CVPR
#583

Learning Compact 3D Representations from Feed-Forward Novel View Synthesis

Anonymous CVPR submission

Paper ID 583

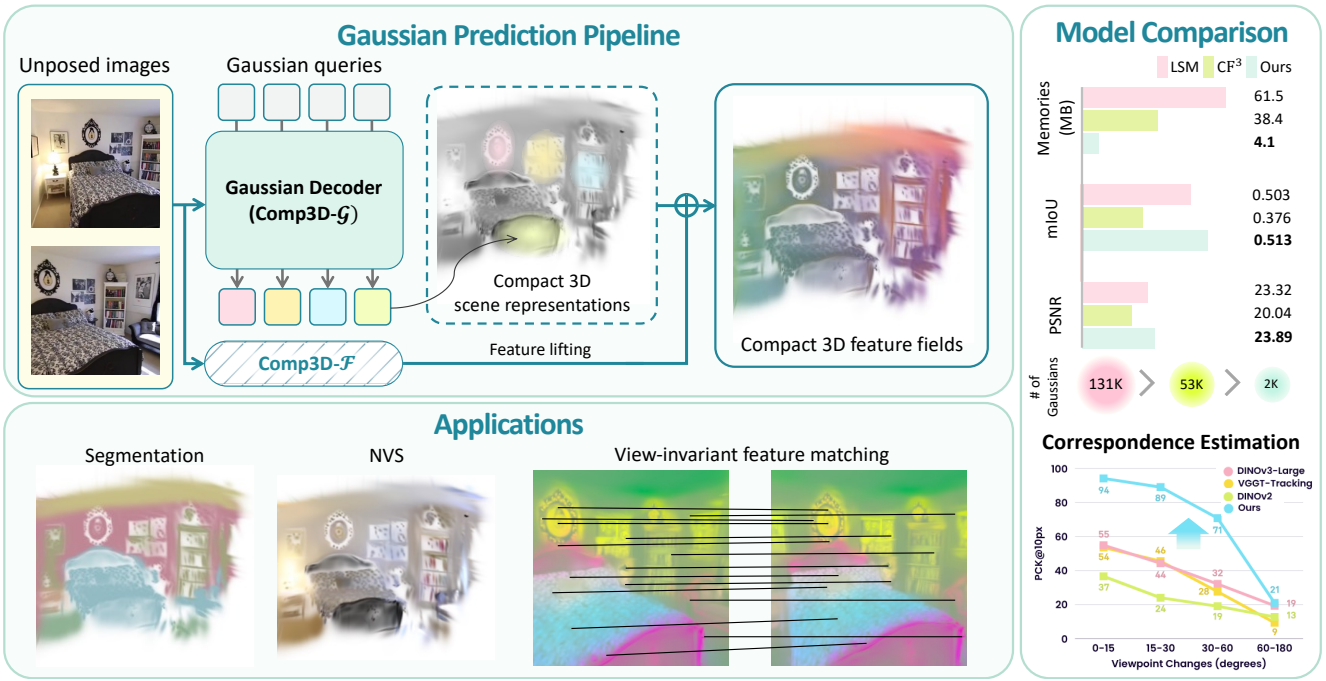


Figure 1. **Teaser.** Our method learns compact 3D Gaussians from unposed multi-view images through a query-based Gaussian decoding pipeline. Compact representations enable efficient 2D-to-3D feature lifting (called compact 3D feature fields) for downstream applications, including open-vocabulary segmentation and view-invariant feature matching. Compared to prior works (LSM [9] and CF^3 [19]), ours results in fewest Gaussians (about **50**$\times$ fewer Gaussians than [9, 19]) with superior memory efficiency and novel view synthesis quality.

Abstract

Reconstructing and understanding 3D scenes from sparse views in a feed-forward manner remains challenging. While recent approaches use per-pixel 3D Gaussian Splatting for reconstruction and 2D-to-3D feature lifting for scene understanding, they generate excessive redundant Gaussians, causing high memory overhead and sub-optimal multi-view feature aggregation. We propose a feed-forward framework that estimates compact Gaussians only at essential spatial locations, minimizing redundancy while enabling effective feature lifting. We introduce learnable tokens that aggregate multi-view features through self-attention to guide Gaussian generation, ensuring each Gaussian integrates relevant visual features across views. We then exploit the learned attention patterns to efficiently lift features. Extensive experiments on 3D open-vocabulary segmentation and view-invariant feature generation demonstrate our approach's effectiveness. Results show that a compact yet geometrically meaningful representation is sufficient for high-quality scene reconstruction, achieving superior memory efficiency and feature fidelity compared to existing methods. All of our code will be made publicly available.

1. Introduction

Obtaining 3D scene representations from sparse multi-view images in a feed-forward manner remains a fundamental

CVPR
#583

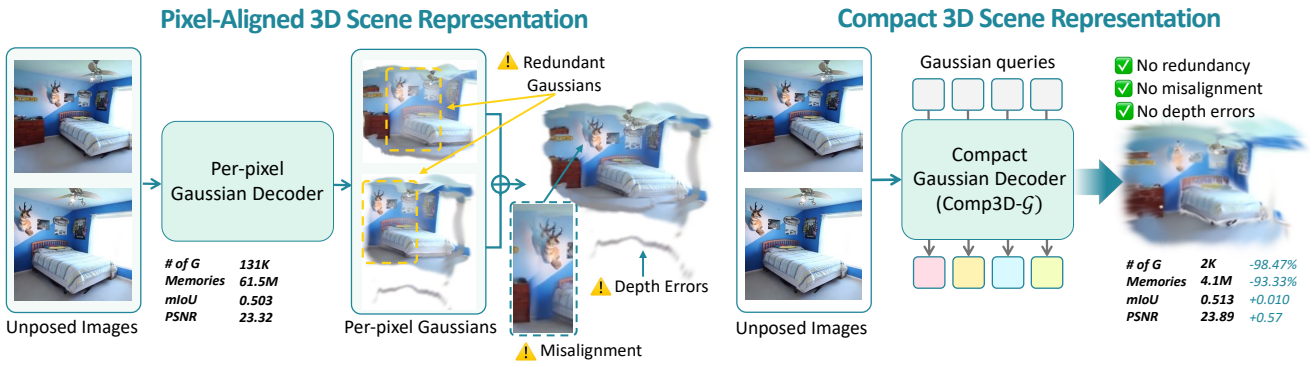


Figure 2. **Comparison of per-pixel and compact scene representations. (Left)**: Existing per-pixel estimators predict one or multiple Gaussians per pixel, resulting in redundant Gaussians with misalignments across views. **(Right)**: Our method uses learnable Gaussian queries to discover and decode only compact 3D Gaussians at essential locations, achieving a compact representation with only 2K Gaussians and 0.1M memory while avoiding redundancy and achieving superior segmentation and novel view synthesis performance.

challenge in computer vision and graphics, with broad implications for robotics [33], scene understanding [10], and novel view synthesis [48]. Recently, feed-forward 3D Gaussian splatting frameworks have gained considerable attention, demonstrating impressive performance in reconstruction and understanding [9, 11, 14, 34, 48].

However, these approaches rely on *dense, per-pixel* Gaussian predictions, which often lead to degraded performance due to misaligned primitives across views (Fig. 2) and incur substantial computational overhead when incorporating semantic features [9, 34]. Consequently, prior works [9, 34] compress rich semantic information into lower-dimensional embeddings at the cost of information loss, yielding sub-optimal scene understanding. This raises a fundamental question: ***do we need such pixel-aligned Gaussians to reconstruct and understand 3D scenes?***

As humans, we do not maintain pixel-perfect mental reconstructions of every surface to understand our surroundings. Instead, we form compact, semantically meaningful abstractions of identifying key objects, their rough spatial relationships, and overall scene structure [1, 35]. Drawing direct inspiration from human visual cognition, we propose a novel framework for learning ***compact 3D scene representations*** from unposed image observations in a feed-forward fashion.

Similar to prior approaches [14, 48], we first extract image features from visual encoders with rich geometric priors (e.g., VGGT [41]). However, instead of learning to estimate per-pixel Gaussians directly from the extracted feature maps, we introduce a compact set of learnable query tokens that discover and decode essential 3D Gaussians. Specifically, we adopt a transformer architecture [40] where learnable query tokens and the image features are processed through multiple self-attention blocks. We decode the refined learnable query tokens as 3D Gaussians, where the query tokens learn to aggregate essential information across multiple views to faithfully represent the scene.

Crucially, our framework requires no explicit supervision from ground-truth depths or scene decompositions. Despite training solely on photometric reconstruction, each token naturally learns to represent different regions, with each token attending to coherent spatial regions across views. This emergent behavior arises from the inherent structure of the task: to efficiently reconstruct novel views with a limited number of Gaussians, the model must learn to allocate Gaussians to meaningful regions. We show that after training, our model can estimate a compact set of 3D Gaussians, which enables efficient novel view synthesis and 2D-to-3D feature lifting without compression, significantly improving 3D scene understanding tasks where the rich representation of the semantic features is critical.

Our framework also provides a novel solution to a key challenge in 2D-to-3D feature lifting: handling multi-view feature inconsistencies. While other methods [4, 19, 24, 50] require additional aggregation methods to handle inconsistent features across viewpoints, we observe that our model's emergent property of tokens attending to spatially coherent regions across views can be directly leveraged for feature aggregation. Specifically, we propose a view-invariant feature decoder that reuses the attention maps from our learned Gaussian decoder while training only the value projections. This feature decoder can then take features from any desired visual encoder as input and decode multi-view aggregated features. By attaching these aggregated features to our estimated 3D Gaussians, we enable efficient novel view rendering with view-invariant features.

We validate the effectiveness of our approach through extensive experiments on real-world datasets. For novel view synthesis, despite using **50**× fewer Gaussians than per-pixel methods, we achieve competitive visual quality while enabling substantially faster rendering. More importantly, for 3D semantic understanding tasks, our compact

CVPR #583

Gaussians combined with the semantic features aggregated with the view-invariant feature decoder significantly outperform previous feed-forward approaches that attempt to lift features through dense per-pixel Gaussians. We further demonstrate that the compact set of 3D Gaussians can achieve high-fidelity novel view synthesis when combined with efficient test-time optimization steps.

2. Related work

Learning compact 3D scene representations. Decomposing scenes into geometric primitives (e.g., meshes, polygons, superquadric primitives) has been extensively studied [29, 30, 39], but these methods typically require 3D data (e.g., point clouds) as input. SuperDec [10] recently proposed feed-forward decomposition into superquadric primitives, but relies on pre-computed 3D point clouds and an external 3D instance segmentation model to identify instances. The differentiable blocks world (DBW) [26] learns superquadric parameters directly from multi-view images through photometric optimization, but can only model up to 10 primitives and object-centric scenes. In contrast, we aim to learn an efficient solution that can estimate compact scene representations in a feed-forward manner, only given multi-view images captured in real-world scenarios.

Feed-forward 3D Gaussian splatting. Recently, 3D Gaussian Splatting (3DGS) [18] has gained significant attention for 3D reconstruction and novel view synthesis. However, it requires dense multi-view captures with known poses and time-consuming per-scene optimization. To address these limitations, recent approaches [2, 3, 7, 16, 42, 47, 49, 51] have explored generalizable feed-forward networks [11, 12, 14, 15, 37, 48] that synthesize novel views from sparse inputs by learning priors from large-scale datasets or leveraging foundation models such as pointmap regression models [20, 41, 43]. Despite these advances, existing feed-forward models [14, 48] typically predict one or multiple Gaussians per-pixel, resulting in millions of Gaussians for multiple views or high-resolution images. This strategy produces excessive redundant Gaussians, degrading performance and causing artifacts as input views increase [45, 46]. When incorporating semantic features through 2D-to-3D lifting, the excessive Gaussians create significant computational overhead [1, 19]. Previous works [1, 9, 31, 53] address this by compressing semantic features into lower dimensions using specialized autoencoders, but this causes information loss and sub-optimal scene understanding [22].

Towards compact 3D Gaussian splatting. To address the redundancy problem in per-pixel Gaussian estimation, recent works [13, 15, 44, 46, 52, 55] have attempted to mitigate this issue by reducing the number of Gaussians post-hoc. FreeSplat [46] and LongSplat [13] iteratively add pixel-wise Gaussians only where existing projections are insufficient. ZPressor [44] and Long-LRM [55] employ token merging to reduce redundant Gaussians with similar features before decoding them to per-pixel Gaussians. Anysplat [15] voxelizes the Gaussians in 3D space. However, these approaches do not fundamentally address the input-view bias inherent in per-pixel processing. EvolSplat [25] and VolSplat [45] employ global voxel representations but remain domain-constrained or limited by fixed resolutions. We instead propose using learnable tokens to generate compact global Gaussians guided by input features, producing only essential Gaussians for scene representation.

3. Methodology

We introduce Comp3D-$\mathcal{G}$, a compact 3D Gaussian splatting decoder built upon a transformer [40] that takes multi-view images of a scene as input and produces a compact set of 3D Gaussians that best represent the scene. We start by introducing the problem (§ 3.1), followed by our architectural details (§ 3.2), and the training setup (§ 3.3). We further analyze the emergent properties in our learned Comp3D-$\mathcal{G}$ (§ 3.4), and show how this property can be leveraged to effectively lift any 2D features into 3D in a view-invariant manner (§ 3.5).

3.1. Problem definition and notion

Given a set of V multi-view images $\{I_v\}_{v=1}^{V}$ capturing the same scene where $I_v \in \mathbb{R}^{H\times W\times 3}$, the model outputs a set of N 3D Gaussians $\{G_i\}_{i=1}^{N}$ with $G_i = \{\mu_i, \sigma_i, \Sigma_i, c_i\}_{i=1}^{N}$, where $\mu_i \in \mathbb{R}^3$ denotes 3D Gaussian center, $\sigma_i \in [0, 1)$ represents opacity, $\Sigma_i \in \mathbb{R}^{3\times 3}$ denotes the covariance matrix, and $c_i \in \mathbb{R}^{3(L+1)}$ indicates spherical harmonics coefficients with L levels that encode color attributes. Although other formats of 3D scene representations (e.g., point clouds [41, 43], polygons [27]) could be considered, we adopt 3D Gaussians [18] as our default representation due to their efficient rendering speeds and simplicity to further incorporate features as an additional attribute that enables multiple downstream tasks.

3.2. Architecture

Drawing inspiration from how humans naturally form abstract scene representations through selective attention, we design a remarkably simple architecture that learns to decode compact 3D Gaussians from multi-view observations.

Multi-view feature encoding. To decode the 3D Gaussians, given the V input images $\{I_v\}_{v=1}^{V}$, we first extract visual features using a pre-trained visual encoder $\mathcal{E}(\cdot)$, yielding feature maps $\mathbf{F}_v \in \mathbb{R}^{h\times w\times d}$ for each view, where h and w are the height and width of the feature map and d is the feature dimension. To effectively learn to estimate 3D Gaussians, we follow prior works and adopt VGGT [41] as our default visual encoder, which has learned rich geometric priors from large-scale geometry learning.

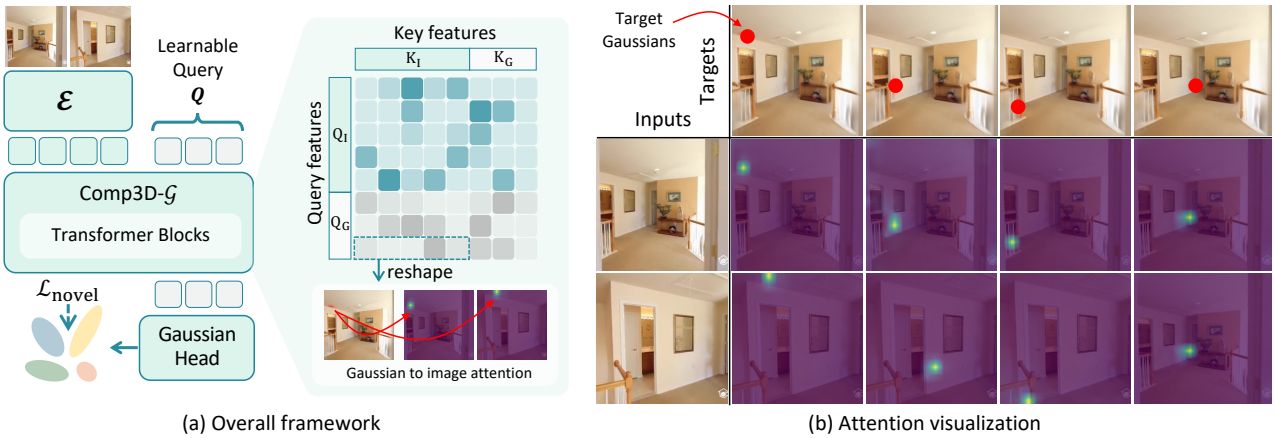


Figure 3. **Architecture and emergent attention behaviors of our Gaussian decoder (Comp3D-$\mathcal{G}$).** (a) Our framework extracts multi-view features using VGGT, then processes them with learnable query tokens through transformer blocks in our Gaussian decoder (Comp3D-$\mathcal{G}$). The refined queries are decoded into compact 3D Gaussians via a Gaussian Head, trained with novel view synthesis loss $\mathcal{L}_{novel}$. (b) Visualization of learned attention patterns between a target Gaussian (red box) and input images. Without explicit supervision, each query token learns to attend to spatially coherent regions across multiple views, naturally discovering corresponding regions.

Query-based scene decoding. At the core of our architecture is the compact set of N learnable query tokens $\mathbf{Q} \in \mathbb{R}^{N\times d}$, where each token is tasked with discovering and representing a specific region of the 3D scene. These tokens serve as abstraction units that learn to aggregate relevant information from the extracted multi-view features F_v to form coherent 3D Gaussians. Unlike per-pixel Gaussian estimation methods [2, 48] that rigidly map each pixel to Gaussians, our query tokens can flexibly attend to any region across all input views, learning to allocate representational capacity where it is most needed.

Cross-view attention aggregation. The key to our approach lies in how query tokens interact with multi-view features. We concatenate the learnable query tokens with the image features to form a unified sequence: $\mathbf{X} = [\mathbf{Q}; \mathbf{F}] \in \mathbb{R}^{N+(V\times h\times d)\times d}$. This combined representation is processed through K transformer layers with full self-attention, enabling bidirectional information flow. Through these attention layers, each query token learns to: (1) aggregate relevant visual information from specific regions across all input views, (2) exchange information with other query tokens to avoid redundancy and ensure comprehensive coverage, and (3) progressively refine its understanding of which 3D region it should represent.

Gaussian parameter decoding. After the transformer blocks, we extract the refined query tokens $\bar{\mathbf{Q}} \in \mathbb{R}^{N\times d}$ and decode each token $\bar{\mathbf{Q}}_i$ into a single Gaussian G_i, estimating Gaussian attributes through lightweight MLP heads.

3.3. Training

Unlike previous methods that learn scene decompositions from ground-truth labels [10, 29], adopting 3D Gaussians as our representation enables our framework to be learned solely through the objective of novel view synthesis.

Learning compact scene representations from novel view synthesis. Given the predicted 3D Gaussians $\{G_i\}_{i=1}^{N}$ from our query tokens, we train the model by rendering these Gaussians at novel target viewpoints and minimizing the photometric difference with ground-truth images. Following feed-forward novel view synthesis frameworks [48], we project the 3D Gaussians to a target view I_t where $I_t \notin \{I_v\}_{v=1}^{V}$ with known camera pose π_t during training. Each pixel p of the target view image is rendered via alpha blending of Gaussian color attributes according to their depth order [18]:

$$\hat{I}_t(p) = \sum_{i=1}^{N} \tilde{c}_i \sigma_i G'_i(p) \prod_{j=1}^{i-1} (1 - \sigma_j G'_j(p)), \qquad (1)$$

where $\tilde{c}_i$ is the view-dependent color attribute of each Gaussian obtained by decoding spherical harmonics coefficients, and G'_i is the 3D Gaussian projected onto 2D screen space. Our training objective combines the mean squared error between rendered and ground-truth images $\mathcal{L}_{\text{MSE}}$ and the perceptual loss $\mathcal{L}_{\text{LPIPS}}$ as

$$\mathcal{L}_{\text{novel}} = \lambda_{\text{MSE}} \mathcal{L}_{\text{MSE}}(\hat{I}_t, I_t) + \lambda_{\text{LPIPS}} \mathcal{L}_{\text{LPIPS}}(\hat{I}_t, I_t), \qquad (2)$$

following prior works [48], where λ_{MSE} and λ_{LPIPS} are hyperparameters.

Low-pass filtering for robust training. One of the key challenges in learning feed-forward 3D Gaussian splatting is correctly locating Gaussian positions (μ_i). Without accurate positions, prior works show that the Gaussians often fail to be positioned inside the view frustum of the target image viewpoint, leading to sparse gradients and mode collapse [2, 11]. Aligned with these analyses, we also observe that naively training the network with photometric

loss leads to unstable training. To address this, we adopt the progressive low-pass filter from RAIN-GS [17]. For rendering Gaussian G_i, the projected 2D Gaussian G'_i is defined as:

$$G'_i(p) = e^{-\frac{1}{2}(p-\mu'_i)^T(\Sigma'_i+sI)^{-1}(p-\mu'_i)}, \quad (3)$$

where p is the 2D pixel location, I is the identity matrix, and s controls the Gaussian size. While 3DGS [18] uses $s = 0.3$ to ensure 1-pixel coverage, RAIN-GS shows that progressively annealing from $s = 300$ to $s = 0.3$ stabilizes per-scene optimization by allowing Gaussians to learn from enlarged regions initially. We adopt this strategy in our feed-forward training, gradually annealing s during training. This ensures robust gradients during early training when position predictions are suboptimal, while enabling fine-grained detail to be modeled as the network learns accurate positions. Our ablations (§ 4.5) further verify that this strategy is crucial for stable training.

3.4. Analysis

Although we do not provide any supervision for how the N query tokens should partition the scene, we observe that the model eventually learns to effectively estimate a set of N Gaussians that best reconstructs the scene purely from the photometric reconstruction objective.

Emergent properties within learned attentions. To understand how each query token learns to aggregate information from multi-view features, we examine the attention weights between learnable tokens $\mathbf{Q}$ and the multi-view image features F_v inside the *self-attention blocks* of Comp3D-$\mathcal{G}$. As visualized in Fig. 3-(a), we examine the attention weights where the attention query is from the N learnable tokens Q_G and the attention key is from the multi-view image features K_I.

As illustrated in Fig. 3-(b), when we select a specific Gaussian and visualize its corresponding query token's attention map across input views, we observe sharp, focused attention patterns on spatially coherent regions across multiple views, effectively discovering multi-view correspondences without any explicit supervision. For instance, the target Gaussian highlighted in red attends strongly to the corresponding object regions across different viewpoints. We believe that this emergent behavior arises from an implicit optimization pressure: to accurately reconstruct novel views with a limited number of N Gaussians, the model has learned to position 3D Gaussians to geometrically coherent regions.

3.5. Any-Feature 3D Lifting

Building upon the emergent property within the self-attention maps of Comp3D-$\mathcal{G}$ (§ 3.4), we present a simple yet effective approach for lifting arbitrary 2D features into view-invariant 3D representations, dubbed Comp3D-$\mathcal{F}$.

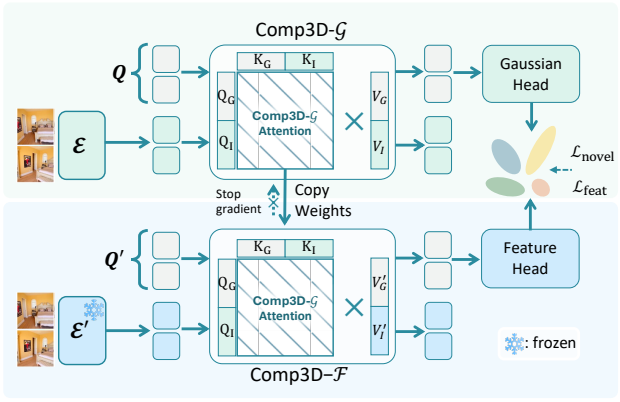


Figure 4. **Comp3D-$\mathcal{F}$ training scheme.** We leverage the learned attention patterns from the Gaussian decoder Comp3D-$\mathcal{G}$ to efficiently learn view-invariant feature decoder Comp3D-$\mathcal{F}$ for feature lifting. We initialize Comp3D-$\mathcal{F}$ (bottom) by copying Comp3D-$\mathcal{G}$'s (top) architecture and keeping the attention weights frozen, using new learnable feature queries Q' and features $\mathcal{E}'$ from any desired encoder. Only the value projections V'_i are trainable, enabling efficient training with $\mathcal{L}_{\text{feat}}$.

Challenges in existing feature lifting approaches. Previous methods [4, 24, 50] for lifting 2D features to 3D face two fundamental challenges: (1) *Correspondence identification*: for each 2D patch whose features need to be lifted, they must identify which 3D Gaussians contribute to rendering that pixel location, often requiring computationally expensive backward mapping operations [4, 19, 24, 50]. (2) *Multi-view inconsistency*: since image encoders extract features independently for each view, patches corresponding to the same 3D region can produce different feature representations across views, requiring additional aggregation schemes [1, 9].

View-invariant feature decoder for any-feature 3D lifting. Surprisingly, we find that the learned self-attention patterns of Comp3D-$\mathcal{G}$ can be used to sidestep both challenges elegantly. For the correspondence identification, each learnable token shows high attention weights to the regions in each image where the 3D Gaussians are projected, removing the need for expensive backward mapping. For multi-view inconsistency, we directly use attention weights as interpolation weights to aggregate inconsistent features, instead of heuristically defining them as in previous works [4].

Building on this insight, we introduce an efficient method to initialize a new *view-invariant feature decoder* Comp3D-$\mathcal{F}$ which leverages the geometric understanding already learned by our Comp3D-$\mathcal{G}$ (Fig. 4). The input of Comp3D-$\mathcal{F}$ is a set of features $\mathbf{F}'_v \in \mathbb{R}^{h\times w\times d'}$ extracted from the same set of input images $\{I_v\}_{v=1}^{V}$, but with a different visual encoder $\mathcal{E}'(\cdot)$ the user wants to lift to 3D. To consider different feature dimension sizes between $\mathcal{E}(\cdot)$ and $\mathcal{E}'(\cdot)$, we also initialize new learnable feature tokens $\mathbf{Q}' \in \mathbb{R}^{N\times d'}$.

To effectively utilize the learned knowledge of the attention weights within our Gaussian decoder, we initialize Comp3D-$\mathcal{F}$ by copying Comp3D-$\mathcal{G}$'s architecture and parameters, but only allow value projections inside the attention operation to be trainable while *freezing the learned attention weights*. This ensures each feature token attends to the same multi-view regions as its corresponding Gaussian token, effectively reusing learned correspondences for feature aggregation. The refined tokens $\bar{\mathbf{Q}}'$ pass through an MLP head to produce multi-view aggregated features $\mathbf{f}_i \in \mathbb{R}^{d'}$ for each Gaussian G_i.

The multi-view aggregated features are attached to their corresponding Gaussians as additional attributes, enabling rendering of novel view feature maps via the same alpha-blending process. We train Comp3D-$\mathcal{F}$ by minimizing the feature similarity loss with the ground truth feature and rendered feature at the target image I_t at target pose π_t:

$$\mathcal{L}_{\text{feat}} = 1 - \cos(\hat{\mathbf{F}}_t/\|\hat{\mathbf{F}}_t\|, \mathbf{F}_t/\|\mathbf{F}_t\|) \tag{4}$$

where $\hat{\mathbf{F}}_t$ is the rendered feature map and $\mathbf{F}_t$ is the ground truth feature, $\cos(\cdot,\cdot)$ indicates the cosine similarity operation, and $\|\cdot\|$ is the L2-norm operation.

4. Experiments

4.1. Implementation details

Here, we specify the architectural details of Comp3D-$\mathcal{G}$. For the visual encoder $\mathcal{E}(\cdot)$, we adopt pretrained VGGT as default. We set $N = 2048$ learnable query tokens and $K = 2$ transformer layers. Following 3DGS [18], we use default Gaussian attributes except setting spherical harmonics degree to 0, which we find to stabilize training with compact Gaussians by modeling only RGB color without view-directional biases. For training, we use 224×224 resolution inputs with photometric loss weights $\lambda_{\text{MSE}} = 1$ and $\lambda_{\text{LPIPS}} = 0.05$. We employ AdamW optimizer [23] with learning rates of 1e-4 for both decoders and 1e-6 for the visual encoder, using cosine annealing (minimum ratio 0.1). The model trains for 450K steps with batch size 8 per GPU across 8 NVIDIA H100 GPUs. For progressive low-pass filtering [17], we decrease s from 10 to 0.3 over the first 4K steps with decay ratio 1/3 every 1K steps.

For feature lifting, we use LSeg [21] and MaskCLIP [6] features for 3D scene understanding, and VGGT tracking features [41], DINOv2 [28], and DINOv3 [36] to demonstrate Comp3D-$\mathcal{F}$'s effectiveness as a view-invariant feature decoder by evaluating in two-view correspondence evaluations following Probe3D [8]. As described in **§ 3.5**, we initialize Comp3D-$\mathcal{F}$ from Comp3D-$\mathcal{G}$ and train only value projections for 1K steps, simultaneously training both decoders.

Table 1. **Comparison of novel view synthesis on RealEstate10K.** Our method maintains competitive results while using far fewer Gaussians. †: Reproduced with VGGT backbone.

Pose-free	Methods	Average #G ↓	Memories↓	FPS↑	PSNR↑	SSIM↑	LPIPS↓
✗	PixelSplat [2]	131K	33.6MB	388.03	23.848	0.806	0.185
	MVSplat [3]	131K	33.6MB	392.6	23.977	0.811	0.176
✓	CoPoNeRF [12]	-	-	0.4	18.938	0.619	0.388
	Splatt3R [37]	131K	33.6MB	393.1	15.318	0.490	0.436
	PF3plat [11]	131K	33.6MB	397.1	21.042	0.739	0.233
	SPFSplat [14]	131K	33.6MB	397.3	25.845	0.852	0.152
	NoPoSplat [48]	131K	33.6MB	369.8	25.033	0.838	0.160
	VGGT+NoPo† [41, 48]	100K	25.6MB	419.8	23.015	0.762	0.187
	Ours	2K	0.1MB	451.7	22.387	0.713	0.259

Table 2. **Comparison of novel view synthesis with 24 input images.** Our method generates fewer Gaussians while achieving competitive or superior quality. †: applied test-time optimization of Gaussians.

Methods	Average #G↓	PSNR↑	SSIM↑	LPIPS↓
AnySplat [15]	2636K	24.105	0.838	0.198
AnySplat† [15]	2636K	27.471	0.898	0.180
Ours	2K	23.797	0.747	0.198
Ours†	27K	29.987	0.916	0.136

4.2. Efficient Novel View Synthesis

In this section, we evaluate novel view synthesis performance on the RealEstate10K dataset [54], following NoPoSplat's [48] protocol where two input images are used to estimate 3D Gaussians and render a target view. We compare with both *pose-dependent* models [2, 3] and *pose-free* models [11, 12, 14, 37, 48]. Since we use VGGT [41] as our visual encoder for Comp3D-$\mathcal{G}$, we additionally train VGGT+NoPo†, which replaces NoPoSplat's MASt3R [20] backbone with VGGT [41] while maintaining NoPoSplat's pipeline to estimate per-pixel Gaussians. Note that Comp3D-$\mathcal{G}$ directly estimates Gaussians from unposed images, falling into the *pose-free* category. For NoPoSplat, VGGT+NoPo†, and ours, we follow NoPoSplat's test-time camera pose optimization, which is only necessary for evaluation. As shown in Tab. 1, despite estimating **50**× fewer Gaussians than per-pixel methods, our approach achieves comparable rendering quality with much faster speeds, validating that our compact Gaussians is sufficient for 3D scene reconstruction.

In Tab. 2, we evaluate with 24 input images, comparing with AnySplat [15], which enables feed-forward 3DGS estimation with more than two views. While AnySplat reduces Gaussians via voxel merging, it produces **100**× more Gaussians than ours with similar performance. We also perform short test-time optimization following 3DGS [18], where ours largely outperform AnySplat (see Ours† and AnySplat†). For AnySplat†, we follow their default test-time optimization method as the 3DGS optimization results in `Out-Of-Memory` due to the number of Gaussians.

Table 3. **Comparison of 3D scene understanding on Scannet.** We lift LSeg [21] and MaskCLIP[6] features from two input views and evaluate open-vocabulary segmentation on target views. Our method generates fewer Gaussians while outperforming feed-forward and per-scene optimization methods trained with substantially more posed inputs. ∗: Features directly extracted from target view images.

Target View

Methods	Feature	Feed Forward	Input Pose	LSeg [21] mIoU↑	Acc↑	PSNR↑	SSIM↑	LPIPS↓	MaskCLIP [32] mIoU↑	Acc↑	PSNR↑	SSIM↑	LPIPS↓	#G↓	Memories↓	FPS↑
LSeg / MaskCLIP* [21, 32]	✓	✗	-	0.506	0.797	-	-	-	0.341	0.667	-	-	-	-	-	-
Feature-3DGS [53]	✓	✗	✓	0.379	0.644	19.83	0.684	0.357	0.353	0.663	17.47	0.612	0.420	1,185K	845.2MB	19.2
CF^3 [19]	✓	✗	✓	0.376	0.657	20.04	0.691	0.359	0.336	0.634	20.14	0.695	0.354	53K	38.4MB	252.4
NoPoSplat [48]	✗	✓	✗	-	-	24.59	0.792	0.228	-	-	24.59	0.792	0.228	131K	33.6MB	369.8
LSM [9]	✓	✓	✗	0.503	0.793	23.32	0.767	0.250	0.286	0.505	22.87	0.737	0.286	131K	61.5MB	254.5
Ours	✓	✓	✗	0.513	0.783	23.89	0.770	0.285	0.369	0.675	23.75	0.763	0.290	2K	4.1MB	243.4

Source View

Methods	Feature	Feed Forward	Input Pose	LSeg [21] mIoU↑	Acc↑	PSNR↑	SSIM↑	LPIPS↓	MaskCLIP [32] mIoU↑	Acc↑	PSNR↑	SSIM↑	LPIPS↓	#G↓	Memories↓	FPS↑
LSeg / MaskCLIP* [21, 32]	✓	✗	-	0.521	0.820	-	-	-	0.344	0.665	-	-	-	-	-	-
Feature-3DGS [53]	✓	✗	✓	0.392	0.655	21.73	0.757	0.314	0.353	0.674	22.25	0.777	0.308	1,185K	845.2MB	19.2
CF^3 [19]	✓	✗	✓	0.390	0.668	22.99	0.804	0.272	0.342	0.642	23.16	0.812	0.265	53K	38.4MB	252.4
NoPoSplat [48]	✗	✓	✗	-	-	25.20	0.812	0.217	-	-	25.20	0.812	0.217	131K	33.6MB	369.8
LSM [9]	✓	✓	✗	0.511	0.798	25.44	0.811	0.214	0.251	0.516	25.01	0.824	0.230	131K	61.5MB	254.5
Ours	✓	✓	✗	0.542	0.803	23.92	0.766	0.278	0.361	0.668	23.39	0.759	0.284	2K	4.1MB	243.4

Table 4. **Comparison of 3D scene understanding on Replica.** We lift LSeg [21] and MaskCLIP[6] features from two input views and evaluate open-vocabulary segmentation on target views. Our method generates fewer Gaussians while outperforming feed-forward methods and achieving comparable results to per-scene optimization methods trained with substantially more posed inputs. ∗: Features directly extracted from target view images.

Target View

Methods	Feature	Feed Forward	Input Pose	LSeg [21] mIoU↑	Acc↑	PSNR↑	SSIM↑	LPIPS↓	MaskCLIP [32] mIoU↑	Acc↑	PSNR↑	SSIM↑	LPIPS↓	#G↓	Memories↓	FPS↑
LSeg / MaskCLIP* [21, 32]	✓	✗	-	0.618	0.887	-	-	-	0.412	0.668	-	-	-	-	-	-
Feature-3DGS [53]	✓	✗	✓	0.730	0.936	35.70	0.972	0.044	0.421	0.686	35.90	0.972	0.045	199K	141.8MB	11.3
CF^3 [19]	✓	✗	✓	0.663	0.918	27.49	0.906	0.132	0.380	0.654	27.49	0.906	0.132	10K	7.13MB	114.7
NoPoSplat [48]	✗	✓	✗	-	-	23.95	0.791	0.149	-	-	23.95	0.791	0.149	131K	33.6MB	369.8
LSM [9]	✓	✓	✗	0.600	0.823	21.86	0.753	0.213	0.241	0.411	17.01	0.637	0.377	131K	61.5MB	254.5
Ours	✓	✓	✗	0.630	0.893	25.43	0.818	0.173	0.416	0.692	25.00	0.809	0.182	2K	4.1MB	243.4

Source View

Methods	Feature	Feed Forward	Input Pose	LSeg [21] mIoU↑	Acc↑	PSNR↑	SSIM↑	LPIPS↓	MaskCLIP [32] mIoU↑	Acc↑	PSNR↑	SSIM↑	LPIPS↓	#G↓	Memories↓	FPS↑
LSeg / MaskCLIP* [21, 32]	✓	✗	-	0.647	0.896	-	-	-	0.414	0.674	-	-	-	-	-	-
Feature-3DGS [53]	✓	✗	✓	0.729	0.930	36.46	0.975	0.043	0.416	0.680	36.63	0.975	0.043	199K	141.8MB	11.3
CF^3 [19]	✓	✗	✓	0.664	0.916	27.95	0.913	0.127	0.375	0.649	27.95	0.913	0.127	10K	7.1MB	114.7
NoPoSplat [48]	✗	✓	✗	-	-	24.36	0.809	0.141	-	-	24.36	0.809	0.141	131K	33.6MB	369.8
LSM [9]	✓	✓	✗	0.600	0.823	19.27	0.760	0.230	0.241	0.439	17.53	0.670	0.377	131K	61.5MB	254.5
Ours	✓	✓	✗	0.649	0.894	25.35	0.815	0.177	0.421	0.695	25.07	0.811	0.185	2K	4.1MB	243.4

Table 5. **Correspondence estimation on Scannet.** We evaluate PCK@10px across two images captured from different angle. Our feature aggregation significantly improves correspondence accuracy across the VGGT-Tracking, DINOv2, and DINOv3-Large.

Methods	θ_0^{15}	θ_{15}^{30}	θ_{30}^{60}	θ_{60}^{180}	Avg.
VGGT-Tracking [41]	53.6	45.5	27.6	9.3	34.0
VGGT-Tracking [41] + Ours	93.5	88.1	70.4	20.3	68.1
DINOv2 [28]	36.6	24.0	19.0	12.7	23.1
DINOv2 [28] + Ours	94.2	89.0	70.5	21.0	68.7
DINOv3-Large [36]	54.9	44.3	32.1	19.3	37.7
DINOv3-Large [36] + Ours	94.2	89.1	70.8	20.9	68.8

Table 6. **Ablation studies for Comp3D-$\mathcal{G}$ on RealEstate10K.**

Lowpass Filter	Unfreeze Backbone	Average PSNR↑	SSIM↑	LPIPS↓
✗	✓	N/A	N/A	N/A
✓	✗	18.441	0.553	0.408
✓	✓	22.387	0.713	0.259

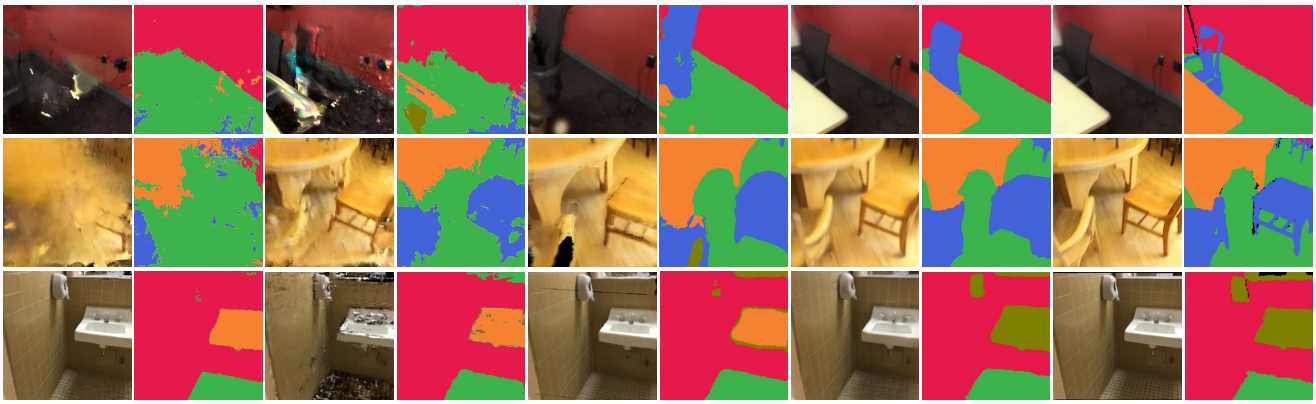
Table 7. **Ablation studies on the number of Gaussian.**

#G	PSNR↑	SSIM↑	LPIPS↓
256	19.713	0.589	0.425
512	20.223	0.607	0.378
1024	20.559	0.619	0.338
2048	20.625	0.623	0.321
4096	19.012	0.568	0.450

4.3. 3D Scene Understanding

Following previous approaches [1, 9], we evaluate 3D scene understanding performance on ScanNet [5] and Replica [38] datasets. To enable open-vocabulary segmentation at novel viewpoints, we lift language-aligned features

CVPR #583



(a) Feature-3DGS (b) CF3 (c) LSM (d) Ours (e) Ground Truth

Figure 5. **Qualitative results of 3D scene understanding on Scannet dataset.** We conduct qualitative comparison for 3D scene understanding via novel view synthesis and open-vocabulary segmentation. When compared to both per-scene optimization ((a),(b)) and feed-forward ((c)) methods, ours show the most high-fidelity renderings and accurate segmentation maps compared to the ground-truth.

Table 8. **Ablation studies for Comp3D-$\mathcal{F}$ on Scannet.**

Comp3D-$\mathcal{F}$	Autoencoder	mIOU↑	Acc.↑	PSNR↑	SSIM↑	LPIPS↓
✗	✗	0.193	0.413	21.77	0.706	0.418
✓	✓	0.512	0.782	23.604	0.756	0.277
✓	✗	0.513	0.783	23.886	0.770	0.285

(LSeg [21] and MaskCLIP [6]) extracted from two input views. We evaluate segmentation performance on features rendered from the estimated 3D Gaussians at target poses.

We compare with both feed-forward approaches [9] and per-scene optimization methods [19, 53]. Note that feed-forward methods use two input images, while optimization-based methods use all scene images for training. As shown in Tab. 3 and Tab. 4, our method outperforms LSM [9] in segmentation while achieving competitive or better reconstruction quality. Despite using far fewer input images, we outperform per-scene optimization methods on ScanNet and show comparable performance on Replica. Surprisingly, our rendered features at target views obtained without accessing the target image outperform features directly extracted from target images using LSeg or MaskCLIP. This validates that the compact Gaussians is much more effective than previous per-pixel Gaussian estimation methods, and verifies that Comp3D-$\mathcal{F}$ effectively aggregates features from the input features, enabling the rendering of a multi-view aware feature map.

4.4. Multi-View Feature Encoding

In this section, we further validate the effectiveness of Comp3D-$\mathcal{F}$ as a view-invariant feature encoder. By lifting the multi-view aggregated features from Comp3D-$\mathcal{F}$ and re-rendering to the input view camera poses, we can achieve view-invariant features from the input views. Following Probe3D [8], we evaluate two-view correspondence performance in ScanNet [5] with PCK @ 10px, where the selected two views are captured within $0{\sim}15°$ (θ_0^{15}), $15{\sim}30°$ (θ_{15}^{30}), $30{\sim}60°$ (θ_{30}^{60}), and $60{\sim}180°$ (θ_{60}^{180}). We compare the features from the visual encoder $\mathcal{E}'(\cdot)$ and the features obtained from our re-rendering process. We evaluate with three different visual encoders, including the tracking features of VGGT [41], DINOv2 [28], and DINOv3 [36]. As shown in Tab. 5, the aggregated features show significant performance improvement in all settings, validating the effectiveness of Comp3D-$\mathcal{F}$ as a view-invariant feature aggregator.

4.5. Ablation Studies

In this section, we go through the ablation studies done for Comp3D-$\mathcal{G}$ and Comp3D-$\mathcal{F}$. In Tab. 6, we validate our core training components. Without progressive low-pass filter control, 3D Gaussians fail to localize within the view frustum, causing training collapse. We also validate that freezing the visual encoder $\mathcal{E}(\cdot)$ prevents effective Gaussian generation in appropriate regions. In Tab. 7, we investigate the optimal number of learnable tokens (Gaussians) N. Reconstruction performance gradually improves as Gaussians increase to 2048. However, with 4096 Gaussians, training becomes unstable, leading to degradation. We hypothesize that having larger number of Gaussians at sub-optimal positions is more prone to falling into local minima, as discussed in prior per-scene optimization methods [17]. Based on these findings, we set the number of Gaussians to 2048 for all experiments. In Tab. 9, we analyze different visual encoders $\mathcal{E}(\cdot)$ for Comp3D-$\mathcal{G}$ using VGGT [41] and DINOv3 [36]. Although DINOv3 lacks explicit geometric supervision, its features are effectively aggregated by learnable queries in the transformer to generate coherent 3D Gaussians. These results reveal the potential that our framework can also be learned without features with strong geometric priors, where previous per-pixel Gaussian estimation frameworks struggle to learn [48].

CVPR
#583

Table 9. **Ablation stuides on visual encoder choice.**

Backbones	PSNR↑	SSIM↑	LPIPS↓
VGGT [41]	22.387	0.713	0.259
DINOv3 [36]	20.292	0.631	0.313

5. Conclusion

We presented Comp3D-$\mathcal{G}$, a framework that learns compact 3D Gaussians with learnable query tokens, which can discover geometrically meaningful regions through self-attention, resolving redundancy and high computational overhead issues from previous dense per-pixel predictions. In addition, we present Comp3D-$\mathcal{F}$, which leverages the attention weights learned in Comp3D-$\mathcal{G}$ to effectively decode multi-view consistent features. By combining both Comp3D-$\mathcal{G}$ and Comp3D-$\mathcal{F}$ we achieve competitive performance in novel view synthesis and outperform previous works in 3D scene understanding while being significantly more efficient in memory and rendering speeds. We believe that our approach opens up new directions for feed-forward reconstruction and scene understanding, mitigating the need for per-pixel estimations.

References

[1] Neil Burgess. Spatial memory: how egocentric and allocentric combine. *Trends in cognitive sciences*, 10(12):551–557, 2006. 2, 3, 5, 7

[2] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. 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*, pages 19457–19467, 2024. 3, 4, 6

[3] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *European Conference on Computer Vision*, pages 370–386. Springer, 2024. 3, 6

[4] Jiahuan Cheng, Jan-Nico Zaech, Luc Van Gool, and Danda Pani Paudel. Occam's lgs: An efficient approach for language gaussian splatting. *arXiv preprint arXiv:2412.01807*, 2024. 2, 5

[5] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 7, 8

[6] Xiaoyi Dong, Jianmin Bao, Yinglin Zheng, Ting Zhang, Dongdong Chen, Hao Yang, Ming Zeng, Weiming Zhang, Lu Yuan, Dong Chen, et al. Maskclip: Masked self-distillation advances contrastive language-image pretraining. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10995–11005, 2023. 6, 7, 8

[7] Yilun Du, Cameron Smith, Ayush Tewari, and Vincent Sitzmann. Learning to render novel views from wide-baseline stereo pairs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4970–4980, 2023. 3

[8] Mohamed El Banani, Amit Raj, Kevis-Kokitsi Maninis, Abhishek Kar, Yuanzhen Li, Michael Rubinstein, Deqing Sun, Leonidas Guibas, Justin Johnson, and Varun Jampani. Probing the 3d awareness of visual foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21795–21806, 2024. 6, 8

[9] Zhiwen Fan, Jian Zhang, Wenyan Cong, Peihao Wang, Renjie Li, Kairun Wen, Shijie Zhou, Achuta Kadambi, Zhangyang Wang, Danfei Xu, et al. Large spatial model: End-to-end unposed images to semantic 3d. *Advances in neural information processing systems*, 37:40212–40229, 2024. 1, 2, 3, 5, 7, 8

[10] Elisabetta Fedele, Boyang Sun, Leonidas Guibas, Marc Pollefeys, and Francis Engelmann. Superdec: 3d scene decomposition with superquadric primitives. *arXiv preprint arXiv:2504.00992*, 2025. 2, 3, 4

[11] Sunghwan Hong, Jaewoo Jung, Heeseong Shin, Jisang Han, Jiaolong Yang, Chong Luo, and Seungryong Kim. Pf3plat: Pose-free feed-forward 3d gaussian splatting. *arXiv preprint arXiv:2410.22128*, 2024. 2, 3, 4, 6

[12] Sunghwan Hong, Jaewoo Jung, Heeseong Shin, Jiaolong Yang, Seungryong Kim, and Chong Luo. Unifying correspondence pose and nerf for generalized pose-free novel view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20196–20206, 2024. 3, 6

[13] Guichen Huang, Ruoyu Wang, Xiangjun Gao, Che Sun, Yuwei Wu, Shenghua Gao, and Yunde Jia. Longsplat: Online generalizable 3d gaussian splatting from long sequence images. *arXiv preprint arXiv:2507.16144*, 2025. 3

[14] Ranran Huang and Krystian Mikolajczyk. No pose at all: Self-supervised pose-free 3d gaussian splatting from sparse views. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 27947–27957, 2025. 2, 3, 6

[15] Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, et al. Anysplat: Feed-forward 3d gaussian splatting from unconstrained views. *arXiv preprint arXiv:2505.23716*, 2025. 3, 6

[16] Mohammad Mahdi Johari, Yann Lepoittevin, and François Fleuret. Geonerf: Generalizing nerf with geometry priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18365–18375, 2022. 3

[17] Jaewoo Jung, Jisang Han, Honggyu An, Jiwon Kang, Seonghoon Park, and Seungryong Kim. Relaxing accurate initialization constraint for 3d gaussian splatting. 2024. 5, 6, 8

[18] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 3, 4, 5, 6

[19] Hyunjoon Lee, Joonkyu Min, and Jaesik Park. Cf3: Compact and fast 3d feature fields. *arXiv preprint arXiv:2508.05254*, 2025. 1, 2, 3, 5, 7, 8

[20] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision*, pages 71–91. Springer, 2024. 3, 6

[21] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. *arXiv preprint arXiv:2201.03546*, 2022. 6, 7, 8

[22] Wanhua Li, Yujie Zhao, Minghan Qin, Yang Liu, Yuanhao Cai, Chuang Gan, and Hanspeter Pfister. Langsplatv2: High-dimensional 3d language gaussian splatting with 450+ fps. *arXiv preprint arXiv:2507.07136*, 2025. 3

[23] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[24] Juliette Marrie, Romain Ménégaux, Michael Arbel, Diane Larlus, and Julien Mairal. Ludvig: Learning-free uplifting of 2d visual features to gaussian splatting scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7440–7450, 2025. 2, 5

[25] Sheng Miao, Jiaxin Huang, Dongfeng Bai, Xu Yan, Hongyu Zhou, Yue Wang, Bingbing Liu, Andreas Geiger, and Yiyi Liao. Evolsplat: Efficient volume-based gaussian splatting for urban view synthesis. In *Proceedings of the Computer*

Vision and Pattern Recognition Conference, pages 11286–11296, 2025. 3

[26] Tom Monnier, Jake Austin, Angjoo Kanazawa, Alexei Efros, and Mathieu Aubry. Differentiable blocks world: Qualitative 3d decomposition by rendering primitives. *Advances in Neural Information Processing Systems*, 36:5791–5807, 2023. 3

[27] Liangliang Nan and Peter Wonka. Polyfit: Polygonal surface reconstruction from point clouds. In *Proceedings of the IEEE international conference on computer vision*, pages 2353–2361, 2017. 3

[28] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 6, 7, 8

[29] Despoina Paschalidou, Ali Osman Ulusoy, and Andreas Geiger. Superquadrics revisited: Learning 3d shape parsing beyond cuboids. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10344–10353, 2019. 3, 4

[30] Despoina Paschalidou, Luc Van Gool, and Andreas Geiger. Learning unsupervised hierarchical part decomposition of 3d objects from a single rgb image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1060–1070, 2020. 3

[31] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20051–20060, 2024. 3

[32] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 7

[33] William Shen, Ge Yang, Alan Yu, Jansen Wong, Leslie Pack Kaelbling, and Phillip Isola. Distilled feature fields enable few-shot language-guided manipulation. *arXiv preprint arXiv:2308.07931*, 2023. 2

[34] Yu Sheng, Jiajun Deng, Xinran Zhang, Yu Zhang, Bei Hua, Yanyong Zhang, and Jianmin Ji. Spatialsplat: Efficient semantic 3d from sparse unposed images. *arXiv preprint arXiv:2505.23044*, 2025. 2

[35] Roger N Shepard and Jacqueline Metzler. Mental rotation of three-dimensional objects. *Science*, 171(3972):701–703, 1971. 2

[36] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025. 6, 7, 8, 9

[37] Brandon Smart, Chuanxia Zheng, Iro Laina, and Victor Adrian Prisacariu. Splatt3r: Zero-shot gaussian splatting from uncalibrated image pairs. *arXiv preprint arXiv:2408.13912*, 2024. 3, 6

[38] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 7

[39] Shubham Tulsiani, Hao Su, Leonidas J Guibas, Alexei A Efros, and Jitendra Malik. Learning shape abstractions by assembling volumetric primitives. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2635–2643, 2017. 3

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2, 3

[41] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025. 2, 3, 6, 7, 8, 9

[42] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2021. 3

[43] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024. 3

[44] Weijie Wang, Donny Y Chen, Zeyu Zhang, Duochao Shi, Akide Liu, and Bohan Zhuang. Zpressor: Bottleneck-aware compression for scalable feed-forward 3dgs. *arXiv preprint arXiv:2505.23734*, 2025. 3

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

[46] Yunsong Wang, Tianxin Huang, Hanlin Chen, and Gim Hee Lee. Freesplat: Generalizable 3d gaussian splatting towards free view synthesis of indoor scenes. *Advances in Neural Information Processing Systems*, 37:107326–107349, 2024. 3

[47] Haofei Xu, Anpei Chen, Yuedong Chen, Christos Sakaridis, Yulun Zhang, Marc Pollefeys, Andreas Geiger, and Fisher Yu. Murf: multi-baseline radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20041–20050, 2024. 3

[48] Botao Ye, Sifei Liu, Haofei Xu, Xueting Li, Marc Pollefeys, Ming-Hsuan Yang, and Songyou Peng. No pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images. *arXiv preprint arXiv:2410.24207*, 2024. 2, 3, 4, 6, 7, 8

[49] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4578–4587, 2021. 3

[50] Karim Abou Zeid, Kadir Yilmaz, Daan de Geus, Alexander Hermans, David Adrian, Timm Linder, and Bastian Leibe.

CVPR
#583

Dino in the room: Leveraging 2d foundation models for 3d segmentation. *arXiv preprint arXiv:2503.18944*, 2025. 2, 5

[51] Chuanrui Zhang, Yingshuang Zou, Zhuoling Li, Minmin Yi, and Haoqian Wang. Transplat: Generalizable 3d gaussian splatting from sparse multi-view images with transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9869–9877, 2025. 3

[52] Shengjun Zhang, Xin Fei, Fangfu Liu, Haixu Song, and Yueqi Duan. Gaussian graph network: Learning efficient and generalizable gaussian representations from multi-view images. *Advances in Neural Information Processing Systems*, 37:50361–50380, 2024. 3

[53] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang, and Achuta Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21676–21685, 2024. 3, 7, 8

[54] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018. 6

[55] Chen Ziwen, Hao Tan, Kai Zhang, Sai Bi, Fujun Luan, Yicong Hong, Li Fuxin, and Zexiang Xu. Long-lrm: Long-sequence large reconstruction model for wide-coverage gaussian splats. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4349–4359, 2025. 3