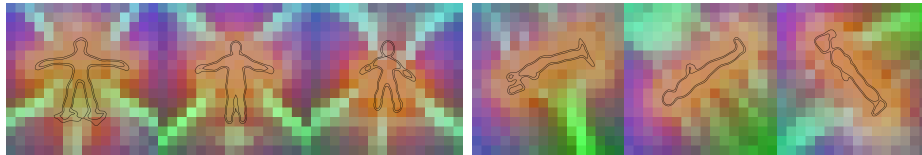


Fig. 9: Visualization of the PCA of $\hat{z}_{3D}$ of 3 different avatar-like assets. Images show the central slices, with the first three components used as RGB channels. The contours indicate the zero-crossing of the SDF min-projection along the visualized axis.

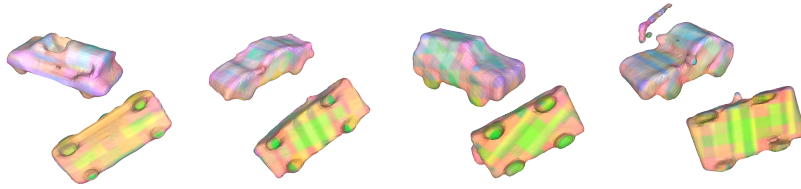


Fig. 10: PCA Visualization of occupied voxels of $\hat{z}_{3D}$ on four car-like assets. First three components are used as RGB channels and visualized as surface colors.

Inference Evaluation. Fig. 11 reports inference times as a function of the number of input views. For VGGT-based baselines, only the relevant heads are evaluated, e.g. $\mathcal{H}_{\text{depth}}$ and $\mathcal{H}_{\text{cam}}$ for TSDF, omitting $\mathcal{H}_{\text{pts}}$. Both post-hoc fusion approaches add substantial overhead to the total reconstruction time, growing with view count. Fus3D introduces a higher constant cost relative to the backbone alone, visible as a larger gap in sparse-view settings; however, this gap narrows as view count increases, since Fus3D omits VGGT’s 2D heads $\mathcal{H}_{\text{depth}}$ and $\mathcal{H}_{\text{pts}}$ entirely, leaving the runtime dominated by the shared per-view feature extraction common to all methods. At higher view counts, Fus3D therefore scales more favourably than the predict-then-fuse baselines.

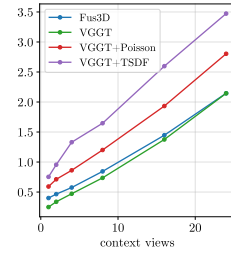


Fig. 11: Inference times in seconds w.r.t the number of input views.

3D Querying. Our model predicts SDF values for the query volume Ω by conditioning the initial learned embedding z_{3D} with its 3D voxel positions in Ω . In the table on the right, we report metrics (bold is better) for a briefly finetuned variant that handles varying query domains (*Fus3D+*), thus allowing for a control mechanism to select the region of interest (see Fig. 12 for a visualization). *Fus3D+ wo. Q* ablates the use of a learned embedding, replacing the initial query entirely by the linearly projected 3D voxel positions. *Default Ω* evaluates the default region of interest. For *Variable Ω* , a volumetric query with random center and extent was evaluated for every scene. Notably, the difference between *Fus3D+* and *Fus3D+ wo. Q* is especially pronounced in case of *Variable Ω* .

	Variable Ω		Default Ω	
	F0.5 ϵ	CD	F0.5 ϵ	CD
Fus3D+	0.281	0.018	0.467	0.021
Fus3D+ wo. Q	0.274	0.021	0.466	0.022

6 Conclusion

Discussion & Future Work. While our reconstructions are dense, complete, and plausible even in unobserved regions, fine-scale details can be missing and surfaces may appear over-smoothed. One contributing factor is the relatively low resolution of our grid. We deliberately use a simple dense decoder to focus on the 3D feature extraction. Its dense output also represents free space which may be

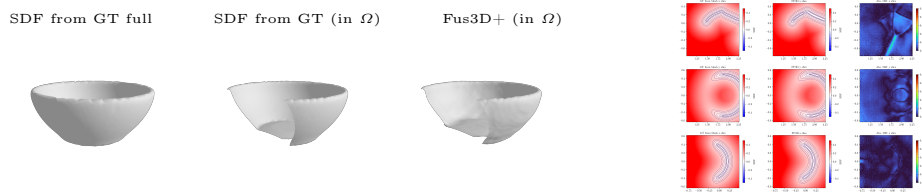


Fig. 12: Varying the query domain Ω , potentially leading to zoomed or cropped results. Left: extracted isosurfaces. Right: slices of the SDF volumes.

valuable for tasks such as trajectory planning or robotic manipulation. For high-fidelity surface extraction, however, sparse representations or multi-scale upsampling are promising directions to improve resolution without prohibitive memory costs. In Sec. 5, we show that Fus3D can be extended to handle shifted and rescaled query domains, which suggests that iterative extraction is a plausible extension. Another promising direction is to more tightly integrate our extractor with the backbone, instead of treating it as a “read-only” feature provider. This is conceptually related to spatial-memory approaches, where the model maintains and updates a consistent state as additional views are processed.

Summary. We present Fus3D, a feed-forward pipeline for dense surface reconstruction that directly regresses a fused signed distance functions from the intermediate feature space of a pretrained multi-view geometry transformer. To enable training on large-scale real-world data, we introduced a validity-aware SDF supervision scheme that handles non-watertight meshes and unobserved regions without expensive or appearance altering preprocessing. By replacing per-view decoder heads with a learned volumetric extraction module, we preserve the joint multi-view prior assembled by the backbone. By doing so, we address two fundamental failure modes of predict-then-fuse pipelines: sparse-view incompleteness and accumulation of inaccuracies under many views. Experiments demonstrate consistent improvements in surface completeness, SDF quality, and scaling behavior over both generalizable SDF baselines and feed-forward fusion pipelines.

Acknowledgments

The authors gratefully acknowledge the scientific support and resources provided by the National High Performance Computing Center of the Friedrich-Alexander-Universität Erlangen-Nürnberg (NHR@FAU) under the projects b212dc, b175dc, and b201dc. NHR funding is provided by federal and Bavarian state authorities. NHR@FAU hardware is partially funded by the German Research Foundation (DFG) – 440719683. Linus Franke was supported in part by the ERC Advanced Grant NERPHYS (101141721). The authors are responsible for the content of this publication.

References

1. 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)
2. Chang, J., Ye, C., Wu, Y., Chen, Y., Zhang, Y., Luo, Z., Li, C., Zhi, Y., Han, X.: Reconviagen: Towards accurate multi-view 3d object reconstruction via generation. arXiv preprint arXiv:2510.23306 (2025)
3. 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)
4. 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)
5. Cheng, Y.C., Lee, H.Y., Tulyakov, S., Schwing, A.G., Gui, L.Y.: Sdfusion: Multi-modal 3d shape completion, reconstruction, and generation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4456–4465 (2023)
6. Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., Farhadi, A.: Objaverse: A universe of annotated 3d objects. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13142–13153 (2023)
7. Ding, Y., Yuan, W., Zhu, Q., Zhang, H., Liu, X., Wang, Y., Liu, X.: Transmvsnet: Global context-aware multi-view stereo network with transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8585–8594 (2022)
8. Gao, Q., Georgiev, I., Wang, T.Y., Singh, K.K., Neumann, U., Yoon, J.S.: Can3tok: Canonical 3d tokenization and latent modeling of scene-level 3d gaussians. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9320–9331 (2025)
9. He, Z., Wang, T.: Openlrm: Open-source large reconstruction models. <https://github.com/3DTopia/OpenLRM> (2023), visited on July 24th 2026
10. Hong, Y., Zhang, K., Gu, J., Bi, S., Zhou, Y., Liu, D., Liu, F., Sunkavalli, K., Bui, T., Tan, H.: Lrm: Large reconstruction model for single image to 3d. In: ICLR (2024)
11. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. *Iclr* **1**(2), 3 (2022)
12. Jensen, R., Dahl, A., Vogiatzis, G., Tola, E., Aanæs, H.: Large scale multi-view stereopsis evaluation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. pp. 406–413. IEEE (2014)
13. Jiang, H., Tan, H., Wang, P., Jin, H., Zhao, Y., Bi, S., Zhang, K., Luan, F., Sunkavalli, K., Huang, Q., et al.: Rayzer: A self-supervised large view synthesis model. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4918–4929 (2025)
14. Jiang, L., Mao, Y., Xu, L., Lu, T., Ren, K., Jin, Y., Xu, X., Yu, M., Pang, J., Zhao, F., et al.: Anysplat: Feed-forward 3d gaussian splatting from unconstrained views. *ACM Transactions on Graphics (TOG)* **44**(6), 1–16 (2025)
15. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: Proceedings of the fourth Eurographics symposium on Geometry processing. vol. 7 (2006)

16. Keetha, N., Müller, N., Schönberger, J., Porzi, L., Zhang, Y., Fischer, T., Knapitsch, A., Zaus, D., Weber, E., Antunes, N., Luiten, J., Lopez-Antequera, M., Bulò, S.R., Richardt, C., Ramanan, D., Scherer, S., Kotschieder, P.: MapAnything: Universal feed-forward metric 3D reconstruction. In: International Conference on 3D Vision (3DV). IEEE (2026)
17. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics* **42**(4), 139–1 (July 2023)
18. Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V.: Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM TOG* **36**(4) (2017)
19. Leroy, V., Cabon, Y., Revaud, J.: Grounding image matching in 3d with mast3r. In: European conference on computer vision. pp. 71–91. Springer (2024)
20. Li, M., Long, X., Liang, Y., Li, W., Liu, Y., Li, P., Luo, W., Wang, W., Guo, Y.: Multi-view large reconstruction model via geometry-aware positional encoding and attention. *IEEE Transactions on Visualization and Computer Graphics* (2025)
21. Li, Z., Müller, T., Evans, A., Taylor, R., Unberath, M., Liu, M., Lin, C.: Neuralangelo: High-Fidelity Neural Surface Reconstruction. In: Conference on Computer Vision and Pattern Recognition (2023)
22. Li, Z., Wang, Y., Zheng, H., Luo, Y., Wen, B.: Sparc3d: Sparse representation and construction for high-resolution 3d shapes modeling. *Advances in Neural Information Processing Systems* **38**, 118582–118600 (2026)
23. Lin, H., Chen, S., Liew, J.H., 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)
24. Long, X., Lin, C., Wang, P., Komura, T., Wang, W.: Sparseneus: Fast generalizable neural surface reconstruction from sparse views. In: European Conference on Computer Vision. pp. 210–227. Springer (2022)
25. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017)
26. Mascaro, R., Chli, M.: Scene representations for robotic spatial perception. *Annual Review of Control, Robotics, and Autonomous Systems* **8**(1), 351–377 (2025)
27. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4460–4470 (2019)
28. 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)
29. Millane, A., Oleynikova, H., Wirbel, E., Steiner, R., Ramasamy, V., Tingdahl, D., Siegwart, R.: nvblox: Gpu-accelerated incremental signed distance field mapping. In: 2024 IEEE International Conference on Robotics and Automation (ICRA). pp. 2698–2705. IEEE (2024)
30. Mittal, P., Cheng, Y.C., Singh, M., Tulsiani, S.: Autosdf: Shape priors for 3d completion, reconstruction and generation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 306–315 (2022)
31. Na, Y., Kim, W.J., Han, K.B., Ha, S., Yoon, S.E.: Uforecon: Generalizable sparse-view surface reconstruction from arbitrary and unfavorable sets. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5094–5104 (2024)

32. Nießner, M., Zollhöfer, M., Izadi, S., Stamminger, M.: Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (ToG)* **32**(6), 1–11 (2013)
33. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., HAZIZA, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.Y., Li, S.W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research* (2024)
34. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 165–174 (2019)
35. Ren, W., Wang, H., Tan, X., Han, K.: Fin3r: Fine-tuning feed-forward 3d reconstruction models via monocular knowledge distillation. *arXiv preprint arXiv:2511.22429* (2025)
36. Ren, Y., Wang, F., Zhang, T., Pollefeys, M., Süssstrunk, S.: Volrecon: Volume rendering of signed ray distance functions for generalizable multi-view reconstruction. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 16685–16695 (2023)
37. Sayed, M., Gibson, J., Watson, J., Prisacariu, V., Firman, M., Godard, C.: Simplexcon: 3d reconstruction without 3d convolutions. In: *Proceedings of the European Conference on Computer Vision (ECCV)* (2022)
38. Schönberger, J.L., Zheng, E., Frahm, J.M., Pollefeys, M.: Pixelwise view selection for unstructured multi-view stereo. In: *ECCV*. pp. 501–518 (2016)
39. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: *Conference on Computer Vision and Pattern Recognition (CVPR)* (2016)
40. Seitz, S., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. In: *Conference on Computer Vision and Pattern Recognition*. pp. 519–528 (2006)
41. Vaswani, A.: Attention is all you need. *Advances in Neural Information Processing Systems* (2017)
42. Wang, H., Agapito, L.: 3d reconstruction with spatial memory. In: *2025 International Conference on 3D Vision (3DV)*. pp. 78–89. IEEE (2025)
43. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt checkpoint. <https://huggingface.co/facebook/VGGT-1B> (2025), visited on July 24th 2026
44. 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)
45. Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W.: Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689* (2021)
46. Wang, Q., Zhang, Y., Holynski, A., Efros, A.A., Kanazawa, A.: Continuous 3d perception model with persistent state. In: *CVPR* (2025)
47. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: Dust3r: Geometric 3d vision made easy. In: *CVPR* (2024)
48. Wang, W., Chen, Y., Zhang, Z., Liu, H., Wang, H., Feng, Z., Qin, W., Zhu, Z., Chen, D.Y., Zhuang, B.: Volsplat: Rethinking feed-forward 3d gaussian splatting with voxel-aligned prediction. *arXiv preprint arXiv:2509.19297* (2025)

49. Wang, Y., Zhou, J., Zhu, H., Chang, W., Zhou, Y., Li, Z., Chen, J., Pang, J., Shen, C., He, T.: π^3 : Permutation-equivariant visual geometry learning. arXiv preprint arXiv:2507.13347 (2025)
50. Wang, Y., Han, Q., Habermann, M., Daniilidis, K., Theobalt, C., Liu, L.: Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2023)
51. Wei, X., Zhang, K., Bi, S., Tan, H., Luan, F., Deschaintre, V., Sunkavalli, K., Su, H., Xu, Z.: Meshlrn: Large reconstruction model for high-quality mesh. arXiv preprint arXiv:2404.12385 (2025)
52. Xiang, J., Lv, Z., Xu, S., Deng, Y., Wang, R., Zhang, B., Chen, D., Tong, X., Yang, J.: Structured 3d latents for scalable and versatile 3d generation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 21469–21480 (2025)
53. 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)
54. Yao, Y., Luo, Z., Li, S., Fang, T., Quan, L.: Mvsnet: Depth inference for unstructured multi-view stereo. In: Proceedings of the European conference on computer vision (ECCV). pp. 767–783 (2018)
55. Yariv, L., Kasten, Y., Moran, D., Galun, M., Atzmon, M., Ronen, B., Lipman, Y.: Multiview Neural Surface Reconstruction by Disentangling Geometry and Appearance. In: Advances in Neural Information Processing Systems (2020)
56. Yariv, L., Puny, O., Gafni, O., Lipman, Y.: Mosaic-sdf for 3d generative models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4630–4639 (2024)
57. Ye, B., Chen, B., Xu, H., Barath, D., Pollefeys, M.: Yonosplat: You only need one model for feedforward 3d gaussian splatting. arXiv preprint arXiv:2511.07321 (2025)
58. Zhang, C., Zou, Y., Li, Z., Yi, M., Wang, H.: Transplat: Generalizable 3d gaussian splatting from sparse multi-view images with transformers. arXiv Preprint (2024)
59. Zhang, C., Song, H., Wei, Y., Yu, C., Lu, J., Tang, Y.: Geolrm: Geometry-aware large reconstruction model for high-quality 3d gaussian generation. Advances in Neural Information Processing Systems **37**, 55761–55784 (2024)
60. Zhou, Q.Y., Park, J., Koltun, V.: Open3D: A modern library for 3D data processing. arXiv:1801.09847 (2018)
61. Zhuang, P., Han, S., Wang, C., Siarohin, A., Zou, J., Vasilkovsky, M., Shakhrai, V., Korolev, S., Tulyakov, S., Lee, H.Y.: Gtr: Improving large 3d reconstruction models through geometry and texture refinement. In: International Conference on Learning Representations. vol. 2025, pp. 20466–20482 (2025)