MuSASplat: Efficient Sparse-View 3D Gaussian Splats via Lightweight Multi-Scale Adaptation

Muyu Xu[1], Fangneng Zhan[2], Xiaoqin Zhang[3], Ling Shao[4], Shijian Lu[1]
[1]Nanyang Technological University, Singapore
[2]Harvard University, USA
[3]Zhejiang University of Technology, China
[4]UCAS-Terminus AI Lab, University of Chinese Academy of Sciences, China
muyu001@e.ntu.edu.sg, fnzhan@seas.harvard.edu, zhangxiaoqinnan@gmail.com
ling.shao@ieee.org, Shijian.Lu@ntu.edu.sg

Abstract

Sparse-view 3D Gaussian splatting seeks to render high-quality novel views of 3D scenes from a limited set of input images. While recent pose-free feed-forward methods leveraging pre-trained 3D priors have achieved impressive results, most of them rely on full fine-tuning of large Vision Transformer (ViT) backbones and incur substantial GPU costs. In this work, we introduce MuSASplat, a novel framework that dramatically reduces the computational burden of training pose-free feed-forward 3D Gaussian splats models with little compromise of rendering quality. Central to our approach is a lightweight Multi-Scale Adapter that enables efficient fine-tuning of ViT-based architectures with only a small fraction of training parameters. This design avoids the prohibitive GPU overhead associated with previous full-model adaptation techniques while maintaining high fidelity in novel view synthesis, even with very sparse input views. In addition, we introduce a Feature Fusion Aggregator that integrates features across input views effectively and efficiently. Unlike widely adopted memory banks, the Feature Fusion Aggregator ensures consistent geometric integration across input views and meanwhile mitigates the memory usage, training complexity, and computational costs significantly. Extensive experiments across diverse datasets show that MuSASplat achieves state-of-the-art rendering quality but has significantly reduced parameters and training resource requirements as compared with existing methods.

Introduction

Though 3D reconstruction with neural radiance field (NeRF) (Mildenhall et al. 2021) and 3D Gaussian Splatting (3DGS) (Kerbl et al. 2023) has achieved impressive novel-view synthesis, most existing methods still rely heavily on hundreds of posed images and per-scene optimization. The recent feed-forward networks (Charatan et al. 2024; Chen et al. 2024b,c) obviate per-scene optimization but still require known camera poses, usually estimated by structure-from-motion algorithms such as COLMAP (Schonberger and Frahm 2016) which is computationally expensive and ill-suited under sparse-view settings. It remains a grand challenge to achieve robust and high-quality 3D scene reconstruction under the conditions of 1) few-shot unposed images, 2) without per-scene optimization, and 3) without high demand of GPU resources.

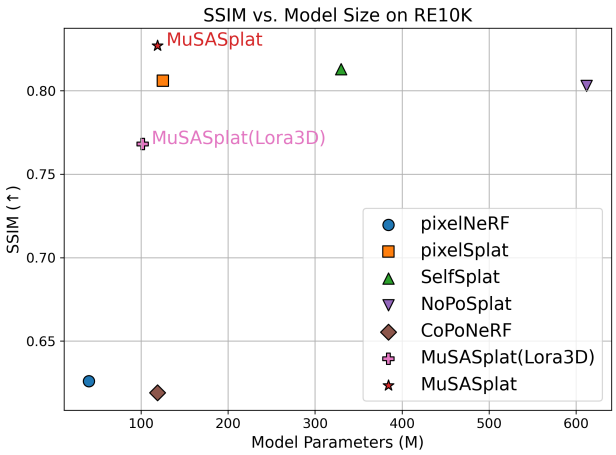

Figure 1: Unlike state-of-the-art sparse-view 3D rendering approaches that require either camera poses (like pixelSplat) or computationally intensive full model fine-tuning (like NoPoSplat), the proposed MuSASplat achieves superior reconstruction performance with unposed images and much reduced network parameters and computation costs. The experiments evaluate SSIM versus model size (in millions of parameters) on the RE10K dataset. We also report MuSASplat (LoRA3D), a variant where our adapter is replaced with LoRA3D (Lu et al. 2024), showing that the proposed Multi-Scale Adapter provides a clear advantage in accuracy.

Recent pose-free methods (Smart et al. 2024; Ye et al. 2024; Hong et al. 2024a; Chen et al. 2024a) leverage pre-trained feed-forward 3D geometry networks (Zhang et al. 2025; Wang et al. 2024; Leroy, Cabon, and Revaud 2024) to overcome the reliance on camera poses and extract point clouds directly from sparse views. For example, NoPoSplat (Ye et al. 2024) combines DUSt3R (Wang et al. 2024) with a 3D Gaussian head and achieves competitive performance through full-model fine-tuning. However, the full-model fine-tuning demands substantial GPU resources due to the need for adapting large vision transformer (ViT) back-

bones.

This paper presents *MuSASplat*, a lightweight and scalable framework for pose-free sparse-view 3D reconstruction. MuSASplat explicitly targets the computational bottlenecks of existing methods, significantly reducing both training time and GPU memory usage while maintaining high-quality rendering, as shown in Figure 1. It comes with two novel designs. First, it introduces a *Multi-Scale Adapter* that enables efficient fine-tuning of ViT-based backbones. Unlike prior fine-tuning techniques such as LoRA (Hu et al. 2022) that apply low-rank adaptations to linear projections, our adapter leverages the spatial structure implicit in ViT token sequences. Specifically, we rebuild the patchified token sequence back into a spatial feature map, preserving the image's spatial layout, and apply a set of multi-scale depth-wise convolutions. These spatial-aware operations empower the adapter to better capture local geometric context across different receptive fields. By inserting a small number of lightweight convolutional layers into the transformer pipeline, this design achieves expressive adaptation capacity with minimal parameter overhead and significantly lowers GPU consumption compared to full-model fine-tuning.

Second, we introduce a *Feature Fusion Aggregator* to replace conventional memory bank structures as used in mainstream multi-view 3D reconstruction. Existing memory bank designs, such as those in Spann3R (Wang and Agapito 2024) and CUT3R (Wang et al. 2025b), typically require sequentially encoding and decoding image pairs while updating a shared memory. Such processes are inefficient and memory-intensive, especially while handling dense input views. In contrast, our aggregator performs *batch-wise* encoding and decoding of all input views simultaneously. It inserts a single lightweight aggregation module between the encoder and decoder to fuse features across viewpoints, enabling efficient global geometric consistency without iterative memory updates. This not only improves training throughput but also dramatically reduces GPU memory usage. Experiments on multiple datasets show that MuSASplat achieves state-of-the-art performance in pose-free sparse-view 3D reconstruction, while requiring only a fraction of the training cost of existing methods.

The contributions of this work can be summarized in three major aspects:

- We propose *MuSASplat*, a lightweight framework for pose-free sparse-view 3D Gaussian splats that achieves strong rendering performance with minimal GPU overhead.
- We design a *Multi-Scale Adapter* for ViT fine-tuning, which incorporates spatial-aware multi-scale depth-wise convolutions to improve scene understanding with a minimal number of parameters.
- We propose a *Feature Fusion Aggregator* that enables efficient, batch-wise multi-view feature fusion without the need for iterative memory updates, improving both training efficiency and memory usage.

Related Work

2.1 Generalizable Novel View Synthesis

NeRF (Mildenhall et al. 2021) inaugurated neural rendering by modelling scenes as continuous radiance fields, but its reliance on hundreds of posed images and per-scene optimisation limits practical use. 3D Gaussian Splatting (3DGS) (Kerbl et al. 2023) accelerates rendering by analytically rasterising anisotropic Gaussians, yet it inherits the same dense-capture requirement. To alleviate per-scene training, a growing body of feed-forward methods (Yu et al. 2021; Wang et al. 2021; Chen et al. 2021; Johari, Lepoittevin, and Fleuret 2022; Xu et al. 2023; Charatan et al. 2024; Chen et al. 2024b) build cost volumes or exploit epipolar geometry, but they still assume accurate camera poses and appreciable viewpoint overlap. The introduction of DUSt3R (Wang et al. 2024) moves a step further by predicting calibrated point maps directly from unposed stereo pairs. Built on this backbone, Splatt3R (Smart et al. 2024) attaches a Gaussian decoder yet updates only a handful of weights, whereas NoPoSplat (Ye et al. 2024) fully fine-tunes the network and attains the current best fidelity at the cost of heavy GPU usage. The present work continues the pose-free direction but replaces full-model tuning with an efficient adapter and aggregator, thereby slashing training cost while preserving high quality.

2.2 Parameter-Efficient Fine-Tuning

Low-rank adaptation (LoRA) (Hu et al. 2022) demonstrates that inserting rank-constrained matrices into otherwise frozen Transformers transfers knowledge with only a few per-cent of the original parameters; QR-LoRA (Yang et al. 2025) further stabilises the optimiser by factorising the LoRA weights using QR decomposition. More recently, the "5% >100%" study (Yin et al. 2025) shows that multi-scale depth-wise adapters can rival or even surpass full fine-tuning on vision tasks while adding roughly five per-cent parameters, and Adapter-X (Li et al. 2024) extends the idea to 3D point clouds. In the 3D domain specifically, LoRA3D (Lu et al. 2024) affirms that parameter-efficient techniques can adapt neural rendering pipelines with limited hardware. Guided by these findings, our Multi-Scale Adapter rebuilds ViT tokens back into feature maps and applies multi-scale depth-wise convolutions, injecting spatial priors at a minimal parameter cost.

2.3 Efficient Multi-View Feature Aggregation

Consistent fusion of unposed views is commonly addressed with external memory structures. Spann3R (Wang and Agapito 2024) and CUT3R (Wang et al. 2025b) sequentially process image pairs and update a global memory bank, which leads to $V - 1$ times encoder–decoder passes and high memory footprints. PreF3R (Chen, Yang, and Yang 2024) incrementally merges variable-length sequences but still performs iterative updates. In contrast, our Feature Fusion Aggregator encodes all views in a single batch and inserts just one lightweight fusion layer, so aggregation latency remains constant and GPU memory usage stays low even when the number of input images grows.

Method

Let $\mathcal{I} = \{I^v\}_{v=1}^V$ be V unposed RGB images. Our MuSASplat predicts a set of anisotropic Gaussians

$$\mathcal{G} = \{\mathrm{G}_n\}_{n=1}^N,\ \mathrm{G}_n = (\boldsymbol{\mu}_n, \alpha_n, \boldsymbol{\Sigma}_n, \mathbf{sh}_n), \tag{1}$$

which are splatted for novel-view synthesis. Only three lightweight modules, the *Multi-Scale Adapter (MuSA)*, the *Feature Fusion Aggregator (FFA)*, and the Gaussian/point heads, are trainable. The large ViT backbone remains frozen. Figure 2 sketches the whole pipeline.

3.1 Preliminaries

3D Gaussian Splatting (3D-GS). 3DGS (Kerbl et al. 2023) represents a scene's radiance field using a set of anisotropic 3D Gaussians, each encoding the radiance of its surrounding region. Each Gaussian is parameterized by its mean position $\boldsymbol{\mu} \in \mathbb{R}^3$, opacity $\alpha \in \mathbb{R}$, covariance $\boldsymbol{\Sigma} \in \mathbb{R}^{3\times3}$, and SH $\boldsymbol{sh} \in \mathbb{R}^k$ representing color. However, traditional 3D-GS fits Gaussian splats to a scene through iterative optimization and is unsuitable for generalizable feed-forward models.

Recent generalizable 3D-GS methods (Charatan et al. 2024; Chen et al. 2024b; Szymanowicz et al. 2024; Szymanowicz, Rupprecht, and Vedaldi 2024) directly predict pixel-aligned 3D Gaussians from a set of N images $I = \{I_i\}_{i=1}^N$. However, these approaches rely on known camera parameters, which limits their applicability to calibrated settings. In contrast, our method estimates per-view point maps and projects them to a canonical space, enabling consistent 3D Gaussian supervision directly from uncalibrated RGB images.

DUSt3R. DUSt3R (Wang et al. 2024) is a ViT-based approach that jointly solves camera calibration and 3D reconstruction using only images. Given two input images, it predicts dense 3D point clouds, referred to as *pointmaps*, which establish a per-pixel 2D-to-3D mapping. Specifically, a pointmap $X_{a,b}$ maps each pixel $i = (u, v)$ in image I_a to a corresponding 3D point $X_{a,b}^{u,v}$, expressed in the coordinate system of camera C_b. By regressing two pointmaps $X_{a,a}$ and $X_{b,a}$ in a shared coordinate frame (C_a), DUSt3R effectively performs joint calibration and 3D reconstruction.

3.2 Multi-Scale Adapter (MuSA)

Low-Rank Adaptation (LoRA) (Hu et al. 2022) is a widely adopted strategy for parameter-efficient fine-tuning. As illustrated in Figure 3, LoRA modifies a frozen model by injecting two lightweight linear layers into the residual path, effectively learning a low-rank update to the output of a linear projection. This approach works well in language modeling and some 2D vision tasks where token-wise dependencies dominate. However, its effectiveness is limited in 3D vision tasks such as novel view synthesis, where spatial coherence and geometric consistency are essential.

The core limitation lies in LoRA's inability to reintroduce spatial priors. Since a vanilla ViT processes each image as a sequence of independent patch tokens, it discards the 2D structure within and between patches. LoRA, operating only on linear transformations of these 1D tokens, does not recover any notion of proximity, local structure, or continuity across the image plane. Consequently, when applied to sparse-view 3D reconstruction, LoRA-equipped models often generate inconsistent geometry, overfit to local appearance cues, or hallucinate non-existent structures.

To overcome these limitations, we propose MuSA: a Multi-Scale Adapter that restores spatial reasoning to the ViT encoding process, while maintaining parameter efficiency. The key idea is to reinterpret the token sequence as a feature map and process it using depth-wise convolutions with multiple kernel sizes. In this way, MuSA enables the model to reason about local and global spatial relationships, which LoRA cannot capture natively.

Concretely, for each input image I^v, the frozen encoder produces a token sequence $\mathbf{X}^v \in \mathbb{R}^{N\times C}$, where $N = HW/P^2$ is the number of patches and C the channel dimension. We first project the tokens to a reduced dimension via a 1×1 linear projection:

$$\mathbf{Y}^v = \mathrm{reshape}(\mathbf{X}^v\mathbf{W}_\downarrow) \in \mathbb{R}^{C'\times h\times w}, \tag{2}$$

where $C' = C/r$ with reduction ratio $r = 4$, and $h \times w = N$ corresponds to the spatial grid of patches.

We then apply three depth-wise convolutions with kernel sizes 3, 5, and 7, and average their outputs to obtain:

$$\mathbf{Z}^v = \tfrac{1}{3} \sum_{k\in\{3,5,7\}} \mathrm{DWConv}_k(\mathbf{Y}^v). \tag{3}$$

This multi-scale aggregation captures both fine and coarse spatial dependencies within the patch grid. The result is passed through a point-wise convolution and a GELU activation, then projected back to the original dimension:

$$\widehat{\mathbf{X}}^v = (\mathrm{GELU}\,(\mathrm{PWConv}(\mathbf{Z}^v)))\,\mathbf{W}_\uparrow. \tag{4}$$

We finally add the residual to the frozen token stream:

$$\mathbf{X}_{\text{out}}^v = \mathbf{X}^v + \widehat{\mathbf{X}}^v. \tag{5}$$

All learnable parameters are zero-initialized so that the model starts from the behavior of the pre-trained backbone, ensuring stability at the start of fine-tuning.

3.3 Mini-Grid Branch in MuSA

One might suspect that further gains could come from recovering *intra-patch* structure, whose information is irretrievably lost once a 16×16 region is collapsed into a single token. We therefore insert a *mini-grid branch* that maps each token into a small $p \times p$ feature map (here p=4), applies one depth-wise convolution, and averages the result back into a token-sized residual. Because the original spatial arrangement of pixels inside a patch is unknown, the branch effectively has to *relearn* a permutation mapping from scratch, a task that is under-constrained by sparse 3D supervision. In practice, its weights remain near-zero and the branch yields no measurable improvement on any dataset we tested. The negative result is nevertheless informative: it reveals that the primary benefit of our adapter is not recovering sub-patch detail but rather exploiting the cross-patch spatial relations that ViT discards. In other words, the small, multi-scale convolutions are sufficient to inject the geometric cues that the 3D Gaussian decoder needs.

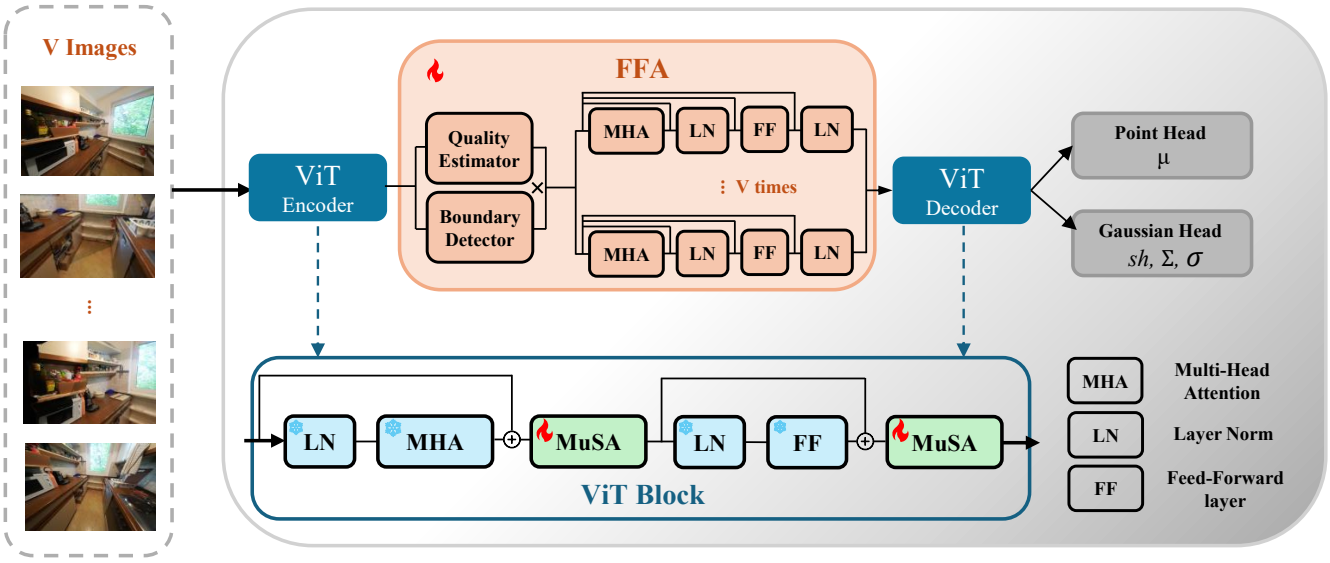

Figure 2: Overview of the MuSASplat architecture. Given unposed multi-view input images, we extract per-view features using a ViT encoder. ViT consists of multiple stacked ViT blocks, which are the basic computational units containing self-attention and feed-forward sublayers. Our proposed Multi-Scale Adapter (MuSA) modules are inserted into the blocks to enhance spatial awareness while introducing minimal extra parameters. The resulting features from different views are fused in a single forward pass by the Feature Fusion Aggregator (FFA), which adaptively integrates geometric information using view-specific quality estimator and boundary detector as elaborated in section 3.4. The fused feature is then decoded by a lightweight ViT decoder and passed to a point head and a Gaussian head to generate the parameters of 3D Gaussian primitives for rendering.

3.4 Feature Fusion Aggregator (FFA)

Some existing multi-view pose-free pipelines, such as Spann3R (Wang and Agapito 2024) and CUT3R (Wang et al. 2025b), rely on an external memory bank that is updated once per image pair. The memory must be read and written $O(V)$ times, which slows training when the number of input views $V > 2$ and stores features from every view, inflating GPU memory.

Instead of iterative memory updates, we process *all* input views simultaneously in each batch. The encoded features from all views are stacked as $\mathbf{F} \in \mathbb{R}^{B\times V\times L\times C}$, where V is the number of views. To assess the per-token quality, a lightweight MLP (*quality estimator*) estimates a confidence score for each token:

$$q_\ell^v = \sigma(\mathbf{w}_q^\top \mathbf{f}_\ell^v). \tag{6}$$

In parallel, a second MLP (*boundary detector*) detects boundary-view indicators from global pooled features $\bar{\mathbf{f}}^v = \frac{1}{L}\sum_\ell \mathbf{f}_\ell^v$, producing a binary mask $b^v \in \{0, 1\}$ to identify potentially valuable views. A tunable weight $\lambda > 1$ boosts their contribution. The final attention weights are modulated by both token-wise confidence and boundary-view weight:

$$w_\ell^v = \begin{cases} q_\ell^v \cdot \lambda & \text{if } b^v = 1 \\ q_\ell^v & \text{otherwise} \end{cases}. \tag{7}$$

Low-confidence tokens ($q_\ell^v < \tau$) are masked. For each query view v, the remaining views serve as key/value pairs:

$$\mathbf{A}^v = \mathrm{softmax}\left(\frac{\mathbf{Q}^v(\mathbf{K}^v)^\top}{\sqrt{d}} + \log \mathbf{M}^v\right), \quad \widetilde{\mathbf{F}}^v = \mathbf{A}^v\mathbf{V}^v. \tag{8}$$

Finally, we fuse the attended features with the original using an MLP residual connection:

$$\widehat{\mathbf{F}}^v = \mathbf{F}^v + \mathrm{MLP}\left(\left[\mathbf{F}^v, \widetilde{\mathbf{F}}^v\right]\right). \tag{9}$$

Because the aggregator is invoked only once, its latency is constant with respect to V, and its memory footprint is bounded by the batch itself. Replacing the memory bank with this lightweight aggregator reduces peak GPU memory by 0.3× and accelerates training by 4.2×.

Experiments

We validate the effectiveness of our proposed MuSASplat through extensive experiments. We first present our training setup and datasets, followed by main comparisons against prior methods, and finally evaluate the contribution of each component via ablations and targeted substitution studies.

4.1 Training and Evaluation Setup

Datasets. To ensure consistency with prior work, we adopt the same training and evaluation protocol as NoPoSplat (Ye et al. 2024). We train on three large-scale pose-free datasets: RealEstate10K (RE10K) (Zhou et al. 2018) and ACID (Liu et al. 2021), which collectively span indoor and outdoor

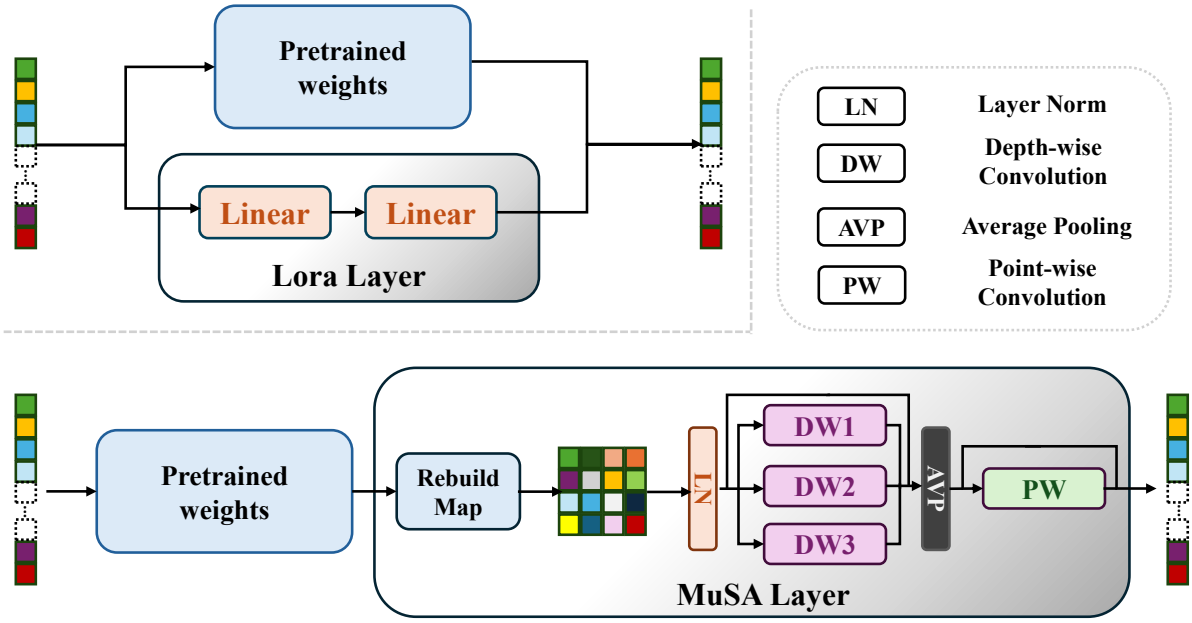

Figure 3: Comparison between LoRA and our proposed MuSA layer. **Top**: LoRA injects a low-rank residual update into the frozen pre-trained model via two linear layers, operating purely in the token space without spatial awareness. **Bottom**: MuSA reconstructs the spatial layout of tokens into a feature map, applies depth-wise convolutions at multiple kernel sizes to capture local structure, and projects the adapted features back into the token stream. This design enables spatial reasoning while maintaining parameter efficiency.

scenes across diverse conditions. Evaluation on RE10K and ACID follows NoPoSplat's overlap-based partitioning.

Baselines. We compare against a range of generalizable 3D reconstruction methods under both two-view and multi-view settings. For the two-view input case, we evaluate pose-required models such as PixelNeRF (Yu et al. 2021), AttnRend (Du et al. 2023), PixelSplat (Charatan et al. 2024), and MVSplat (Chen et al. 2024b), as well as recent pose-free methods including Splatt3R (Smart et al. 2024), CoPoNeRF (Hong et al. 2024b), NoPoSplat (Ye et al. 2024), and SelfSplat (Kang et al. 2025). In addition, to further assess the effectiveness of our proposed feature fusion aggregator, we conduct experiments using five-view input and compare against methods that rely on traditional memory banks. Specifically, we include PREF3R (Chen, Yang, and Yang 2024) and CUT3R (Wang et al. 2025b) equipped with a Gaussian head. Since PREF3R is not open-sourced, we re-implement its architecture based on the original paper, using Spann3R (Wang and Agapito 2024) as the backbone and appending a Gaussian rendering head. The model is retrained under the same setting as ours for a fair comparison. These comparisons provide a comprehensive evaluation across both input regimes and highlight the generalizability of our approach.

MuSA Adapter Extension to Other Models. To test the generalizability of our adapter design, we inject our proposed MuSA Adapter into the transformer blocks of both NoPoSplat and SelfSplat, resulting in NoPoSplat-MuSA and SelfSplat-MuSA variants. These are compared against LoRA3D (Lu et al. 2024) inserted into the same backbones under identical settings. All adapter comparisons are conducted on RE10K.

Implementation Details. Our model is implemented in PyTorch and trained using AdamW with learning rate $5 \cdot 10^{-5}$, weight decay 0.05, and gradient clipping of 0.5. We freeze all pre-trained ViT backbone weights and only train the Multi-Scale Adapter, Feature Fusion Aggregator, and Gaussian/point heads. All images are resized to 256×256 resolution. Training takes place on a single RTX3090 GPU with a batch size of 4. Training lasts 75K iterations.

4.2 Main Results

Quantitative Results. We report PSNR, SSIM, and LPIPS (Zhang et al. 2018) on both the RE10K and ACID datasets in Table 1. MuSASplat consistently outperforms all pose-required baselines across both datasets, achieving +0.35 dB higher PSNR than the best pose-required method (MVSplat) on RE10K and +0.29 dB on ACID. Although NoPoSplat achieves slightly better PSNR on RE10K, our method delivers more balanced performance across all metrics, including a lower LPIPS on ACID (0.189 vs. 0.205), indicating more perceptually faithful reconstructions. In the 5-view setting, MuSASplat surpasses both memory-based pose-free baselines by a significant margin, outperforming PREF3R by +2.5 dB PSNR and CUT3R by +3.0 dB on RE10K. It also achieves the best SSIM and LPIPS on both datasets, showcasing the effectiveness of our feed-forward multi-view aggregation in high-visibility settings.

	Method	Params(M)	RE10K			ACID		
			PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
2-View Reconstruction								
Pose-Required	pixelNeRF (Yu et al. 2021)	-	19.824	0.626	0.485	20.323	0.561	0.533
	AttnRend (Du et al. 2023)	-	22.664	0.762	0.269	24.475	0.730	0.287
	pixelSplat (Charatan et al. 2024)	-	23.848	0.806	0.185	*25.819*	0.779	0.195
	MVSplat (Chen et al. 2024b)	-	23.977	*0.811*	0.176	25.512	0.773	*0.196*
Pose-Free	DUSt3R (Wang et al. 2024)	-	15.382	0.447	0.432	16.286	0.411	0.447
	MASt3R (Leroy, Cabon, and Revaud 2024)	-	14.907	0.431	0.452	16.179	0.409	0.461
	Splatt3R (Smart et al. 2024)	80	15.318	0.490	0.425	16.754	0.472	0.448
	CoPoNeRF (Hong et al. 2024b)	286	18.938	0.619	0.226	20.950	0.606	0.406
	SelfSplat (Kang et al. 2025)	330	*24.220*	0.813	0.188	**26.710**	**0.801**	*0.196*
	NoPoSplat (Ye et al. 2024)	612	**24.833**	0.803	**0.172**	25.961	0.781	0.205
	MuSASplat (Ours)	119	24.324	**0.827**	*0.182*	25.802	*0.780*	**0.189**
5-View Reconstruction								
Pose-Free	CUT3R + GS Head (Wang et al. 2025b)	170	*21.751*	*0.701*	0.284	*22.981*	*0.782*	*0.223*
	PREF3R (Chen, Yang, and Yang 2024)	161	22.181	0.749	*0.298*	23.212	0.796	0.191
	MuSASplat (Ours)	119	**24.721**	**0.832**	**0.155**	**26.362**	**0.812**	**0.171**

Table 1: Quantitative comparison on RE10K and ACID datasets under 2-view and 5-view settings. **The best results are formatted in bold, the second-best results are underlined, and the third-best results are italicized.** MuSASplat has a competitive performance with both pose-required and pose-free baselines in 2-view settings, and demonstrates superior performance in 5-view comparisons against memory-based methods like PREF3R and CUT3R. To ensure fair comparison of model efficiency, the parameter count column is only reported for pose-free methods. DUSt3R and MASt3R are pre-trained on external datasets for point cloud prediction and are included solely for inference-time comparison, so their parameter counts are not listed. For the 5-view setting, we freeze the backbone of each model and retrain only the remaining modules to better isolate and compare the efficiency and performance of memory bank versus feature fusion aggregator designs.

Qualitative Results. As shown in Fig. 4, MuSASplat recovers more complete geometry and precise textures in both indoor and outdoor scenes. Notably, fewer floaters and holes appear around edges and occlusion boundaries as highlighted with the red boxes.

4.3 Component Analysis and Ablation Studies

Ablation studies. We conduct ablation studies to examine the contributions of each module in MuSASplat, including the Multi-Scale Adapter, the Feature Fusion Aggregator, and the point cloud augmentation strategy. We also construct a variant named MuSASplat (LoRA3D) by replacing our adapter with LoRA3D (Hu et al. 2022) which uses LoRA to fine-tune large 3D geometric foundation models. As shown in Table 2, removing the Multi-Scale Adapter leads to a significant performance drop of 4.2 dB. Using Lora3D instead of our Multi-Scale Adapter also leads to a drop of 1.4 dB, confirming the importance of spatial-aware feature adaptation. Without the Aggregator, performance drops by 0.8 dB despite having similar parameter count and training speed, showing that single-pass feature fusion is more effective than independent view processing.

Finally, another variant MuSASplat(Memory Bank) obtained by replacing the Aggregator with a memory bank mechanism similar to Spann3R causes a large reduction in training efficiency: the frame processing speed drops from 1.81 it/s (iteration per second) to 0.43 it/s, and the model size increases by over 30%. However, this change does not yield any performance gain, highlighting the efficiency advantage of our feed-forward design.

Variant	PSNR↑	Speed (it/s)	Params (M)
MuSASplat (full)	24.32	1.81	119
w/o Adapter	20.12	2.02	92
w/o Aggregator	23.50	1.96	113
MuSASplat(Lora3D)	22.89	1.65	102
MuSASplat(Memory Bank)	23.66	0.43	155

Table 2: Ablation study on MuSA and FFA. Removing either module or using Lora3D instead of our MuSA layer hurts performance. Replacing our aggregator with the memory bank further drastically reduces training efficiency.

Adapter Transferability compared with Lora. To assess the general applicability of the MuSA Adapter across different architectures, we compare it against LoRA3D by integrating both adapters into two distinct base models: NoPoSplat and SelfSplat. Specifically, we insert either MuSA or

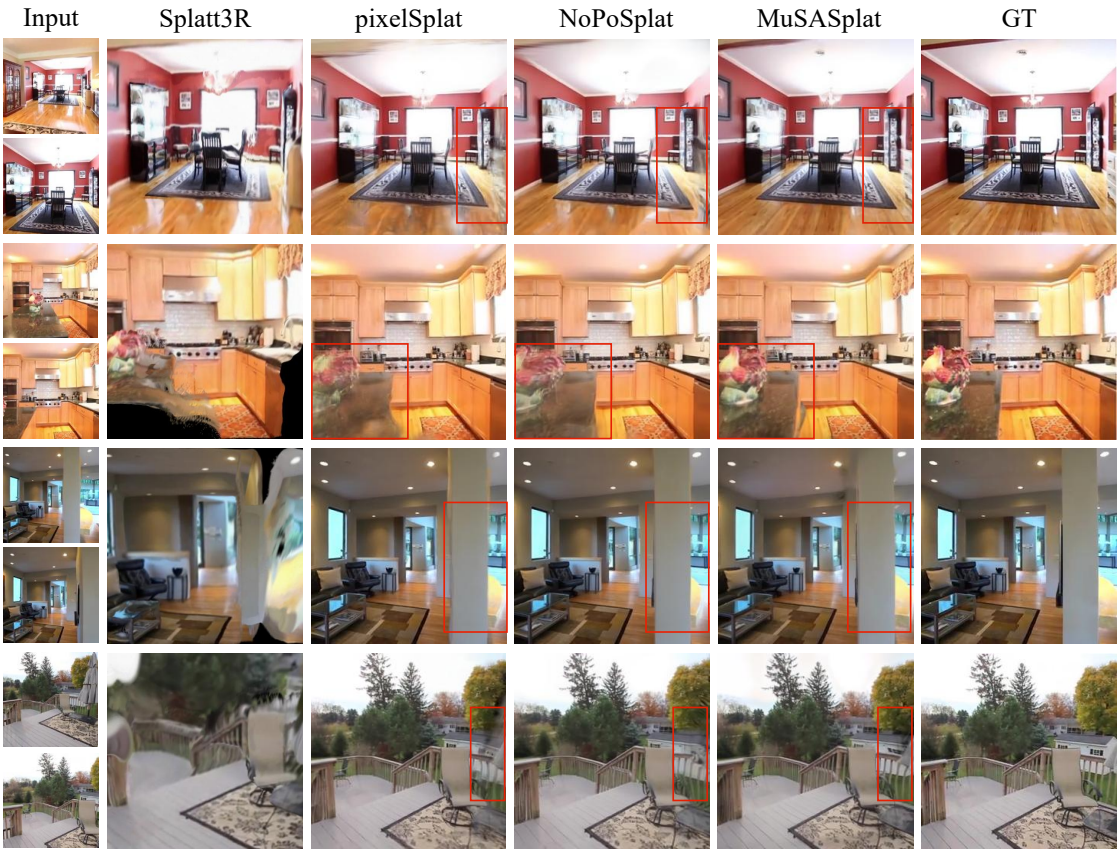

Figure 4: Qualitative comparison on RE10K. Compared to NoPoSplat and PixelSplat, our MuSASplat yields more complete geometry and fewer artifacts in occluded regions. The major differences are highlighted with red boxes.

Model Variant	PSNR↑	SSIM↑	LPIPS↓
NoPoSplat-LoRA3D	21.72	0.741	0.283
NoPoSplat-MuSA	**23.39**	**0.794**	**0.221**
SelfSplat-LoRA3D	21.60	0.748	0.271
SelfSplat-MuSA	**23.14**	**0.799**	**0.223**

Table 3: MuSA Adapter vs. LoRA3D on RE10K. Replacing LoRA3D by MuSA Adapter consistently improves performance in both NoPoSplat and SelfSplat.

LoRA3D into the backbone of each model and freeze the pretrained weights of the backbones, training each model with the rest of the architecture unchanged. This allows for a fair evaluation of the fine-tuning mechanism itself. As shown in Table 3, replacing LoRA3D with MuSA consistently improves performance in terms of PSNR, SSIM, and LPIPS on both models. While SelfSplat employs a Swin Transformer backbone that preserves spatial structures, we adapt our MuSA design by omitting the reshaping step to match the architectural differences. Despite this, MuSA still yields notable gains, demonstrating its versatility and effectiveness across various encoder types.

Conclusion

We propose MuSASplat, a lightweight framework for pose-free sparse-view 3D Gaussian reconstruction that delivers competitive rendering quality with substantially lower GPU memory usage and training cost. Our Multi-Scale Adapter enables spatially aware fine-tuning of ViT backbones, while the Feature Fusion Aggregator efficiently fuses multi-view features in a single pass. Together, they achieve strong performance with 5× fewer parameters than existing pose-free baselines. Despite its effectiveness, MuSASplat is currently tailored to ViT-based 3D models and may not fully exploit newer architectures such as VGGT (Wang et al. 2025a). Furthermore, it is designed for static scene reconstruction, and extending it to dynamic settings remains a promising direction.

Acknowledgments

This work is funded by the Ministry of Education, Singapore, under the Tier-1 project scheme with a project number RT18/22.

References

Charatan, D.; Li, S. L.; Tagliasacchi, A.; and Sitzmann, V. 2024. 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*, 19457–19467.

Chen, A.; Xu, Z.; Zhao, F.; Zhang, X.; Xiang, F.; Yu, J.; and Su, H. 2021. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF international conference on computer vision*, 14124–14133.

Chen, Y.; Potamias, R. A.; Ververas, E.; Song, J.; Deng, J.; and Lee, G. H. 2024a. ZeroGS: Training 3D Gaussian Splatting from Unposed Images. *arXiv preprint arXiv:2411.15779*.

Chen, Y.; Xu, H.; Zheng, C.; Zhuang, B.; Pollefeys, M.; Geiger, A.; Cham, T.-J.; and Cai, J. 2024b. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *European Conference on Computer Vision*, 370–386. Springer.

Chen, Y.; Zheng, C.; Xu, H.; Zhuang, B.; Vedaldi, A.; Cham, T.-J.; and Cai, J. 2024c. Mvsplat360: Feed-forward 360 scene synthesis from sparse views. *arXiv preprint arXiv:2411.04924*.

Chen, Z.; Yang, J.; and Yang, H. 2024. Pref3r: Pose-free feed-forward 3d gaussian splatting from variable-length image sequence. *arXiv preprint arXiv:2411.16877*.

Du, Y.; Smith, C.; Tewari, A.; and Sitzmann, V. 2023. Learning to render novel views from wide-baseline stereo pairs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4970–4980.

Hong, S.; Jung, J.; Shin, H.; Han, J.; Yang, J.; Luo, C.; and Kim, S. 2024a. Pf3plat: Pose-free feed-forward 3d gaussian splatting. *arXiv preprint arXiv:2410.22128*.

Hong, S.; Jung, J.; Shin, H.; Yang, J.; Kim, S.; and Luo, C. 2024b. 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*, 20196–20206.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2): 3.

Johari, M. M.; Lepoittevin, Y.; and Fleuret, F. 2022. Geonerf: Generalizing nerf with geometry priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18365–18375.

Kang, G.; Yoo, J.; Park, J.; Nam, S.; Im, H.; Shin, S.; Kim, S.; and Park, E. 2025. SelfSplat: Pose-free and 3D prior-free generalizable 3D Gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 22012–22022.

Kerbl, B.; Kopanas, G.; Leimkühler, T.; and Drettakis, G. 2023. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4): 139–1.

Leroy, V.; Cabon, Y.; and Revaud, J. 2024. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision*, 71–91. Springer.

Li, M.; Ye, P.; Huang, Y.; Zhang, L.; Chen, T.; He, T.; Fan, J.; and Ouyang, W. 2024. Adapter-x: A novel general parameter-efficient fine-tuning framework for vision. *arXiv preprint arXiv:2406.03051*.

Liu, A.; Tucker, R.; Jampani, V.; Makadia, A.; Snavely, N.; and Kanazawa, A. 2021. Infinite nature: Perpetual view generation of natural scenes from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 14458–14467.

Lu, Z.; Yang, H.; Xu, D.; Li, B.; Ivanovic, B.; Pavone, M.; and Wang, Y. 2024. Lora3d: Low-rank self-calibration of 3d geometric foundation models. *arXiv preprint arXiv:2412.07746*.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106.

Schonberger, J. L.; and Frahm, J.-M. 2016. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4104–4113.

Smart, B.; Zheng, C.; Laina, I.; and Prisacariu, V. A. 2024. Splatt3r: Zero-shot gaussian splatting from uncalibrated image pairs. *arXiv preprint arXiv:2408.13912*.

Szymanowicz, S.; Insafutdinov, E.; Zheng, C.; Campbell, D.; Henriques, J. F.; Rupprecht, C.; and Vedaldi, A. 2024. Flash3D: Feed-Forward Generalisable 3D Scene Reconstruction from a Single Image. *arXiv preprint arXiv:2406.04343*.

Szymanowicz, S.; Rupprecht, C.; and Vedaldi, A. 2024. Splatter image: Ultra-fast single-view 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10208–10217.

Wang, H.; and Agapito, L. 2024. 3d reconstruction with spatial memory. *arXiv preprint arXiv:2408.16061*.

Wang, J.; Chen, M.; Karaev, N.; Vedaldi, A.; Rupprecht, C.; and Novotny, D. 2025a. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 5294–5306.

Wang, Q.; Wang, Z.; Genova, K.; Srinivasan, P. P.; Zhou, H.; Barron, J. T.; Martin-Brualla, R.; Snavely, N.; and Funkhouser, T. 2021. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4690–4699.

Wang, Q.; Zhang, Y.; Holynski, A.; Efros, A. A.; and Kanazawa, A. 2025b. Continuous 3d perception model with persistent state. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 10510–10522.

Wang, S.; Leroy, V.; Cabon, Y.; Chidlovskii, B.; and Revaud, J. 2024. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20697–20709.

Xu, M.; Zhan, F.; Zhang, J.; Yu, Y.; Zhang, X.; Theobalt, C.; Shao, L.; and Lu, S. 2023. Wavenerf: Wavelet-based generalizable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 18195–18204.

Yang, J.; Ma, Y.; Di, D.; Li, H.; Chen, W.; Xie, Y.; Cui, J.; Yang, X.; and Zuo, W. 2025. QR-LoRA: Efficient and Disentangled Fine-tuning via QR Decomposition for Customized Generation. *arXiv preprint arXiv:2507.04599*.

Ye, B.; Liu, S.; Xu, H.; Li, X.; Pollefeys, M.; Yang, M.-H.; and Peng, S. 2024. No pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images. *arXiv preprint arXiv:2410.24207*.

Yin, D.; Hu, L.; Li, B.; Zhang, Y.; and Yang, X. 2025. 5%¿ 100%: Breaking performance shackles of full fine-tuning on visual recognition tasks. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 20071–20081.

Yu, A.; Ye, V.; Tancik, M.; and Kanazawa, A. 2021. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4578–4587.

Zhang, J.; Li, Y.; Chen, A.; Xu, M.; Liu, K.; Wang, J.; Long, X.-X.; Liang, H.; Xu, Z.; Su, H.; et al. 2025. Advances in Feed-Forward 3D Reconstruction and View Synthesis: A Survey. *arXiv preprint arXiv:2507.14501*.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 586–595.

Zhou, T.; Tucker, R.; Flynn, J.; Fyffe, G.; and Snavely, N. 2018. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*.