UniQueR: Unified Query-based Feedforward 3D Reconstruction

Chensheng Peng[1,2*], Quentin Herau[1], Jiezhi Yang[1], Yichen Xie[1,2*], Yihan Hu[1], Wenzhao Zheng[2], Matthew Strong[1,3*], Masayoshi Tomizuka[2], and Wei Zhan[1,2†]

[1]Applied Intuition [2]UC Berkeley [3]Stanford University

Abstract. We present UniQueR, a unified query-based feedforward framework for efficient and accurate 3D reconstruction from unposed images. Existing feedforward models such as DUSt3R, VGGT, and AnySplat typically predict per-pixel point maps or pixel-aligned Gaussians, which remain fundamentally 2.5D and limited to visible surfaces. In contrast, UniQueR formulates reconstruction as a sparse 3D query inference problem. Our model learns a compact set of 3D anchor points that act as explicit geometric queries, enabling the network to infer scene structure—including geometry in occluded regions—in a single forward pass. Each query encodes spatial and appearance priors directly in global 3D space (instead of per-frame camera space) and spawns a set of 3D Gaussians for differentiable rendering. By leveraging unified query interactions across multi-view features and a decoupled cross-attention design, UniQueR achieves strong geometric expressiveness while substantially reducing memory and computational cost. Experiments on Mip-NeRF 360 and VR-NeRF demonstrate that UniQueR surpasses recent feedforward baselines in both rendering quality and geometric accuracy, using an order of magnitude fewer primitives than dense alternatives.

Keywords: Feedforward reconstruction · Gaussian splatting · Queries

1 Introduction

3D reconstruction from 2D captures is a fundamental task in computer vision [58], with wide-ranging applications in robotics, autonomous driving, and digital content creation. The goal is to recover accurate 3D geometry and appearance from one or more 2D observations, enabling machines to perceive and interact with the world in a spatially coherent manner.

Traditional methods such as Structure-from-Motion (SfM) [48], Multi-View Stereo (MVS) [74], and Simultaneous Localization and Mapping (SLAM) [7] rely on geometric optimization and feature matching to estimate camera poses and dense point clouds. While effective under controlled settings, they often struggle with visual ambiguity, sparse viewpoints, or textureless regions.

Project page: `https://uniquer3d.github.io/`

* Work done during an internship at Applied Intuition.

** Corresponding author: `wei.zhan@applied.co`.

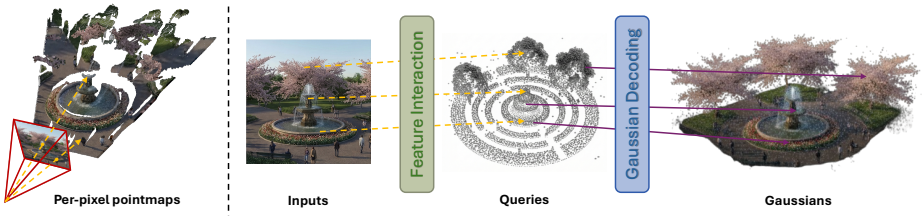


Fig. 1: Illustration of image-aligned and query-based pipelines. Given input images, UniQueR predicts a sparse set of 3D queries that spawn Gaussians in a shared scene coordinate system. The query-based prediction can allocate primitives beyond observed input surfaces. Unlike pixel-aligned methods (e.g., AnySplat) that produce holes in unobserved areas, our query-based representation enables more complete 3D reconstruction with accurate geometry.

With the rise of deep learning, optimization-based representations—notably Neural Radiance Fields (NeRFs) [41] and 3D Gaussian Splatting [34]—have produced impressive visual fidelity and multi-view consistency. However, their per-scene optimization nature limits scalability, generalization, and runtime efficiency. Such methods cannot leverage large-scale data to learn transferable geometric priors, and reconstruction for a new scene remains computationally expensive. Meanwhile, the research community has increasingly explored feedforward 3D reconstruction frameworks that support real-time inference, end-to-end learning, and data-driven geometric priors. Unlike optimization-based methods that require per-scene fitting, these architectures learn a shared reconstruction model across large-scale datasets, making single-pass inference possible on in-the-wild captures.

Existing works such as Dust3R [62], VGGT [58], and Pi3 [63] demonstrate that 3D geometry can be recovered directly from 2D observations in a single forward pass. These models typically extract multi-view features using transformers or cost volumes and predict intermediate 2.5D representations such as depth maps, normal maps, or point maps. Large-scale training with depth priors enables these models to transfer learned geometry across scenes. This paradigm shift highlights the potential of feedforward architectures to serve as general-purpose 3D perception backbones for robotics and embodied AI.

Building on this idea, recent methods have extended feedforward prediction beyond point-based outputs to more expressive representations. Approaches such as MVSplat [11], NoPoSplat [77], FLARE [85] and AnySplat [29] generate sets of 3D Gaussians directly from images, representing both geometry and appearance in a compact and differentiable form. These Gaussian-based outputs allow efficient rendering with photorealistic details and natural view-dependent effects, bridging the gap between feedforward prediction and neural representations. Moreover, they unify geometry and appearance modeling under a single representation, enabling downstream tasks like relighting, editing, and novel view synthesis.

However, despite their impressive progress, these feedforward methods still rely on 2.5D representations: they predict surface-aligned Gaussians or camera-view

features that only capture the visible surfaces within the observed viewpoints. As a result, they remain inherently view-anchored. Because the learned features are tied to specific camera projections, these models struggle to allocate geometry to occluded or unseen regions, leading to holes and artifacts in novel views that deviate from the input cameras.

To address this limitation, we propose UniQueR, a Unified Query-based Feedforward 3D Reconstruction framework that directly operates in 3D space. Inspired by query-based architectures from object detection and scene understanding (e.g., DETR [9], DETR3D [61], PETR [39]), we introduce learnable 3D queries as the core scene representation. Each query acts as a spatial anchor in 3D, encoding geometric and appearance attributes. Without requiring ground-truth 3D supervision, we leverage Gaussian Splatting [34] as a differentiable rendering bridge between 3D queries and 2D observations. Through a cross attention mechanism, the learnable 3D queries interact with the multi-view images and integrate features into their own embeddings. Each query then spawns a set of Gaussians, which are rendered to 2D images with color and depth—enabling supervision from RGB images and depth maps.

Crucially, our training procedure supervises the spawned Gaussians using novel views beyond the input cameras. For example, with 2 input views, we would rasterize the inferred Gaussians to 4 novel poses and supervise the renderings with corresponding ground truth images. Such paradigm encourages 3D queries to not only recover the observed surfaces from inputs but also allocate Gaussians to regions not directly observed. Combined with a decoupled cross attention module for efficient feature interaction, UniQueR can reconstruct more complete scenes even under sparse or partial observations with a faster speed.

Our main contributions are as follows:

- We introduce UniQueR, a scene-level feedforward 3D reconstruction framework based on learnable 3D queries that decouple geometry from input viewpoints, enabling the model to place Gaussians in unobserved regions.
- We devise a decoupled cross-attention mechanism for integrating per-view image features into global learnable queries, which scales efficiently to a large number of input views.

We demonstrate comparable novel-view synthesis *and* geometric accuracy on Mip-NeRF 360 [4] and VR-NeRF [70], using an order of magnitude fewer Gaussians and less memory than dense feedforward baselines.

2 Related Work

Optimization-based 3D reconstruction. Traditional pipelines for 3D reconstruction and novel view synthesis typically follow two sequential stages. The first estimates camera parameters using geometric and multi-view stereo methods such as Structure-from-Motion (SfM) [2, 13, 48, 51, 52, 66, 67] and stereo matching [18, 19, 49]. SfM systems—most notably COLMAP [48]—align images via feature correspondences to generate sparse point clouds, refined through bundle

Table 1: Comparison of representations for 3D reconstruction. ✓ = favorable, ▬ = moderate, ✗ = unfavorable. *Volumetric*: can represent geometry beyond visible surfaces (not pixel-aligned). *Fast Inference*: single forward pass without per-scene optimization. *Low Memory*: compact representation. UniQueR achieves the best overall trade-off.

	Representation	Methods	Volumetric	Fast Inference	Low Memory
Per-scene Optimization	GS	3D GS, Mip-Splatting	✓	✗	▬
	NeRF	Mip-NeRF	✓	✗	✓
Feedforward	Pixel-aligned Pointmaps	VGGT, Pi3	✗	✓	✓
	Voxel-aligned GS	AnySplat	▬	✓	▬
	Voxel-aligned NeRF	DistillNeRF	✓	✗	▬
	Queries (Ours)	**UniQueR**	✓	✓	✓

adjustment [1, 56, 68, 80]. Related approaches such as V-SLAM [14, 36, 43] extend these ideas to real-time tracking and mapping. Although robust, these pipelines incur high computational cost due to hand-crafted feature extraction, iterative correspondence search, and large-scale nonlinear optimization [44], leading to significant runtime overhead and limited scalability.

Following geometry-based reconstruction, per-scene optimization has become standard for novel view synthesis. Differentiable rendering methods like NeRF [41] and its variants [5, 6, 21–24, 26–28, 34, 59, 72, 73, 76] optimize neural scene representations per scene given known camera poses. While highly accurate, they require minutes to hours of optimization for a single scene.

Recent approaches such as hash-encoded fields (InstantNGP) [42] and 3D Gaussian Splatting [34] substantially reduce training time, but still depend on nontrivial per-scene optimization, limiting their practicality for real-time or large-scale reconstruction.

Feedforward reconstruction. By leveraging large-scale 3D datasets, recent feedforward 3D reconstruction methods directly regress pixel-aligned point maps or Gaussian primitives from posed multi-view images. Splatter Image [53] predicts pixel-aligned Gaussians from a single image, while PixelSplat [10] extends this formulation to stereo pairs. MVSplat [11] further generalizes to variable-length sequences but still relies on known input poses. In parallel, feedforward models for monocular or multi-view depth estimation [15, 17, 32, 47, 74, 79], optical flow [55], and point tracking [16, 31, 69] have steadily advanced the ability to learn geometry from raw visual input. DUSt3R [62] maps image pairs to dense point maps followed by global alignment, while MASt3R [37] improves matching through dense local descriptors. Building on these foundations, Splatt3R [50] and NoPoSplat [77] directly regress Gaussian primitives from uncalibrated image pairs. Spann3R [57], MUSt3R [8], and PreF3R [12] incorporate memory or recurrent processing for longer sequences. GS-LRM [83] instead predicts per-pixel Gaussians from posed sparse views and demonstrates both object- and scene-level reconstruction, making it a direct precursor in feedforward Gaussian decoding rather than an exclusively object-centric method.

A recent line of work jointly infers camera parameters and 3D structure. VGGT [58] and AnySplat [29] provide end-to-end pose-free reconstruction and

rendering, while MapAnything [33] supports a broader set of geometric outputs. These methods obtain strong observed-surface priors from image- or voxel-aligned predictions, but their output density and coverage remain coupled to the input views. Our work studies whether a fixed set of explicit 3D queries can provide a complementary scene-level representation under sparse and uneven coverage.

Anchor- and query-based 3D representations. Scaffold-GS [40] uses optimized anchors to spawn local Gaussians for per-scene reconstruction. In multi-view 3D detection, DETR3D [64] and PETR [39] associate explicit 3D object queries with image features, while Point-DETR3D [20] studies positional query initialization from point priors. These methods decode a sparse set of supervised objects rather than a dense renderable scene. LRM [25], LGM [54], and 3DShape2VecSet [82] use learned tokens for object reconstruction or generation. LVSM [30] and SceneTok [3] more closely match the *role* of a compact scene-token representation, but their tokens are latent and are decoded into target views or generative scene representations. In UniQueR, each token carries an explicit 3D coordinate and spawns a local Gaussian cluster, while its DETR-like cross-attention supplies the *mechanism* for collecting multi-view evidence.

Concurrent work explores complementary directions. AnchorSplat [86] uses 3D geometric anchors and a Gaussian refiner for scene-level feedforward reconstruction, whereas TokenGS [46] decouples 3D Gaussian prediction from pixels with learnable implicit tokens. Tab. 1 summarizes the factual representation differences most relevant to our setting.

3 UniQueR

We propose UniQueR, an efficient framework based on sparse queries for fast reconstruction from unposed images. We start with the problem statement and notations in Sec. 3.1, followed by the network structure in Sec. 3.2, and the training procedure in Sec. 3.3.

3.1 Notations

Given N RGB images $\{\mathbf{I}_i\}_{i=1}^{N}$ of the same scene, with $\mathbf{I}_i \in \mathbb{R}^{3\times H\times W}$, UniQueR associates their features with Q learnable queries $\mathcal{Q} = \{\mathbf{q}_i\}_{i=1}^{Q}$. Feedforward models predict camera-to-world transforms and per-view local point maps; applying each predicted transform to its local point map yields global points in one shared per-scene gauge. This coordinate system is non-metric and is not a first-camera frame. Queries, spawned Gaussians, and rendering cameras all use the same gauge, so a global similarity change does not alter the rendered supervision. Each query carries a location $\mathbf{p}_i$ in this frame and spawns K Gaussians through learned local offsets. The resulting set $\mathcal{G} = \{\mathbf{g}_{ik} \mid i = 1,\dots,Q;\ k = 1,\dots,K\}$ contains QK colored Gaussian primitives [34] and is rendered as $\hat{\mathbf{I}} = \texttt{Render}(\mathcal{G}, \pi)$ for a camera $\pi \in \mathbb{SE}(3)$.

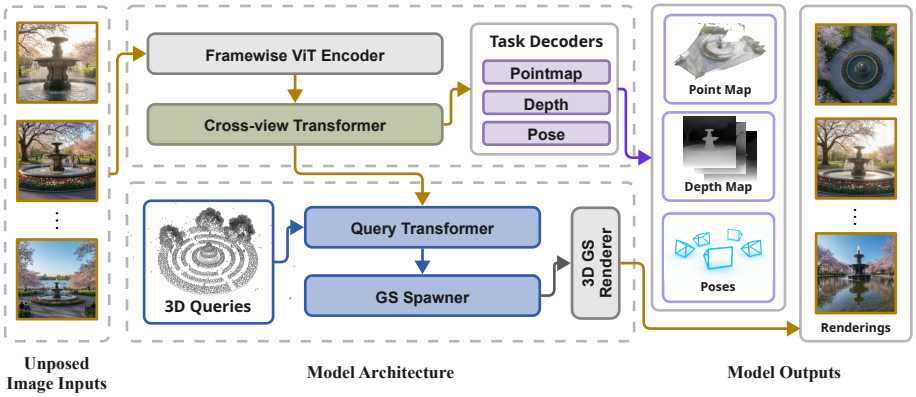


Fig. 2: UniQueR pipeline overview. Given multi-view images, a ViT encoder with alternating attention extracts per-frame tokens and decodes camera poses and point maps. A set of 3D queries is refined through cross-attention with image tokens and self-attention among queries. Each query then spawns K Gaussians, which are rendered via differentiable splatting for RGB and depth supervision.

In addition, following the formulation of VGGT [58] and Pi3 [63], the image input is also mapped to a corresponding set of per-frame 3D annotations:

$$f\big(\{\mathbf{I}_i\}_{i=1}^{N}\big) = \{(\mathbf{P}_i, \mathbf{X}_i, \mathbf{C}_i)\}_{i=1}^{N}, \tag{1}$$

where for each frame $\mathbf{I}_i$, the transformer predicts (1) a camera-to-world pose $\mathbf{P}_i \in \mathbb{SE}(3)$, (2) a local point map $\mathbf{X}_i \in \mathbb{R}^{3\times H\times W}$, and (3) a confidence map $\mathbf{C}_i \in \mathbb{R}^{H\times W}$.

3.2 Query-based Reconstructor

Rather than relying on dense 3D representations such as voxels or large sets of Gaussians, which are computationally expensive and scale poorly, we adopt a sparse set of learnable 3D queries as our core scene representation. Each query corresponds to a point in 3D space with an associated high-dimensional latent embedding. This sparse representation captures the coarse structure and appearance of the scene while enabling efficient feedforward inference. An overview of the UniQueR pipeline is shown in Fig. 2.

Image tokenization. Given N unposed images $\mathbf{I}_i \in \mathbb{R}^{3\times H\times W}$, we extract per-frame visual tokens with a Vision Transformer (ViT). We use DINOv2 as the backbone for its strong semantic representations:

$$\mathbf{T}_i = \texttt{DINO}(\mathbf{I}_i) \in \mathbb{R}^{HW/p^2\times d} \tag{2}$$

where p is the patch size and d is the token dimensionality.

To aggregate cross-view information, we follow VGGT [58] and Pi3 [63] and apply an alternating-attention (AA) transformer, which alternates between intra-frame and inter-frame attention operations:

$$\{\mathbf{T}_i^{l+1}\} = \texttt{AA-Transformer}(\{\mathbf{T}_i^{l}\}) \tag{3}$$

where l denotes the transformer-layer index.

The aggregated image tokens serve two purposes. First, they are decoded into per-frame geometric annotations—camera poses $\mathbf{P}_i$, point maps $\mathbf{X}_i$, and confidence maps $\mathbf{C}_i$:

$$\{\mathbf{P}_i, \mathbf{X}_i, \mathbf{C}_i\} = \texttt{Decoder}(\{\mathbf{T}_i\}) \tag{4}$$

These predicted geometric attributes provide strong geometric priors that guide the subsequent update of 3D queries, enabling more accurate and consistent reconstruction for the observed surfaces.

Query Initialization. Each query has an explicit 3D coordinate and latent embedding. Detection frameworks such as DETR [9] and DETR3D [64] can learn sparse object queries with direct box supervision. In our dense reconstruction setting, random-only query coordinates are unstable because 2D rendered image losses rather than ground-truth 3D boxes are used during training.

To address this, we adopt a hybrid initialization strategy. Half of the query coordinates are sampled from the predicted non-metric point maps, providing geometric grounding on observed surfaces. The remaining learnable anchors are initialized uniformly in the normalized 3D scene range. Uniformity applies only to their initialization: after transformer refinement and the predicted query deformation, they are free to move away from the initial samples. The final deformed query is the center of a local Gaussian cluster, and neither the query centers nor the spawned Gaussians are constrained to be uniformly distributed on a surface.

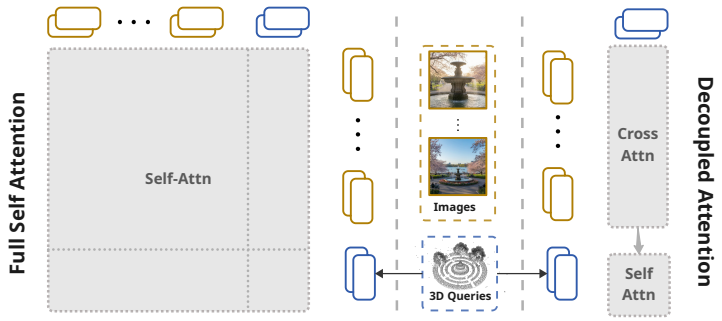


Fig. 3: Architectural comparison of full self-attention over concatenated image/query tokens and the adopted decoupled design with query-to-image cross-attention followed by query-only self-attention.

Query propagation. After query initialization, the image tokens from the tokenizer $\mathcal{T} = \{\mathbf{T}_i\}_{i=1}^{N}$, together with the query tokens $\mathcal{Q} = \{\mathbf{q}_i\}_{i=1}^{Q}$, are fed into the Query Transformer to update the query tokens:

$$\mathcal{Q} = \textsc{QueryTransformer}(\mathcal{Q}, \mathcal{T}) \tag{5}$$

We use a fixed budget of $Q = 4096$ queries and $K = 64$ Gaussians per query in all experiments except the query-count ablation in Fig. 5a. This budget is independent of the number and resolution of the input views, although the image-token count still grows linearly with the number of views.

As illustrated in Fig. 3, we consider two variants of the Query Transformer. The most straightforward approach is to concatenate the query tokens and image tokens and apply full self-attention as follows:

$$\mathcal{Q}, \mathcal{T} = \textsc{Self-Attn}([\mathcal{Q}, \mathcal{T}]) \tag{6}$$

where the output image tokens will be discarded.

However, this results in a computational complexity of $\mathcal{O}\left(\left(Q + NHW/p^2\right)^2\right)$, which is prohibitively expensive for high-resolution inputs.

We therefore use 12 Query Transformer layers with a decoupled attention design. In each layer, cross-attention first updates queries from image tokens, followed by self-attention among the queries:

$$\mathcal{Q} = \textsc{Cross-Attn}(\mathcal{Q}, \mathcal{T}), \mathcal{Q} = \textsc{Self-Attn}(\mathcal{Q}) \tag{7}$$

Writing $T = HW/p^2$ for the number of tokens per image, this changes the theoretical attention complexity from $\mathcal{O}((Q + NT)^2)$ to $\mathcal{O}(QNT + Q^2)$, yielding substantial memory savings and enabling the model to scale up to larger size, high-resolution images and more queries. The end-to-end memory and timing comparison in Sec. 4 also reflects the representation and decoder.

For the image tokens, we convert the predicted camera poses into Plücker ray embeddings, which serve as positional encodings. Each query naturally carries its own 3D coordinate as its positional embedding. This 3D-aware positional parameterization enables more meaningful geometric interaction between image tokens and queries, improving spatial reasoning and reconstruction fidelity.

GS spawning. Without the reliance on ground truth 3D supervision, we leverage Gaussian Splatting as a differentiable bridge between 3D geometry and 2D observations. By rendering 3D primitives onto the image plane, we can supervise reconstruction using abundant 2D signals—RGB and depth maps—which are far more available than explicit 3D annotations.

Each query $\mathbf{q}_i$ at location $\mathbf{p}_i$ spawns K Gaussians $\{\mathbf{g}_{ik}\}_{k=1}^{K}$. Its updated embedding first predicts a deformation $\delta\mathbf{q}_i$, so $\mathbf{p}_i + \delta\mathbf{q}_i$ becomes the cluster center. An MLP then decodes K local offsets $\delta\mathbf{g}_{ik}$ around that center, while attribute heads predict opacity, scale, rotation, and color.

$$\mathcal{G} = \bigcup_{i=1}^{Q} \bigcup_{k=1}^{K} \left(\mathbf{p}_i + \delta\mathbf{q}_i + \delta\mathbf{g}_{ik}, \Sigma_{ik}, \mathbf{c}_{ik}, \alpha_{ik}\right), \tag{8}$$

The resulting Gaussian set $\mathcal{G}$ can then be rendered into RGB and depth efficiently via a GS rasterizer. Our design preserves the efficiency of sparse 3D queries while enabling the spawned dense Gaussians to model fine-grained geometric and appearance details, yielding high-fidelity reconstructions at low computational cost.

3.3 Training

Losses. The Gaussians $\mathcal{G}$ are rendered into RGB images $\hat{\mathbf{I}}$ and depth maps $\hat{\mathbf{D}}$ through differentiable splatting. In Stage 1, $\mathcal{L}_{\text{rgb}}$ includes an ℓ_1 reconstruction term and an LPIPS perceptual term. For high-resolution Stage 2 fine-tuning, we omit the LPIPS term and use only the ℓ_1 term because LPIPS substantially increases training time. Both stages also optimize the scale-invariant rendered-depth loss $\mathcal{L}_{\text{depth}}$ and the backbone camera loss $\mathcal{L}_{\text{cam}}$. The total loss is

$$\mathcal{L} = \mathcal{L}_{\text{rgb}} + \lambda_d \mathcal{L}_{\text{depth}} + \lambda_c \mathcal{L}_{\text{cam}} \tag{9}$$

We set $\lambda_d = 0.1$ and $\lambda_c = 0.2$. Table 7 evaluates the 224^2 Stage 1 model, whereas Tables 2 and 5 use the 448^2 Stage 2 model.

Novel-view supervision. The supervision views are a *superset* of the input views. Given three inputs, for example, we supervise six renderings: the three inputs and three held-out training views. Missing appearance or geometry visible in those held-out targets contributes to the rendering loss, so this objective is intended to discourage reproducing only input-view surfaces. If the model only focuses on reconstructing the observed areas from the input views, empty holes will present in the novel viewpoints, and such artifacts will be corrected with the target novel images as supervision.

Staged training. The DINOv2 encoder, alternating-attention transformer, and point-map head are initialized from Pi3 [63] and kept frozen. We update the camera head, 12-layer Query Transformer, and Gaussian spawning/attribute heads, for 502.31M trainable parameters; the frozen backbone contains 890.99M parameters (1.393B total). We train with AdamW (learning rate 10^{-4} with cosine decay) for 100 epochs at 224^2 on 32 A100 GPUs, then fine-tune for 20 epochs at 448^2. Each sample contains 2–64 input views, while the query budget remains $Q = 4096$.

3.4 Post Optimization

Following AnySplat [29], rendering quality can be further improved through test-time optimization (TTO). Using our predicted Gaussians as initialization and predicted camera poses as input, we render the Gaussians back into the input views and compute reconstruction losses against the original images. Backpropagating these losses refines the Gaussian parameters. We evaluate this protocol in Sec. 4 under the dense-view setting.

Table 2: Sparse-view novel-view synthesis on held-out target views. We report PSNR, SSIM, LPIPS, and the inference time in seconds for a single feedforward pass from images to Gaussians. Best and second-best results are shown in **bold** and underline, respectively.

Dataset	Method	3 Views				6 Views			
		PSNR↑	SSIM↑	LPIPS↓	Time (s)↓	PSNR↑	SSIM↑	LPIPS↓	Time (s)↓
Mip-NeRF	NoPoSplat [77]	18.21	0.482	0.426	0.416	16.06	0.423	0.540	2.121
	AnySplat [29]	20.08	0.606	0.274	0.279	18.29	0.518	0.336	0.646
	UniQueR	**22.70**	**0.660**	**0.261**	**0.213**	**21.80**	**0.622**	**0.300**	**0.447**
VR-NeRF	NoPoSplat [77]	21.28	0.744	0.381	0.406	20.69	**0.728**	0.426	2.176
	AnySplat [29]	19.67	**0.745**	**0.313**	0.290	17.25	0.683	**0.411**	0.656
	UniQueR	**21.99**	0.708	0.446	**0.198**	**20.87**	0.689	0.431	**0.412**

Table 3: Novel-view synthesis on 150 randomly sampled scenes from the RealEstate10K test split used by NoPoSplat, with two input and two disjoint target views per scene. All methods use the same sampled scenes and views.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
FLARE [85]	17.74	0.586	0.371
AnySplat [29]	18.93	0.615	0.226
UniQueR	**20.50**	**0.672**	**0.196**

4 Experiments

4.1 Experimental Setups

Datasets. We use the data-processing pipelines from MapAnything [33] and CUT3R [60]. Training uses DL3DV-10K [38], ScanNet++ [78], BlendedMVS [75], and Co3D [45]. We evaluate NVS on Mip-NeRF 360 [4], VR-NeRF [70], and a 150-scene RealEstate10K [87] subset. Camera-pose evaluation uses RealEstate10K and Co3Dv2 [45].

Evaluation Metrics and Protocol. We adopt PSNR, SSIM [65], and LPIPS [84] for novel-view synthesis, and depth abs-rel error for geometric accuracy. For pose estimation, we follow Pi3 [63] using RRA, RTA, and AUC at a 30-degree threshold.

For Mip-NeRF 360 and VR-NeRF sparse-view evaluation, input and test views are disjoint. We average five input-view subsets per scene and compute metrics only on their held-out targets. For RealEstate10K, we randomly sample 150 scenes from the test split used by NoPoSplat [77], with two input and two disjoint target views per scene. Every method uses the same scenes and view selections. Inference time is the end-to-end latency from images to Gaussians on one A100 80GB GPU.

Baselines. We compare UniQueR against a set of methods for 3D scene reconstruction, novel view synthesis, and camera pose estimation. For novel view

Table 4: Camera Pose Estimation on RealEstate10K [87] and Co3Dv2 [45].

Method	RealEstate10K			Co3Dv2		
	RRA@30 ↑	RTA@30 ↑	AUC@30 ↑	RRA@30 ↑	RTA@30 ↑	AUC@30 ↑
Fast3R [71]	99.05	81.86	61.68	97.49	91.11	73.43
CUT3R [60]	99.82	95.10	81.47	96.19	92.69	75.82
FLARE [85]	99.69	95.23	80.01	96.38	93.76	73.99
VGGT [58]	<u>99.97</u>	93.13	77.62	98.96	97.13	**88.59**
Pi3 [63]	**99.99**	**95.62**	**85.90**	**99.05**	<u>97.33</u>	88.41
UniQueR	**99.99**	<u>95.44</u>	<u>83.69</u>	**99.05**	**97.44**	<u>88.52</u>

synthesis, we include NoPoSplat [77] and multi-view, transformer-based methods such as AnySplat [29]. To evaluate reconstruction quality, we also test the per-scene optimization performance using 3DGS [35] and MipSplatting [81] with different GS initializations and predicted camera poses from VGGT [58], AnySplat [29], and UniQueR. For camera pose estimation, we benchmark against Fast3R [71], CUT3R [60], FLARE [85], VGGT [58], and Pi3 [63].

Implementation details. Training details are described in Sec. 3.3. We use $Q = 4096$ queries with $K = 64$ Gaussians per query by default, resulting in 262K total Gaussians. All runtime measurements use a single A100 80GB GPU with batch size 1. We follow MapAnything [33] for dynamic resolution during training with aspect ratios 1.0, 1.33, 1.52, 1.77, and 2.0.

4.2 Quantitative Results

Following AnySplat [29], we evaluate sparse inputs (3 or 6 views) and dense inputs (32 or 64 views). Tab. 2 shows that UniQueR obtains the best PSNR among the compared feedforward methods for both sparse-view benchmarks. The metric picture is mixed on VR-NeRF: AnySplat has higher SSIM and lower LPIPS for three views, and lower LPIPS for six views. Tab. 3 summarizes the two-view RealEstate10K evaluation, where UniQueR leads the compared methods on all three metrics.

Under the dense-setting evaluation (Tab. 5), our approach achieves competitive feedforward results and provides a significantly better initialization for per-scene optimization (3DGS+Ours, MipSplatting+Ours). We attribute the gap in the feedforward-only dense setting to the inherent sparsity of our representation: we use up to $4096 \times 64 = 262$K Gaussians, significantly fewer than per-pixel methods which generate $448 \times 448 \times N$ primitives. Despite this, UniQueR provides substantially better initialization quality, as shown by the large gains in the post-optimization rows.

The evaluation results on camera pose estimation are detailed in Tab. 4. The performance is comparable to Pi3 [63]. We attribute the slight gap to more limited training data since Pi3 uses private dataset during the training.

Table 5: Quantitative comparison for dense-view novel-view synthesis. We report PSNR, SSIM, and LPIPS. Best and second-best are in **bold** and underline. Top rows: feedforward only. Bottom rows: feedforward initialization followed by per-scene 3DGS/MipSplatting optimization.

Dataset	Method	32 Views			64 Views		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Mip-NeRF 360	AnySplat [29]	**22.32**	**0.660**	**0.258**	21.26	0.607	**0.303**
	Ours	21.51	0.619	0.325	**21.58**	**0.641**	0.335
	3DGS+VGGT [35,58]	22.61	0.701	0.233	23.78	0.729	0.221
	3DGS+AnySplat [29,35]	24.99	0.705	0.227	23.71	0.664	0.266
	3DGS+Ours [35]	**25.14**	**0.774**	**0.183**	**26.00**	**0.784**	**0.176**
	MipSplatting+VGGT [58,81]	22.27	0.686	0.247	23.72	0.721	0.230
	MipSplatting+AnySplat [29,81]	24.98	0.710	0.223	23.84	0.675	0.257
	MipSplatting+Ours [81]	**25.26**	**0.772**	**0.184**	**25.99**	**0.782**	**0.178**
VR-NeRF	AnySplat [29]	**23.21**	**0.773**	**0.308**	23.62	0.802	**0.285**
	Ours	23.18	0.770	0.395	**24.74**	**0.808**	0.334
	3DGS+VGGT [35,58]	21.74	0.720	0.352	23.18	0.748	0.315
	3DGS+AnySplat [29,35]	21.90	0.708	0.377	24.27	0.768	0.306
	3DGS+Ours [35]	**27.03**	**0.846**	**0.223**	**28.56**	**0.879**	**0.174**
	MipSplatting+VGGT [58,81]	21.79	0.715	0.362	23.50	0.759	0.308
	MipSplatting+AnySplat [29,81]	21.93	0.711	0.373	24.08	0.764	0.311
	MipSplatting+Ours [81]	**26.62**	**0.838**	**0.237**	**28.39**	**0.873**	**0.185**

Table 6: End-to-end efficiency and rendered-depth comparison with 32 input views at 448^2 on one A100 80GB (batch size 1). UniQueR uses 14.7× fewer Gaussians, 39% less memory, and 2.35× less time than AnySplat.

Method	# Gaussians	GPU Mem.	Time (s)	Depth Abs Rel ↓
AnySplat	3.85M	18.42 GB	4.63	0.062
UniQueR	262K	11.19 GB	1.97	0.038

It's noteworthy that in Tab. 5, 3DGS+AnySplat occasionally underperforms standalone AnySplat on VR-NeRF. We attribute this to inaccurate pose estimates from AnySplat degrading the per-scene optimization, since the predicted Gaussians from AnySplat are aligned with the predicted camera poses. On the other hand, UniQueR provides both more accurate poses and a better Gaussian initialization.

Efficiency and geometry. Tab. 6 compares end-to-end efficiency and rendered-depth accuracy. Relative to AnySplat, UniQueR uses 14.7× fewer Gaussians, 39% less GPU memory, and 2.35× less inference time, while reducing depth Abs Rel from 0.062 to 0.038.

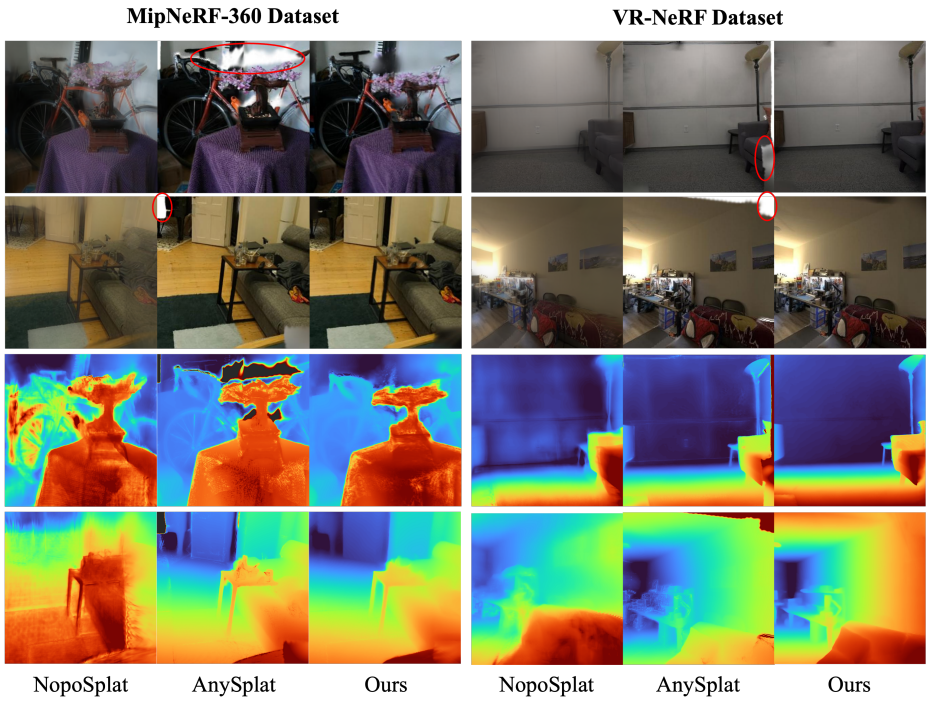


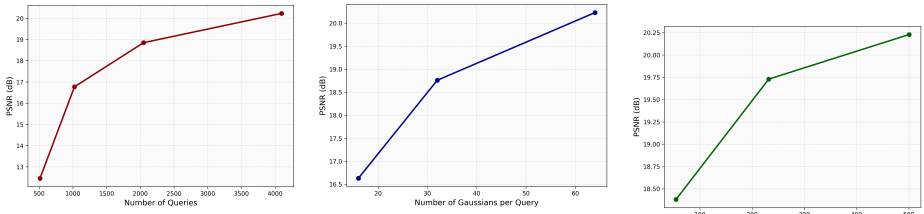
Fig. 4: Qualitative and geometric comparison. Top: RGB rendered on held-out novel views. Bottom: depth rendered from each method's predicted Gaussians, not from our backbone point-map head. In the illustrated regions with limited input-view coverage, AnySplat contains blank RGB regions and depth holes, while UniQueR produces fewer missing regions.

4.3 Qualitative Analysis

Figure 4 presents qualitative comparisons of rendered novel views and depth maps. NoPoSplat struggles to produce high-fidelity outputs, often exhibiting severe artifacts and incomplete geometry. AnySplat, while more stable, frequently generates noticeable blank regions due to its reliance on per-pixel predictions, which cannot allocate Gaussians to unobserved areas. In contrast, UniQueR delivers consistently higher-quality renderings with more complete structures and sharper details. The depth maps further confirm this: AnySplat exhibits holes and noisy depth in occluded regions, while UniQueR produces cleaner and more complete geometry with sharper boundaries, consistent with the quantitative depth improvement in Tab. 6.

4.4 Ablation Studies

We perform a series of ablations to assess the contribution of each component in UniQueR as follows:



(a) Ablation on number of Queries **(b)** Ablation on number of Gaussians **(c)** Trainable parameters (M)

Fig. 5: Stage-1 (224^2) PSNR ablations on (a) query count, (b) Gaussians per query, and (c) trainable model capacity in millions. Increasing any of these factors leads to consistent improvements in PSNR, demonstrating clear scaling behavior in UniQueR.

Table 7: Stage-1 (224^2) ablations on Mip-NeRF 360 with three input views.

Configuration	PSNR ↑	SSIM ↑	LPIPS ↓
(a) w/o depth rendering	19.96	0.718	0.234
(b) Point-map queries only	18.58	0.627	0.313
(c) Random initialization only	12.11	0.259	0.574
Full model (hybrid initialization)	20.23	0.713	0.182

Number of queries. We begin by examining how the reconstruction quality scales with the number of queries. As shown in Fig. 5a, increasing the query count consistently boosts PSNR, indicating that a larger query set provides greater spatial coverage and representational capacity.

Gaussians per query. Next, we vary the number of Gaussians (16, 32 and 64) spawned by each query with a total number of 4096 queries. Figure 5b shows that allocating more Gaussians per query results in higher PSNR, confirming that a richer local density representation enables finer detail synthesis.

Model size. We further ablate the architecture size by adjusting the depth and hidden size of the transformer. As illustrated in Fig. 5c, models with larger capacity achieve noticeably better rendering quality, reflecting a clear scaling trend with respect to the trainable parameter count.

Depth rendering. To understand the role of depth supervision, we disable Gaussian-rendered depth during training. As shown in Tab. 7(a), omitting depth rendering leads to a marked degradation in performance, highlighting the importance of depth cues for stabilizing geometry prediction.

Query initialization. Finally, we compare the two initialization sources in Tab. 7(b-c). Point-map-only initialization remains geometrically grounded but is weaker than the full hybrid model, while random-only initialization collapses. This confirms that grounding queries using coarse point-map estimates is crucial for stable optimization and high-quality rendering.

5 Conclusion

We presented UniQueR, a unified query-based feedforward framework for 3D reconstruction from unposed images. Unlike existing feedforward methods that rely on depth maps or pixel-aligned Gaussians, UniQueR introduces learnable 3D queries that decouple the scene representation from input viewpoints. Each query spawns a set of Gaussians supervised through differentiable rendering of RGB and depth, requiring no ground-truth 3D data. Experiments on Mip-NeRF 360 and VR-NeRF demonstrate improvements in both rendering quality and geometric accuracy over prior feedforward approaches, while using significantly fewer primitives and less memory.

Limitations. the current formulation assumes static scenes and does not model temporal dynamics. The fixed query budget can underrepresent highly complex scenes and trades some high-frequency perceptual detail for compactness. The model also depends on a large frozen pretrained geometry backbone. Adaptive query allocation, lighter backbones, controlled component ablations, and temporally persistent queries are promising directions.

References

1. Agarwal, S., Snavely, N., Seitz, S.M., Szeliski, R.: Bundle adjustment in the large. In: ECCV. pp. 29–42 (2010)
2. Agarwal, S., Snavely, N., Simon, I., Seitz, S.M., Szeliski, R.: Building rome in a day. In: ICCV. pp. 72–79 (2009)
3. Asim, M., Wewer, C., Lenssen, J.E.: Scenetok: A compressed, diffusable token space for 3d scenes. arXiv preprint arXiv:2602.18882 (2026), https://arxiv.org/abs/2602.18882
4. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields (2022), https://arxiv.org/abs/2111.12077
5. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: CVPR. pp. 5470–5479 (2022)
6. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Zip-nerf: Anti-aliased grid-based neural radiance fields. In: ICCV. pp. 19697–19705 (2023)
7. Bloesch, M., Czarnowski, J., Clark, R., Leutenegger, S., Davison, A.J.: Codeslam—learning a compact, optimisable representation for dense visual slam. In: CVPR. pp. 2560–2568 (2018)
8. Cabon, Y., Stoffl, L., Antsfeld, L., Csurka, G., Chidlovskii, B., Revaud, J., Leroy, V.: Must3r: Multi-view network for stereo 3d reconstruction. arXiv preprint arXiv:2503.01661 (2025)
9. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers (2020), https://arxiv.org/abs/2005.12872
10. Charatan, D., Li, S., Tagliasacchi, A., Sitzmann, V.: pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction (2024), https://arxiv.org/abs/2312.12337
11. 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. arXiv preprint arXiv:2403.14627 (2024)

12. Chen, Z., Yang, J., Yang, H.: Pref3r: Pose-free feed-forward 3d gaussian splatting from variable-length image sequence (2024), https://arxiv.org/abs/2411.16877
13. Crandall, D., Owens, A., Snavely, N., Huttenlocher, D.: Discrete-continuous optimization for large-scale structure from motion. In: CVPR. pp. 3001–3008 (2011)
14. Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: Monoslam: Real-time single camera slam. TPAMI **29**(6), 1052–1067 (2007)
15. Dexheimer, E., Davison, A.J.: Learning a depth covariance function. In: CVPR. pp. 13122–13131 (2023)
16. Doersch, C., Yang, Y., Vecerik, M., Gokay, D., Gupta, A., Aytar, Y., Carreira, J., Zisserman, A.: Tapir: Tracking any point with per-frame initialization and temporal refinement. In: ICCV. pp. 10061–10072 (2023)
17. Duzceker, A., Galliani, S., Vogel, C., Speciale, P., Dusmanu, M., Pollefeys, M.: Deepvideomvs: Multi-view stereo on video with recurrent spatio-temporal fusion. In: CVPR. pp. 15324–15333 (2021)
18. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multiview stereopsis. TPAMI **32**(8), 1362–1376 (2009)
19. Galliani, S., Lasinger, K., Schindler, K.: Massively parallel multiview stereopsis by surface normal diffusion. In: ICCV. pp. 873–881 (2015)
20. Gao, H., Chen, Z., Chen, Z., Chen, L., Liu, J., Zhang, S., Zhao, F.: Point-detr3d: Leveraging imagery data with spatial point prior for weakly semi-supervised 3d object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence (2024), https://arxiv.org/abs/2403.15317
21. Herau, Q., Bennehar, M., Moreau, A., Piasco, N., Roldão, L., Tsishkou, D., Migniot, C., Vasseur, P., Demonceaux, C.: 3dgs-calib: 3d gaussian splatting for multimodal spatiotemporal calibration. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 8315–8321 (2024)
22. Herau, Q., Piasco, N., Bennehar, M., Roldão, L., Tsishkou, D., Liu, B., Migniot, C., Vasseur, P., Demonceaux, C.: Pose optimization for autonomous driving datasets using neural rendering models. arXiv preprint arXiv:2504.15776 (2025)
23. Herau, Q., Piasco, N., Bennehar, M., Roldao, L., Tsishkou, D., Migniot, C., Vasseur, P., Demonceaux, C.: Moisst: Multimodal optimization of implicit scene for spatiotemporal calibration. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1810–1817 (2023)
24. Herau, Q., Piasco, N., Bennehar, M., Roldao, L., Tsishkou, D., Migniot, C., Vasseur, P., Demonceaux, C.: Soac: Spatio-temporal overlap-aware multi-sensor calibration using neural radiance fields. In: CVPR. pp. 15131–15140 (2024)
25. 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. arXiv preprint arXiv:2311.04400 (2024)
26. Huang, B., Yu, Z., Chen, A., Geiger, A., Gao, S.: 2d gaussian splatting for geometrically accurate radiance fields. In: ACM SIGGRAPH. pp. 1–11 (2024)
27. Jang, W., Agapito, L.: Codenerf: Disentangled neural radiance fields for object categories. In: ICCV. pp. 12949–12958 (2021)
28. Jang, W., Agapito, L.: Nvist: In the wild new view synthesis from a single image with transformers. In: CVPR. pp. 10181–10193 (2024)
29. 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. arXiv preprint arXiv:2505.23716 (2025)
30. Jin, H., Jiang, H., Tan, H., Zhang, K., Bi, S., Zhang, T., Luan, F., Snavely, N., Xu, Z.: LVSM: A large view synthesis model with minimal 3d inductive bias. arXiv preprint arXiv:2410.17242 (2024), https://arxiv.org/abs/2410.17242

31. Karaev, N., Rocco, I., Graham, B., Neverova, N., Vedaldi, A., Rupprecht, C.: Cotracker: It is better to track together. arXiv preprint arXiv:2307.07635 (2023)
32. Ke, B., Obukhov, A., Huang, S., Metzger, N., Daudt, R.C., Schindler, K.: Repurposing diffusion-based image generators for monocular depth estimation. In: CVPR. pp. 9492–9502 (2024)
33. Keetha, N., Müller, N., Schönberger, J., Porzi, L., Zhang, Y., Fischer, T., Knapitsch, A., Zauss, D., Weber, E., Antunes, N., Luiten, J., Lopez-Antequera, M., Bulò, S.R., Richardt, C., Ramanan, D., Scherer, S., Kontschieder, P.: MapAnything: Universal feed-forward metric 3D reconstruction (2025), arXiv preprint arXiv:2509.13414
34. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. TOG **42**(4), 139–1 (2023)
35. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering (2023), https://arxiv.org/abs/2308.04079
36. Klein, G., Murray, D.: Parallel tracking and mapping for small ar workspaces. In: ISMAR. pp. 1–10 (2007)
37. Leroy, V., Cabon, Y., Revaud, J.: Grounding image matching in 3d with mast3r (2024), https://arxiv.org/abs/2406.09756
38. 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)
39. Liu, Y., Wang, T., Zhang, X., Sun, J.: Petr: Position embedding transformation for multi-view 3d object detection (2022), https://arxiv.org/abs/2203.05625
40. 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 (2024)
41. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV. pp. 405–421 (2020)
42. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Trans. Graph. **41**(4), 102:1–102:15 (Jul 2022). https://doi.org/10.1145/3528223.3530127, https://doi.org/10.1145/3528223.3530127
43. Newcombe, R.A., Lovegrove, S.J., Davison, A.J.: Dtam: Dense tracking and mapping in real-time. In: ICCV. pp. 2320–2327 (2011)
44. Pan, L., Baráth, D., Pollefeys, M., Schönberger, J.L.: Global structure-from-motion revisited (2024), https://arxiv.org/abs/2407.20219
45. Reizenstein, J., Shapovalov, R., Henzler, P., Sbordone, L., Labatut, P., Novotny, D.: Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In: International Conference on Computer Vision (2021)
46. Ren, J., Tyszkiewicz, M.J., Huang, J., Gojcic, Z.: Tokengs: Decoupling 3d gaussian prediction from pixels with learnable tokens. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15365–15375 (2026)
47. Sayed, M., Gibson, J., Watson, J., Prisacariu, V., Firman, M., Godard, C.: Simplerecon: 3d reconstruction without 3d convolutions. In: ECCV. pp. 1–19 (2022)
48. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: CVPR. pp. 4104–4113 (2016)
49. Schönberger, J.L., Zheng, E., Frahm, J.M., Pollefeys, M.: Pixelwise view selection for unstructured multi-view stereo. In: ECCV. pp. 501–518 (2016)

50. Smart, B., Zheng, C., Laina, I., Prisacariu, V.A.: Splatt3r: Zero-shot gaussian splatting from uncalibrated image pairs (2024), https://arxiv.org/abs/2408.13912
51. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3d. TOG **25**(3), 835–846 (2006)
52. Sweeney, C., Sattler, T., Hollerer, T., Turk, M., Pollefeys, M.: Optimizing the viewing graph for structure-from-motion. In: ICCV. pp. 801–809 (2015)
53. Szymanowicz, S., Rupprecht, C., Vedaldi, A.: Splatter image: Ultra-fast single-view 3d reconstruction (2024), https://arxiv.org/abs/2312.13150
54. Tang, J., Chen, Z., Chen, X., Wang, T., Zeng, G., Liu, Z.: LGM: Large multi-view gaussian model for high-resolution 3d content creation. arXiv preprint arXiv:2402.05054 (2024), https://arxiv.org/abs/2402.05054
55. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: ECCV. pp. 402–419 (2020)
56. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment—a modern synthesis. In: ICCVW. pp. 298–372 (2000)
57. Wang, H., Agapito, L.: 3d reconstruction with spatial memory (2024), https://arxiv.org/abs/2408.16061
58. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2025)
59. Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W.: Neus: learning neural implicit surfaces by volume rendering for multi-view reconstruction. In: NIPS. pp. 27171–27183 (2021)
60. Wang, Q., Zhang, Y., Holynski, A., Efros, A.A., Kanazawa, A.: Continuous 3d perception model with persistent state (2025), https://arxiv.org/abs/2501.12387
61. Wang, S., Liu, Y., Wang, T., Li, Y., Zhang, X.: Exploring object-centric temporal modeling for efficient multi-view 3d object detection (2023), https://arxiv.org/abs/2303.11926
62. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: Dust3r: Geometric 3d vision made easy. In: CVPR. pp. 20697–20709 (2024)
63. 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 (2025), https://arxiv.org/abs/2507.13347
64. Wang, Y., Guizilini, V., Zhang, T., Wang, Y., Zhao, H., Solomon, J.: DETR3D: 3d object detection from multi-view images via 3d-to-2d queries. In: Conference on Robot Learning (2022), https://arxiv.org/abs/2110.06922
65. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing **13**(4), 600–612 (2004). https://doi.org/10.1109/TIP.2003.819861
66. Wilson, K., Snavely, N.: Robust global translations with 1dsfm. In: ECCV. pp. 61–75 (2014)
67. Wu, C.: Towards linear-time incremental structure from motion. In: 3DV. pp. 127–134 (2013)
68. Wu, C., Agarwal, S., Curless, B., Seitz, S.M.: Multicore bundle adjustment. In: CVPR. pp. 3057–3064 (2011)
69. Xiao, Y., Wang, Q., Zhang, S., Xue, N., Peng, S., Shen, Y., Zhou, X.: Spatialtracker: Tracking any 2d pixels in 3d space. In: CVPR. pp. 20406–20417 (2024)

70. Xu, L., Agrawal, V., Laney, W., Garcia, T., Bansal, A., Kim, C., Rota Bulò, S., Porzi, L., Kontschieder, P., Božič, A., Lin, D., Zollhöfer, M., Richardt, C.: Vr-nerf: High-fidelity virtualized walkable spaces. In: SIGGRAPH Asia 2023 Conference Papers. pp. 1–12. SA '23, ACM (Dec 2023). https://doi.org/10.1145/3610548.3618139, http://dx.doi.org/10.1145/3610548.3618139
71. 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 (2025), https://arxiv.org/abs/2501.13928
72. Yang, J., Desai, K., Packer, C., Bhatia, H., Rhinehart, N., McAllister, R., Gonzalez, J.E.: Carff: Conditional auto-encoded radiance field for 3d scene forecasting. In: European Conference on Computer Vision. pp. 225–242. Springer (2024)
73. Yang, J., Desai, K., Packer, C., Bhatia, H., Rhinehart, N., McAllister, R., Gonzalez, J.E.: Carff: Conditional auto-encoded radiance field for 3d scene forecasting. In: Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., Varol, G. (eds.) Computer Vision – ECCV 2024. pp. 225–242. Springer Nature Switzerland, Cham (2025)
74. Yao, Y., Luo, Z., Li, S., Fang, T., Quan, L.: Mvsnet: Depth inference for unstructured multi-view stereo. In: ECCV. pp. 767–783 (2018)
75. Yao, Y., Luo, Z., Li, S., Zhang, J., Ren, Y., Zhou, L., Fang, T., Quan, L.: Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In: CVPR. pp. 1790–1799 (2020)
76. Yariv, L., Gu, J., Kasten, Y., Lipman, Y.: Volume rendering of neural implicit surfaces. In: NIPS. pp. 4805–4815 (2021)
77. Ye, B., Liu, S., Xu, H., Li, X., Pollefeys, M., Yang, M.H., Peng, S.: No pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images (2024), https://arxiv.org/abs/2410.24207
78. Yeshwanth, C., Liu, Y.C., Nießner, M., Dai, A.: Scannet++: A high-fidelity dataset of 3d indoor scenes. In: ICCV (2023)
79. Yin, W., Zhang, C., Chen, H., Cai, Z., Yu, G., Wang, K., Chen, X., Shen, C.: Metric3d: Towards zero-shot metric 3d prediction from a single image. In: ICCV. pp. 9043–9053 (2023)
80. Yu, X., Yang, H.: Sim-sync: From certifiably optimal synchronization over the 3d similarity group to scene reconstruction with learned depth. IEEE Robotics and Automation Letters (2024)
81. Yu, Z., Chen, A., Huang, B., Sattler, T., Geiger, A.: Mip-splatting: Alias-free 3d gaussian splatting. Conference on Computer Vision and Pattern Recognition (CVPR) (2024)
82. Zhang, B., Yan, J., Wonka, P.: 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. In: ACM SIGGRAPH 2023 Conference Proceedings (2023)
83. Zhang, K., Bi, S., Tan, H., Xiangli, Y., Zhao, N., Sunkavalli, K., Xu, Z.: GS-LRM: Large reconstruction model for 3d gaussian splatting. In: European Conference on Computer Vision (2024), https://arxiv.org/abs/2404.19702
84. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)
85. Zhang, S., Wang, J., Xu, Y., Xue, N., Rupprecht, C., Zhou, X., Shen, Y., Wetzstein, G.: Flare: Feed-forward geometry, appearance and camera estimation from uncalibrated sparse views. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 21936–21947 (2025)

86. Zhang, X., Zheng, X., Yin, Y., Zhao, T., Tang, K., Mi, M.B., Xu, Z., Chen, D.Z.: Anchorsplat: Feed-forward 3d gaussian splatting with 3d geometric priors. arXiv preprint arXiv:2604.07053 (2026), https://arxiv.org/abs/2604.07053
87. Zhou, T., Tucker, R., Flynn, J., Fyffe, G., Snavely, N.: Stereo magnification: Learning view synthesis using multiplane images. arXiv preprint arXiv:1805.09817 (2018)