Pose-Free 3D Gaussian Splatting for Ordered and Unordered Frame Sequence Scene Reconstruction

Huiyang Li
South China University of Technology
Guangzhou, China
1510834081@qq.com

Yongwei Nie*
South China University of Technology
Guangzhou, China
nieyongwei@scut.edu.cn

Hushan Song
South China University of Technology
Guangzhou, China
2635346060@qq.com

Ping Li
The Hong Kong Polytechnic University
Hong Kong, China
p.li@polyu.edu.hk

Abstract

3D scene reconstruction and novel view synthesis are core tasks in computer vision and graphics. 3D Gaussian Splatting (3DGS) has advanced these tasks, but relies on Structure From Motion (SfM) to obtain camera poses, limiting robustness in texture-sparse scenes and efficiency. In this paper, we propose a pose-free 3DGS-based reconstruction algorithm for ordered frame sequences, and extend it to unordered sequences. For ordered sequences, we design a progressive training framework that iteratively alternates between camera pose optimization (using a noise-robust feature loss and pre-trained feature encoder) and 3D Gaussian model optimization (guided by an effective depth loss and multi-view consistent depth information). For unordered sequences that have sparse overlapping with existing views, the proposed algorithm introduces several innovations building upon the progressive 3DGS training framework. For camera pose optimization, we introduce a point matching loss. For 3D Gaussian model optimization, we additionally design a local point cloud management strategy to promote the reasonable growth of 3D Gaussian in error reconstruction areas. Experiments on Tanks and Temples, CO3D-V2, and LLFF datasets show that our methods outperform state-of-the-art (SOTA) pose-free methods. Our work reduces reliance on SfM while enhancing reconstruction quality and pose estimation accuracy.

***Keywords:** 3D Reconstruction, Novel View Synthesis, Camera Pose Estimation, Neural Radiance Field, 3D Gaussian Splatting*

*Corresponding author.

1. Introduction

Image-based 3D reconstruction is a long-standing and extensively studied problem in the field of computer vision. It aims to recover the three-dimensional geometric structure and appearance texture information of target objects or scenes from a series of two-dimensional images, representing the content of objects or scenes through virtual 3D data. 3D reconstruction serves as a technical foundation for many research fields and holds significant importance for the development of areas such as novel view synthesis, object editing, SLAM, and depth estimation. The goal of novel view synthesis is to generate high-quality images from previously unobserved camera perspectives of objects or scenes, realistically presenting their content from new viewpoints. Improving the quality of 3D reconstruction and the effectiveness of novel view synthesis has long been a key focus in the field of 3D reconstruction. Traditional 3D reconstruction methods (e.g., SfM, Multi-View Stereo (MVS)) involve multi-stage operations (feature detection/matching), leading to low efficiency and suboptimal 3D model accuracy.

Mildenhall et al. [16] proposed NeRF, which utilizes a neural network to implicitly represent the geometric and color information of a scene. It introduced volume rendering technology for rendering and achieved end-to-end training relying solely on images as supervision. This approach enables detailed modeling of scenes and the rendering of high-quality images. Although the NeRF demonstrates outstanding performance, its training and rendering speeds remain relatively slow, making real-time rendering difficult and, to some extent, limiting its application and practical deployment. With the emergence of 3DGS [9], the quality of novel view synthesis and 3D reconstruction has been further improved, while also reducing training time. 3DGS

employs a set of explicit 3D Gaussians to model the scene. By optimizing various attributes of these 3D Gaussians and densifying them, it reconstructs the scene's geometric structure and fits its textural information, making real-time rendering coupled with high-quality output possible.

However, 3DGS methods rely on traditional reconstruction techniques, such as SfM algorithms, to estimate the camera poses of the input images. The reconstruction and estimation process of SfM involves multiple stages, including feature extraction and feature matching. If the extracted features are inaccurate, noise can be introduced, thereby compromising the accuracy of the camera pose estimation. Furthermore, SfM performs poorly in scenarios with weak textures or extensive repetitive textures, limiting its applicable scenarios. The entire process is also time-consuming. While recent feed-forward transformers (e.g., VGGT [23]) offer rapid pose initialization, they often struggle with domain gaps in out-of-distribution scenes and lack the fine-grained detail recoverability of per-scene optimization methods. These factors collectively lead to reduced robustness and sub-optimal reconstruction quality for 3DGS methods in complex real-world situations, which, to some extent, hinders the advancement of 3D reconstruction. Therefore, researching 3D reconstruction without known camera poses is highly significant, as it can reduce reliance on algorithms like SfM. Since camera poses are crucial for 3D reconstruction, the task of pose-free 3D reconstruction is also exceptionally challenging.

To eliminate SfM dependency and address existing pose-free 3DGS limitations, we propose a pose-free algorithm for ordered frame sequences: a progressive framework that iteratively optimizes next-frame poses (via pre-trained global 3DGS for geometric guidance and motion-sensitive feature loss) and 3D Gaussians (guided by multi-view depth alignment to reduce artifacts).

We also extend the above progressive algorithm to real-world unordered sequences (discontinuous trajectories, large viewpoint jumps) with three adaptations. (1) point matching loss (LoFTR-based) for pose estimation accuracy, (2) adaptive depth-aware reconstruction, (3) local point cloud management (promote Gaussian densification in error regions, prune redundant Gaussians) to solve sparse initial point clouds.

In summary, our contributions can be highlighted as follows: 1) For ordered frame sequences, we design a progressive training framework that iteratively optimizes camera poses and 3D Gaussian models, reducing SfM reliance and solving artifacts/background collapse issues. 2) For unordered frame sequences, we extend the progressive framework with three core improvements: LoFTR-based matching loss, scale-invariant depth loss, and local point cloud management. 3) Experiments on Tanks and Temples [11], CO3D-V2 [19] (ordered) and LLFF [3] (unordered) show our methods outperform SOTA pose-free baselines (e.g., CF-3DGS [7]) in novel view synthesis (e.g., +1.66dB PSNR on ordered, +5.94dB PSNR on unordered) and camera pose estimation (e.g., 19.51% lower RPE_t on ordered), even approaching pose-known methods' performance.

2. Related work

3D Gaussian Splatting (3DGS). 3DGS models radiance fields via 3D Gaussians and uses tile-based rasterization for real-time rendering. As an explicit 3D representation, it initializes from point clouds and optimizes under supervision with only multi-view images and their camera poses as input. This enables high-quality novel view synthesis, fast convergence, and real-time high-resolution rendering—though its rendering and reconstruction still have room for improvement. Specifically, trained 3DGS models often produce unsatisfactory results when rendering at resolutions mismatched to training data (e.g., lower resolutions) or from distant camera viewpoints. To address this, MS3DGS [26] employs multi-scale 3D Gaussians for scene representation: smaller Gaussians for high-resolution rendering and larger ones for lower-resolution tasks. Other works focus on reducing 3DGS computational costs by compressing Gaussian representations (without sacrificing synthesis quality). For example, C3DGS [12] uses residual vector quantization (RVQ) [28] to encode Gaussian geometric attributes (3D axis scales and rotation angles), while Mini-Splatting [6] achieves compression through Gaussian sampling (avoiding artifacts from pruning). In large-scale 3D reconstruction, Yan et al.'s StreetGaussians [25] uses two Gaussian models: one for static scene content and a dynamic one for moving objects. CityGaussian [14], by contrast, adopts a divide-and-conquer strategy for large-scale scenes and introduces level-of-detail rendering based on camera-Gaussian distance.

Pose-Free 3D Reconstruction. Approaches aiming to bypass SfM generally fall into two categories. NeRF-based methods (e.g., NeRFmm [20], BARF [13], SC-NeRF [17]) typically treat camera poses as optimizable parameters to be jointly trained with the radiance field, sometimes incorporating depth priors (e.g., Nope-NeRF [2]) to enforce geometric consistency. However, these implicit methods often suffer from slow convergence and geometric ambiguities. With the advent of 3D Gaussian Splatting, recent works such as CF-3DGS [7] and InstantSplat [5] have significantly improved efficiency but still face challenges with unstable initialization or cumulative errors. Most recently, HT-3DGS [8] proposed a hierarchical training strategy combined with Video Frame Interpolation (VFI) to merge multiple local 3DGS models, achieving state-of-the-art results on video sequences. However, HT-3DGS heavily relies on temporal continuity for interpolation and model merging, which restricts its applicability to unordered image collections or

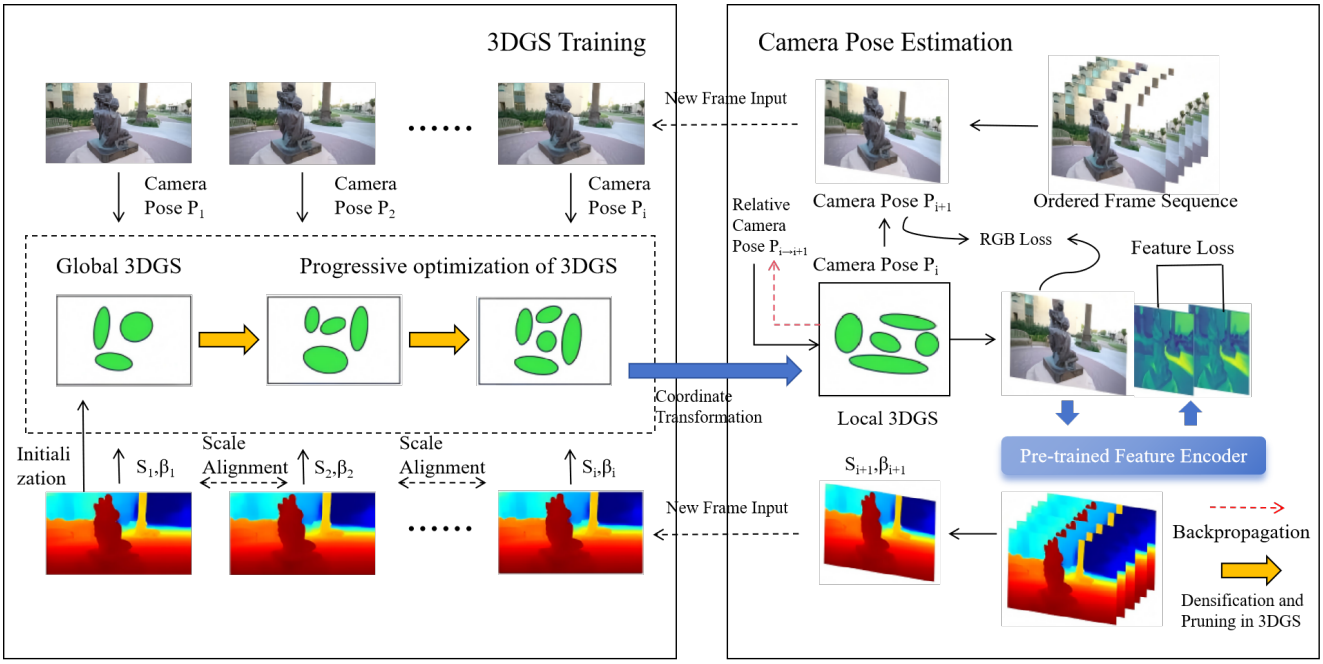

Figure 1. **Overview of Pose-Free 3D Reconstruction Framework for Ordered Frame Sequences.** We propose a progressive framework that iteratively alternates between camera pose estimation (via global 3DGS and noise-robust feature loss) and 3D Gaussian optimization (guided by multi-view consistent depth). Key modules include pre-trained feature encoder, depth alignment, and Gaussian densification, enabling reduced reliance on SfM compared to baselines like CF-3DGS.

scenes with large viewpoint changes. Concurrently, Meuleman et al. [15] proposed an on-the-fly reconstruction framework tailored for large-scale unbounded scenes, utilizing incremental optimization. In a different direction, large-scale feed-forward models like VGGT [23] leverage massive pre-training data to directly predict camera poses and geometry in a single forward pass. However, these approaches have distinct focuses: Meuleman et al. target large-scale traversal rather than object-centric fidelity, while VGGT relies heavily on the domain distribution of training data. In contrast, our method focuses on test-time optimization for high-fidelity object reconstruction without requiring massive pre-training or strictly ordered data streams.

3. Methodology

This paper adopts a progressive training framework for both ordered and unordered frame sequences, iteratively optimizing camera poses and the global 3D Gaussian model. The core designs differ based on the sequence type to address their unique challenges.

3.1. Pose-Free 3D Reconstruction Algorithm for Ordered Frame Sequences

Progressive Training Framework. The framework initializes the global 3D Gaussian model using the first frame's depth map (predicted by DPT [18]) and color map. Through the inverse process of perspective projection, pixels are recovered into 3D points. The number of points in the point cloud is equal to the number of pixels in the image, and the color attributes of the point cloud are initialized with the corresponding pixel colors. Based on this point cloud, the 3D Gaussian splatting collection is initialized.

Then, as shown in Figure 1, as ordered frames are fed into the system sequentially, for each subsequent frame I_{i+1}, we propose a (1) *Noise-Robust Camera Pose Estimation* method to estimate the camera pose of I_{i+1} based on the pre-optimized global 3D Gaussian Splatting model (which remains fixed during pose estimation). The relative camera pose optimization module uses global 3D Gaussians to render intermediate images and compute RGB/feature losses. This frame along with its estimated pose are then added to the training set. Then we propose (2) *Multi-View Consistent Depth-Guided 3D Gaussian Optimization* to update the global 3D Gaussian model using all frames in the training set. This iterative process continues until all frames have been processed. The 3DGS densification and pruning module adjusts Gaussian distribution based on multi-view depth alignment, ensuring the model gradually fits the entire scene.

Noise-Robust Camera Pose Estimation. This algorithm (Figure 2) is used to optimize the relative pose $T_{i\to i+1}$ between the frame I_i (with known pose) and the frame I_{i+1}.

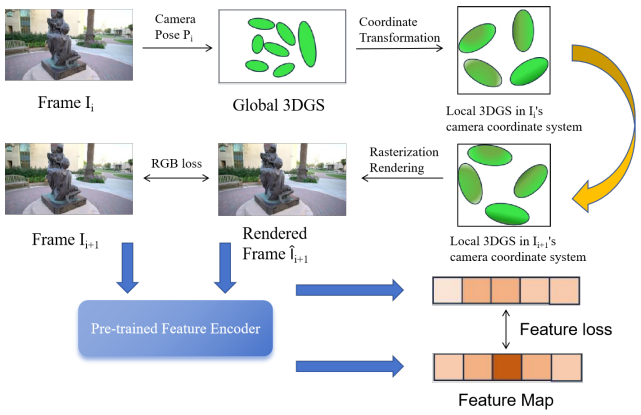

Figure 2. **Overview of the Relative Camera Pose Optimization Module for Ordered Frame Sequences.** We use the global 3DGS of the known frame to obtain the pose and image of the next frame through coordinate transformation and rendering, and introduce feature loss and RGB loss to optimize the relative camera pose, thereby achieving noise-robust pose estimation.

Let P_i be the camera pose of I_i. With the above annotations, the globally trained 3DGS (which has already modeled the geometric structure of local scene regions) is first transformed to the camera coordinate system of I_i to obtain a local 3DGS of image I_i through P_i. Then, the local 3DGS of I_i is transformed to the local coordinate system of I_{i+1} through the relative pose $T_{i\to i+1}$. At the same time, the camera pose P_{i+1} of I_{i+1} can be computed by $P_{i+1} = T_{i\to i+1} \odot P_i$. With P_{i+1}, we project the local 3DGS of I_{i+1} to the image plane to render an image $\hat{I}_{i+1}$. Finally, we compute losses between $\hat{I}_{i+1}$ and the grounding truth image I_{i+1}, and use the losses to optimize the relative transformation $T_{i\to i+1}$.

To prevent noises, two types of losses are introduced for joint optimization including RGB loss and feature loss, especially the feature loss.

RGB loss. The RGB loss combines L_1 loss and D-SSIM loss to align the rendered image with the real image I_{i+1},

$$L_{rgb} = (1-\lambda_1)L_1 + \lambda_1 L_{D\text{-}SSIM}, \tag{1}$$

where $\lambda_1 = 0.2$.

Feature Loss. A pre-trained VGG16-based feature encoder (fine-tuned with Triplet loss [4]) extracts robust features from the rendered and real images. The loss is defined as the cosine similarity between the feature maps, reducing interference from noise and illumination changes,

$$L_{\text{feature}} = \frac{1}{N}\sum_j \left(1 - \frac{m_j \cdot \tilde{m}_j}{\|m_j\|_2 \cdot \|\tilde{m}_j\|_2}\right), \tag{2}$$

where m_j and $\tilde{m}_j$ are features from the real and rendered images, respectively, and N is the number of feature pixels. The feature encoder adopts the network architecture designed in [1] for camera pose estimation, specifically using the pre-trained VGG16 network (without the final classification layer) as the feature extraction backbone.

The above feature loss is effective in preventing noise because we re-train the VGG feature encoder using clean and noisy image pairs using the following way. 1) Dataset construction: For each frame, we create triplets consisting of the real image, a noisy version (regional adaptive noise: Gaussian blur, pixel offset ±2, isotropic blur), and a frame from an adjacent view. 2) Training objective: We use Triplet loss to minimize the feature distance between the real image and noisy image, while maximizing the distance between the real image and adjacent-view image (margin = 1). 3) Optimization: The encoder (VGG16 backbone without classifier) is fine-tuned for 50 epochs using Adam [10] optimizer, ensuring features are invariant to noise but discriminative to view changes.

Finally, as mentioned above, after optimizing $T_{i\to i+1}$, the pose of I_{i+1} is computed as,

$$P_{i+1} = T_{i\to i+1} \odot P_i, \tag{3}$$

where P_i is the pose of I_i.

Multi-View Consistent Depth-Guided 3D Gaussian Optimization. After adding a new image and estimating its camera pose, we now re-optimize the 3D Gaussian Splatting (3DGS). Originally, 3DGS is trained solely using RGB images. In this work, we additionally supervise the training of 3DGS using depth maps. Specifically, given any image I_i, we first employ an existing depth estimation network to predict the depth map of that image D_i, where DPT [18] is adopted for outdoor scenes (with better robustness to large-scale geometric variations), while ZoeDepth [1] is used for texture-sparse scenes (to mitigate depth ambiguity in low-texture regions). At the same time, based on the camera pose of the image, we project the 3DGS onto the image plane and render a depth map $\hat{D}_i$.

Naively, the L1 loss of depth can be computed between D_i and $\hat{D}_i$. However, the depth maps D_i estimated by the monocular depth estimation networks lack multi-view consistency, because every depth map is predicted from a single image only and the depth maps estimated from different viewpoints are not necessarily on the same scale. To solve the problem, we apply a learnable scale s_i and offset β_i to D_i (The scale s_i and offset β_i are optimized jointly with Gaussian parameters (learning rate = $5e^{-4}$)) to align depth scales across views) to reduce the impact of noisy depth values and ensure multi-view consistency. The aligned depth map D_i^* is computed as:

$$D_i^* = s_i D_i + \beta_i \tag{4}$$

The L_{depth} loss that guides the model optimization is thus defined as:

$$L_{\text{depth}} = \|\hat{D}_i - D_i^*\| \tag{5}$$

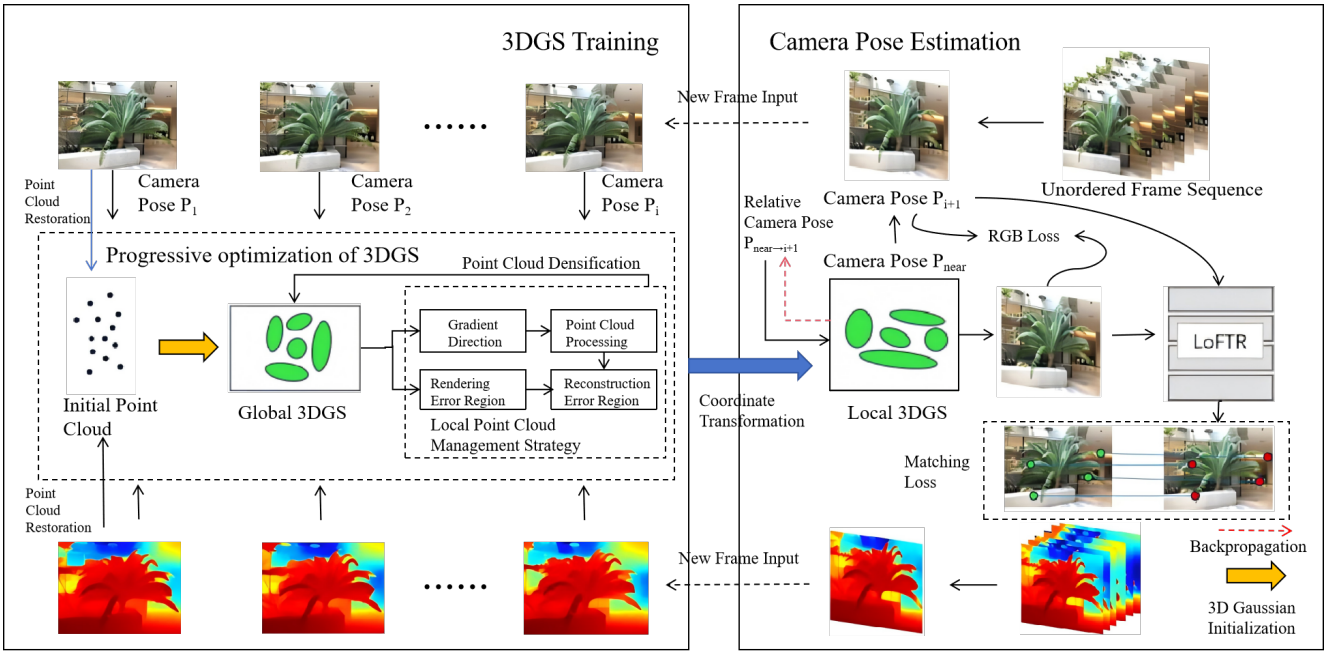

Figure 3. **Overview of Pose-Free 3D Reconstruction Framework for Unordered Frame Sequences.** Compared to our ordered-frame framework, this method adds VGG16-based similar frame selection, LoFTR-based matching loss, and local point cloud management modules—addressing large inter-frame motion and sparse Gaussian distribution in unordered scenes, and outperforming CF-3DGS in pose estimation accuracy.

The total loss for 3D Gaussian optimization is,

$$L = L_{\text{rgb}} + \lambda_2 L_{\text{depth}} \tag{6}$$

where $\lambda_2 = 0.05$ (it is determined via ablation experiments, which balances geometric accuracy (reducing artifacts) and texture fidelity (avoiding over-smoothing).) and L_{rgb} is as shown in Eq. 1.

3.2. Pose-Free 3D Reconstruction Algorithm for Unordered Frame Sequences

Now we extend our method to unordered frame sequences. Unordered frame sequences have arbitrary camera trajectories, leading to large inter-frame motion. We extend the progressive framework with three key adaptations, whose overall workflow is detailed in Figure 3.

Similar Frame Selection. For a new frame I_{t+1} , the most similar frame I_{near} (with known pose) is selected from the training set using feature similarity (extracted by VGG16). This reduces the search space for pose optimization.

Pose Estimation: Enhanced by Point Matching Loss. To tackle the challenge of large inter-frame camera motion in unordered frame sequences, the relative pose $T_{near\to i+1}$ between the selected most similar frame I_{near} (with known and reliable pose from the training set) and the new frame I_{i+1} (to be pose-estimated) is optimized through a two-stage strategy that fuses geometric matching information and photometric consistency, with the specific optimization logic and loss design as follows,

RGB loss. The RGB loss is same as the ordered frame sequences algorithm.

Point Matching loss. LoFTR [22] is used to detect corresponding points between the rendered image (from the local 3DGS of I_{near}) and I_{i+1}. The loss minimizes the Euclidean distance between matching points,

$$L_{\text{match}} = \frac{1}{N} \sum_{i=0}^{N-1} \|m_i - q_i\|^2, \tag{7}$$

where m_i and q_i are matching points from the rendered and real images, respectively.

Two-Stage optimization. Firstly, $0.05L_{\text{rgb}} + L_{match}$ is used for coarse pose estimation. When $L_{match} < 0.001$, switch to L_{rgb} only for fine-tuning.

3DGS Optimization: Scale-Invariant Depth Loss and Local Point Cloud Management. In unordered frame sequences, 3D Gaussian model optimization faces two key challenges. First, for unordered frames, the scale differences between the estimated depth maps are even greater due to large perspective differences. The scale compensation method in Eq. 4 is no longer applicable in this case. Second, the initial 3D Gaussians are sparse in new scene regions. To solve these issues, this paper adopts two strategies for optimization, a scale-invariant depth loss and a local

point cloud management strategy.

Scale-Invariant Depth loss. To handle scale difference problems in monocular depth estimates, the following log-space loss is used:

$$L_{\text{depth}} = \frac{1}{2N} \sum_{k=1}^{N} \left(\log \hat{d}_k - \log d_k + \mathcal{F}(\hat{D}_i, D_i) \right)^2 \quad (8)$$

$$\mathcal{F}(\hat{D}_i, D_i) = \frac{1}{N} \sum_{k} \left(\log d_k - \log \hat{d}_k \right) \quad (9)$$

where $\hat{d}_k$ and d_k are the rendered and estimated depths of the k^{th} pixel in $\hat{D}_i$ and D_i, respectively, and Eq. 9 is the mean of logarithmic depth differences across all pixels, used to offset global scale inconsistencies between rendered and estimated depths.

Local Point Cloud Management. The Local Point Cloud Management strategy addresses two issues in unordered sequences: (1) Initial point clouds only cover the first frame's scene, leaving new regions sparse, and (2) the original 3DGS adaptive density control overlooks error regions and fails to densify point clouds effectively. The implementation of the local point cloud management follows three steps: 1) Generate a 2D rendering error map by computing pixel-wise differences between the rendered and real images. 2) Project error regions back to 3D using conical projection (based on multi-view geometry [27]). 3) Densify 3D Gaussians in error regions (cloning for small Gaussians, splitting for large Gaussians) and prune low-opacity Gaussians.

4. Experiments

4.1. Experimental Setup

Ordered Datasets. To evaluate the performance of the proposed pose-free 3D scene reconstruction algorithm for ordered frame sequences (video frame sequences), comprehensive experiments are conducted on two benchmark datasets: Tanks and Temples [11] and CO3D-V2 [19], which are also widely used in state-of-the-art methods like CF-3DGS [7] for fair comparison. **Tanks and Temples:** A publicly available benchmark for image-based 3D reconstruction, collected in real-world indoor and outdoor environments. It provides high-resolution video sequences with diverse scene types. Eight representative scenes are selected for evaluation, including Church, Barn, Museum, Family, Horse, Ballroom, Francis, and Ignatius. For each scene, the image sequence is split into training and test sets: most scenes follow a "1-in-7" sampling strategy (every 7th frame is selected as test data, and the rest as training data), while the Family scene uses a "1-in-2" sampling strategy. To assess camera pose estimation accuracy, COLMAP [21] is employed to estimate the camera poses of training images as ground-truth; the estimated poses and ground-truth poses are aligned via Procrustes analysis [13, 2] (consistent with CF-3DGS) before error calculation. **CO3D-V2:** A large-scale dataset for 3D object reconstruction, consisting of object-centric videos where cameras rotate 360° around the target object. This dataset poses greater challenges due to larger and more complex camera motions compared to Tanks and Temples. Four object sequences are selected: 415_57112_110099, 106_12648_23157, 245_26182_52130, and 34_1403_4393. The train/test split follows the same sampling strategy as Tanks and Temples. Unlike Tanks and Temples, CO3D-V2 provides official camera poses, eliminating the need for COLMAP-based pose estimation.

Unordered Datasets. The **LLFF** Dataset [3] is selected to evaluate the proposed pose-free 3D reconstruction algorithm for unordered frame sequences. LLFF consists of 8 real-world scenes (Fern, Flower, Fortress, Horns, Leaves, Orchids, Room, Trex) captured by hand-held cameras, with non-continuous camera viewpoints (unordered frames) and varying image counts (20–62 frames per scene). Images of resolution 1008×756 are used for experiments. For each scene, 1/8 of the images are selected as test data (every 7th frame by image ID), and the remaining 7/8 as training data—consistent with the split strategy for ordered sequences.Monocular depth maps are estimated using the pre-trained DPT model [18] (same as CF-3DGS and the ordered sequence method). COLMAP [21] is used to generate ground-truth camera poses for pose evaluation.

Metrics. We conduct evaluations on two key tasks: novel view synthesis and camera pose estimation. For the task of camera pose estimation, we report errors associated with camera rotation and translation. Specifically, we adopt the evaluation metrics of Absolute Trajectory Error (ATE) and Relative Pose Error (RPE), following the protocols outlined in [13, 2]. For novel view synthesis, we utilize a set of standard evaluation metrics. These include Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) [24], and Learned Perceptual Image Patch Similarity (LPIPS) [29].

Implementation Details. The ordered reconstruction algorithm is implemented using PyTorch 1.12 with CUDA 11.8, and all experiments are conducted on a single NVIDIA RTX 3090 GPU equipped with 24GB of VRAM. The Adam optimizer [10] is employed to update the parameters of 3D Gaussians, relative camera poses, and depth alignment, with learning rates following the configurations in the original 3DGS [9] and CF-3DGS [7]—specifically, the 3D Gaussian center is initialized with a learning rate of 1.6×10^{-4} which decays to 1.6×10^{-6} during training, the 3D Gaussian scale, rotation, and opacity use learning rates of 5×10^{-3}, 1×10^{-3}, and 5×10^{-2} respectively, the relative camera pose (parameterized as a quaternion plus a translation vector) has an initial learning rate of 1×10^{-4} (adjusted to 1×10^{-3} for the 415_57112_110099 scene in the CO3D-

Table 1. **Novel view synthesis results on Tanks and Temples for Ordered Frame Sequences.**

scenes	NeRFmm			BARF			SC-NeRF			Nope-NeRF			CF-3DGS			HT-3DGS			Ours		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR↑	SSIM↑	LPIPS↓
Church	21.64	0.58	0.54	23.17	0.62	0.52	21.96	0.60	0.53	25.17	0.73	0.39	30.23	0.93	0.11	**31.34**	**0.94**	**0.08**	31.09	**0.94**	**0.08**
Barn	23.21	0.61	0.53	25.28	0.64	0.48	23.26	0.62	0.51	26.35	0.69	0.44	31.23	0.90	0.10	**34.95**	**0.97**	**0.05**	34.08	0.96	**0.05**
Museum	22.37	0.61	0.53	23.58	0.61	0.55	24.94	0.69	0.45	26.77	0.76	0.35	29.91	0.91	0.11	31.59	**0.95**	**0.08**	**31.62**	0.94	**0.08**
Family	23.04	0.58	0.56	23.04	0.61	0.56	22.60	0.63	0.51	26.01	0.74	0.41	31.27	0.94	0.07	**34.71**	**0.97**	**0.05**	33.18	0.96	0.06
Horse	23.12	0.70	0.43	24.09	0.72	0.41	25.23	0.76	0.37	27.64	0.84	0.26	33.94	0.96	0.05	**35.82**	**0.98**	**0.03**	34.12	0.97	0.04
Ballroom	20.03	0.48	0.57	20.66	0.50	0.60	22.64	0.61	0.48	25.33	0.72	0.38	32.47	0.96	0.07	34.12	**0.97**	0.04	**34.41**	**0.97**	**0.03**
Francis	25.40	0.69	0.52	25.85	0.69	0.57	26.46	0.73	0.49	29.48	0.80	0.38	32.72	0.91	0.14	**34.09**	**0.93**	**0.13**	33.08	**0.93**	**0.13**
Ignatius	21.16	0.45	0.60	21.78	0.47	0.60	23.00	0.55	0.53	23.96	0.61	0.47	28.43	0.90	0.09	31.64	**0.95**	**0.06**	**31.94**	0.94	**0.06**
mean	22.50	0.59	0.54	23.42	0.61	0.54	23.76	0.65	0.48	26.34	0.74	0.39	31.28	0.93	0.09	**33.53**	**0.96**	**0.07**	32.94	0.95	**0.07**

Table 2. **Novel view synthesis results on CO3D-V2 for Ordered Frame Sequences.**

Method	Time	415_57112_110099			106_12648_23157			245_26182_52130			34_1403_4393		
		PSNR↑	SSIM↑	LPIPS↓	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
Nope-NeRF	∼30h	24.78	0.64	0.55	20.41	0.46	0.58	25.05	0.80	0.49	28.62	0.80	0.35
CF-3DGS	∼2h	26.21	0.73	0.32	22.14	0.64	0.34	27.24	0.85	0.30	27.75	0.86	0.20
HT-3DGS	∼4h	27.23	0.78	0.30	23.43	0.73	0.28	**28.59**	**0.87**	**0.27**	**32.52**	**0.93**	**0.14**
Ours	∼3h	**28.58**	**0.85**	**0.26**	**27.05**	**0.88**	**0.17**	28.20	**0.87**	0.29	31.43	**0.93**	0.15

V2 dataset) that decays to 1×10^{-5}, and the depth alignment parameters (scale and offset) adopt a learning rate of 5×10^{-4}. The progressive training framework alternates between 3D Gaussian optimization (with 1000 iterations performed for each newly added frame) and camera pose estimation (with 500 iterations conducted per frame), Gaussian densification operations (including clone and split) are executed synchronously with the addition of new frames, and opacity reset is applied every 3000 iterations to ensure the model can adapt to the newly added scene content, which is consistent with the progressive growth strategy of CF-3DGS.

The unordered implementation builds on the ordered sequence framework. Adam optimizer [10] (β_1=0.9, β_2=0.999) is used. The learning rate for relative poses (parameterized as se(3) Lie algebra) is set to 6×10^{-3} (higher than the ordered sequence method to handle larger motion), decayed to 6×10^{-4}. The progressive framework alternates between 3D Gaussian optimization (1000 iterations per new frame) and camera pose estimation (500 iterations per frame). Training time varies by scene: 0.5 hours for Fern (fewer frames) and 1.6 hours for Horns (more frames).

4.2. Comparative Experiments on Ordered Sequences

In this subsection, we compare our proposed method with five state-of-the-art pose-unknown 3D reconstruction baselines, including two 3DGS-based method (CF-3DGS [7] and HT-3DGS [8]) and four NeRF-based methods (Nope-NeRF [2], BARF [13], NeRFmm [20], SC-NeRF [17]). Note that HT-3DGS [8] is a concurrent work that utilizes a pre-trained Video Frame Interpolation (VFI) network to generate intermediate supervision. All baselines are trained with their public code under original settings and evaluated using the same protocol for fair comparison.

Novel View Synthesis. Unlike the standard setting where test-view camera poses are pre-provided, we first estimate test-view poses for rendering (consistent with CF-3DGS [7] and NeRFmm [20]). For each test view, we freeze the pre-trained 3DGS model (trained on training views) and optimize its camera pose by minimizing the photometric error between synthesized and real test images. To accelerate convergence, the test-view pose is initialized with the closest camera position from the learned training-view poses, followed by fine-tuning with photometric loss. This procedure is applied uniformly to all baselines.

Quantitative results on Tanks and Temples are reported in Table 1. Our method achieves state-of-the-art performance among non-VFI based approaches, significantly outperforming NeRF-based baselines (e.g., +6.6 dB over Nope-NeRF [2]) and the 3DGS-based baseline CF-3DGS [7] (+1.66 dB). Compared to the concurrent work HT-3DGS [8], although it achieves a slightly higher average PSNR (33.53 vs. 32.94 dB) benefiting from intermediate supervision via Video Frame Interpolation (VFI), our method remains highly competitive. Notably, we surpass HT-3DGS in scenes like Museum and Ballroom without the additional computational overhead of training auxiliary VFI models.

On CO3D-V2 (Table 2), our method maintains superior performance. Compared to CF-3DGS, we achieve an average +2.98 dB PSNR, +0.11 SSIM, and -0.07 LPIPS. The 106_12648_23157 scene sees the most significant gains (+4.91 dB PSNR, +0.24 SSIM), demonstrating robustness to large motion, an advantage over CF-3DGS, which struggles with extreme camera trajectories due to its simplistic local Gaussian transformation. While HT-3DGS performs well on sequences with smooth motion, our method significantly outperforms it on challenging object-centric scenes with large camera movements, such as 415_57112_110099 (+1.35 dB) and 106_12648_23157 (+3.62 dB). This highlights that our depth-guided optimization strategy is more

Figure 4. **Qualitative Novel View Synthesis Results on Tanks and Temples for Ordered Frame Sequences.** Our method synthesizes higher-fidelity images (e.g., preserved window details in Museum, correct sculpture shapes in Horse) compared to baselines like Nope-NeRF (blurry textures) and CF-3DGS (artifacts and background collapse) and HT-3DGS (over-smooth high-frequency details).

Table 3. **Camera pose estimation results on Tanks and Temples for Ordered Frame Sequences**

scenes	NeRFmm			BARF			SC-NeRF			Nope-NeRF			CF-3DGS			Ours		
	RPE_t	RPE_r	ATE	RPE_t	RPE_r	ATE	RPE_t	RPE_r	ATE	RPE_t	RPE_r	ATE	RPE_t	RPE_r	ATE	$RPE_t\downarrow$	$RPE_r\downarrow$	ATE$\downarrow$
Church	0.626	0.127	0.065	0.114	0.038	0.052	0.836	0.187	0.108	0.034	0.008	0.008	0.008	0.018	0.002	**0.007**	**0.013**	**0.001**
Barn	1.629	0.494	0.159	0.314	0.265	0.050	1.317	0.429	0.157	0.046	0.032	0.004	0.034	0.034	0.003	**0.009**	**0.016**	**0.001**
Museum	4.134	1.051	0.346	3.442	1.128	0.263	8.339	1.491	0.316	0.207	0.202	0.020	0.052	0.215	0.005	**0.065**	**0.204**	**0.005**
Family	2.743	0.537	0.120	1.371	0.591	0.115	1.171	0.499	0.142	0.047	0.015	0.001	0.022	0.024	0.002	**0.033**	**0.040**	**0.002**
Horse	1.349	0.434	0.018	1.333	0.394	0.014	1.366	0.438	0.019	0.179	0.017	0.003	0.112	0.057	0.003	**0.091**	**0.053**	**0.002**
Ballroom	0.449	0.177	0.031	0.531	0.228	0.018	0.328	0.146	0.012	0.041	0.018	0.002	0.037	0.024	0.003	**0.021**	**0.015**	**0.001**
Francis	1.647	0.618	0.207	1.321	0.558	0.082	1.233	0.483	0.192	0.057	0.009	0.005	0.029	0.154	0.006	**0.021**	**0.119**	**0.005**
Ignatius	1.302	0.379	0.041	0.736	0.324	0.029	0.533	0.240	0.085	0.026	0.005	0.002	0.033	0.032	0.005	**0.023**	**0.031**	**0.004**
mean	1.735	0.477	0.123	1.046	0.441	0.078	1.890	0.489	0.129	0.080	0.038	0.006	0.041	0.069	0.004	**0.033**	**0.061**	**0.003**

effective than frame interpolation when handling sparse or irregular viewpoints. Furthermore, regarding training efficiency (Table 2), our method requires approximately 3 hours per scene, which is faster than HT-3DGS (∼4 hours) and significantly faster than Nope-NeRF (∼30 hours).

Qualitative results (Figure 4) further confirm our superiority: Nope-NeRF produces blurry images with missing details (e.g., distorted "Horse" sculpture), CF-3DGS suffers from artifacts (e.g., "needle-like" noise in Barn) and background collapse (e.g., sky misclassified as foreground in Francis), while our method synthesizes high-fidelity images with preserved fine textures (e.g., Museum window de-

Table 4. **Ablation study results on Tanks and Temples for Ordered Frame Sequences.**

Experiment Settings				Experiment Metrics			
Basic Framework	Depth Guidance	Depth Alignment	Feature Loss	PSNR↑	SSIM↑	RPE_t↓	RPE_r↓
✓				32.95	0.95	0.038	0.062
✓	✓			32.11	0.94	0.038	0.062
✓	✓	✓		32.54	0.94	0.038	0.062
✓	✓	✓	✓	32.94	0.95	0.033	0.061
CF-3DGS				31.28	0.93	0.041	0.069

tails) and correct object shapes.

Camera Pose Estimation. Learned camera poses of training views are post-processed via Procrustes analysis [13, 2] (consistent with CF-3DGS [7] and Nope-NeRF [2]) and compared with COLMAP-estimated ground-truth poses (Tanks and Temples) or official ground-truth poses (CO3D-V2). It is worth noting that the concurrent work HT-3DGS [8] is excluded from our Tanks and Temples pose evaluation (Table 3). This is because HT-3DGS strictly limits its pose estimation comparison to datasets with official ground-truth poses (e.g., CO3D-V2), explicitly avoiding reliance on COLMAP-derived pseudo-ground-truth. Quantitative results on Tanks and Temples are summarized in Table 3. Our method achieves comparable performance to state-of-the-art baselines, with average reductions of 0.008 in RPE_t (19.51% lower) and 0.001 in ATE (25% lower) compared to CF-3DGS. While our RPE_r (0.061) is slightly higher than Nope-NeRF (0.038), this is attributed to Nope-NeRF's additional point cloud constraints (not adopted in our method to avoid complexity). On CO3D-V2 (results reflected in novel view synthesis gains), our method outperforms CF-3DGS and Nope-NeRF by a large margin in pose accuracy, as evidenced by the significant improvement in rendering quality for scenes with large camera motion.

Table 5. **Ablation study results for multi-view consistent depth guidance on Tanks and Temples.**

Scene	Basic Framework		Basic Framework + Depth Guidance + Depth Alignment	
	PSNR↑	SSIM↑	PSNR	SSIM
Church	30.82	0.94	30.81	0.94
Barn	33.19	0.93	31.94	0.92
Museum	31.41	0.94	31.45	0.94
Family	33.53	0.96	33.04	0.96
Horse	34.96	0.97	34.01	0.97
Ballroom	34.35	0.97	34.56	0.97
Francis	33.36	0.93	32.91	0.92
Ignatius	31.80	0.94	31.79	0.94

4.3. Ablation Experiments on Ordered Algorithm

Ablation experiments are conducted on the Tanks and Temples dataset to validate the effectiveness of key components in the proposed method, including the **progressive training framework**, **multi-view consistent depth guidance**, **depth alignment**, and **feature loss**, with CF-3DGS [7] included as a baseline for reference and results shown in Table 4.

The proposed framework maintains a single global 3DGS model and estimates relative poses directly using pre-optimized global Gaussians—distinct from CF-3DGS, which constructs independent local 3DGS models for pose estimation. Compared to CF-3DGS, it achieves +1.67 dB in PSNR (32.95 vs.31.28), +0.02 in SSIM (0.95 vs.0.93), -0.003 in RPE_t (0.038 vs. 0.041), and -0.007 in RPE_r (0.062 vs. 0.069), confirming that the global 3DGS provides more comprehensive scene geometry to improve both reconstruction quality and pose accuracy.

Introducing multi-view consistent depth guidance (with depth alignment) slightly reduces PSNR by 0.41dB (32.54 vs.32.95) but eliminates artifacts and background collapse. Quantitative results per scene (Table 5) show negligible quality differences in most scenes, indicating the trade-off between texture fidelity (slightly reduced) and geometry accuracy (significantly improved) is acceptable—this addresses the artifact and background collapse issues that exist in CF-3DGS (which does not adopt depth guidance).

Depth alignment plays a critical role in resolving the multi-view inconsistency of monocular depth estimates. Without depth alignment, PSNR drops by 0.43 dB (32.11 vs.32.54), highlighting that aligning depth scales across views is essential for effective depth guidance—an issue unaddressed in CF-3DGS. Adding feature loss(extracted via a pre-trained encoder) further enhances performance: PSNR increases by 0.40 dB (32.94 vs.32.54), RPE_t decreases by 0.005 (0.033 vs.0.038), and RPE_r decreases by 0.001 (0.061 vs.0.062). Qualitative results show feature loss reduces visual noise(e.g., blurriness in the Barn scene), confirming its ability to enhance pose estimation robustness—an improvement over CF-3DGS, which relies solely on RGB loss for pose optimization and is more susceptible to visual noise.

4.4. Comparative Experiments on Unordered Sequences

In this subsection, we compare our proposed method with four state-of-the-art baselines, including one pose-unknown method (CF-3DGS [7], the only 3DGS-based pose-free method, designed primarily for ordered sequences and less robust to unordered frames) and three pose-known methods (3DGS [9], BARF [13], NeRF [16]) to establish performance upper bounds. All baselines are evaluated under the same protocol using their official implementations with default hyperparameters. Note on HT-3DGS [8]: It is worth noting that while HT-3DGS shows strong perfor-

Table 6. **Novel view synthesis results on LLFF for Unordered Frame Sequences.**

Scene	With Camera Poses									Without Camera Poses					
	NeRF			BARF			3DGS			CF-3DGS			Ours		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR↑	SSIM↑	LPIPS↓
Fern	23.72	0.733	0.262	23.79	0.710	0.311	23.63	0.794	0.136	17.35	0.494	0.428	**21.87**	**0.700**	**0.232**
Flower	23.24	0.668	0.244	23.37	0.698	0.211	26.91	0.829	0.096	20.17	0.622	0.362	**27.99**	**0.858**	**0.143**
Fortress	25.97	0.786	0.185	29.08	0.823	0.132	29.93	0.880	0.078	14.73	0.395	0.460	**21.93**	**0.564**	**0.312**
Horns	20.35	0.624	0.421	22.78	0.727	0.298	25.02	0.862	0.121	15.60	0.412	0.514	**25.08**	**0.810**	**0.182**
Leaves	15.33	0.306	0.526	18.78	0.537	0.353	17.91	0.593	0.205	15.38	0.416	0.398	**18.27**	**0.568**	**0.314**
Orchids	17.34	0.518	0.307	19.45	0.574	0.291	18.98	0.612	0.159	13.80	0.258	0.516	**16.51**	**0.481**	**0.348**
Room	32.42	0.948	0.080	31.95	0.940	0.099	28.96	0.927	0.115	18.36	0.713	0.382	**25.62**	**0.860**	**0.182**
Trex	22.12	0.739	0.244	22.55	0.767	0.206	24.74	0.881	0.145	16.76	0.522	0.434	**22.33**	**0.770**	**0.264**
Mean	22.56	0.665	0.284	23.97	0.722	0.238	24.51	0.797	0.132	16.51	0.479	0.437	**22.45**	**0.701**	**0.247**

Table 7. **Camera pose estimation results on LLFF for Unordered Frame Sequences.**

Scene	CF-3DGS			Ours		
	RPE_t	RPE_r	ATE	$RPE_t \downarrow$	$RPE_r \downarrow$	$ATE \downarrow$
Fern	7.458	2.590	0.130	**0.370**	**0.115**	**0.005**
Flower	1.885	0.197	0.037	**0.372**	**0.565**	**0.006**
Fortress	7.944	1.070	0.138	**7.262**	**1.959**	**0.044**
Horns	2.526	1.021	0.082	**0.438**	**0.498**	**0.006**
Leaves	15.105	1.028	0.153	**22.072**	**1.227**	**0.036**
Orchids	3.360	1.705	0.074	**1.428**	**1.034**	**0.008**
Room	3.596	1.447	0.102	**1.829**	**1.319**	**0.037**
Trex	4.438	1.802	0.110	**1.942**	**1.166**	**0.024**
Mean	5.789	1.358	0.103	**4.464**	**0.985**	**0.020**

mance on ordered videos, it is excluded from the unordered comparisons in this section. Its core components—Video Frame Interpolation and hierarchical merging—strictly rely on temporal continuity, making the method inapplicable to unordered collections like LLFF where large viewpoint jumps occur. This highlights a significant advantage of our framework: it provides a unified solution that achieves SOTA performance on both continuous videos and challenging unordered image sets.

Experimental Protocol. For novel view synthesis, test-view camera poses are estimated following a unified procedure: we freeze the pre-trained 3DGS model (trained on unordered training frames) and optimize each test-view pose by minimizing a combined loss of RGB photometric error and feature matching loss (via LoFTR [22]). To ensure fair comparison, test poses for all baselines are initialized using the closest training-view pose (selected via VGG feature similarity) before fine-tuning. For camera pose evaluation, estimated training-view poses are aligned with COLMAP-derived ground-truth via Procrustes analysis [13, 2], with errors computed using ATE, RPE_t, and RPE_r.

Quantitative results on the LLFF dataset are summarized in Table 6 and Table 7. Compared to the pose-unknown baseline CF-3DGS, our method achieves substantial improvements across all metrics: +5.94 dB in average PSNR (22.45 vs. 16.51), +0.222 in SSIM (0.701 vs. 0.479), and -0.19 in LPIPS (0.247 vs. 0.437). Pose estimation errors are also significantly reduced: RPE_t by 1.325 (22.89% lower), RPE_r by 0.373 (27.47% lower), and ATE by 0.083 (80.58% lower), confirming our method's robustness to unordered frames—a critical limitation of CF-3DGS, which relies on temporal continuity.

Against pose-known methods, our approach achieves competitive performance: its average PSNR (22.45 dB) is comparable to NeRF (22.56 dB) and BARF (23.97 dB), with SSIM (0.701) outperforming NeRF (0.665). In specific scenes like Flower and Horns, our method even matches 3DGS [9] (e.g., Flower: 27.99 vs. 26.91 PSNR), demonstrating that pose-free 3DGS can approach the performance of pose-known methods with proper handling of unordered frames.

Qualitative comparisons (Figure 5) further validate our advantages. CF-3DGS produces severely distorted images with missing content (e.g., blurry leaves in Fern and malformed Trex), as it fails to model unordered frame relationships. BARF (pose-known) generates clearer results but lacks fine details (e.g., Flower petal textures). Our method synthesizes high-fidelity images with preserved details like leaf veins in Horns and wall textures in Fortress, confirming its ability to leverage unordered frames effectively through matching loss and local point cloud management—key improvements over both CF-3DGS and pose-known baselines.

4.5. Ablation Experiments on Unordered Algorithm

Ablation experiments on the LLFF dataset validate four key components of the proposed method: **progressive training framework**, **matching loss**, **scale-invariant depth loss**, and **local point cloud management**, with results shown in Table 8.

The proposed framework, adapted for unordered frames, outperforms CF-3DGS with +4.56 dB in PSNR (21.08 vs.16.52), +0.17 in SSIM (0.65 vs.0.48), and -0.054 in ATE (0.049 vs.0.103). Though RPE_r increases (1.913 vs.1.358), this is resolved by subsequent modules, confirming the framework's adaptability to unordered frames—an advantage over CF-3DGS, which struggles with non-sequential viewpoints.

Adding matching loss (via LoFTR [22]) significantly improves performance: +0.99 dB in PSNR (22.07 vs.21.08),

Table 8. **Ablation study results on LLFF for Unordered Frame Sequences.**

Experiment Settings				Experiment Metrics			
Basic Framework	Matching Loss	Depth Loss	Local Point Cloud Management Strategy	PSNR↑	SSIM↑	RPE_r↓	ATE↓
✓				21.08	0.65	1.913	0.049
✓	✓			22.07	0.69	0.994	0.019
✓	✓	✓		22.05	0.69	1.168	0.022
✓	✓	✓	✓	22.45	0.70	0.985	0.020
CF-3DGS				16.52	0.48	1.358	0.103

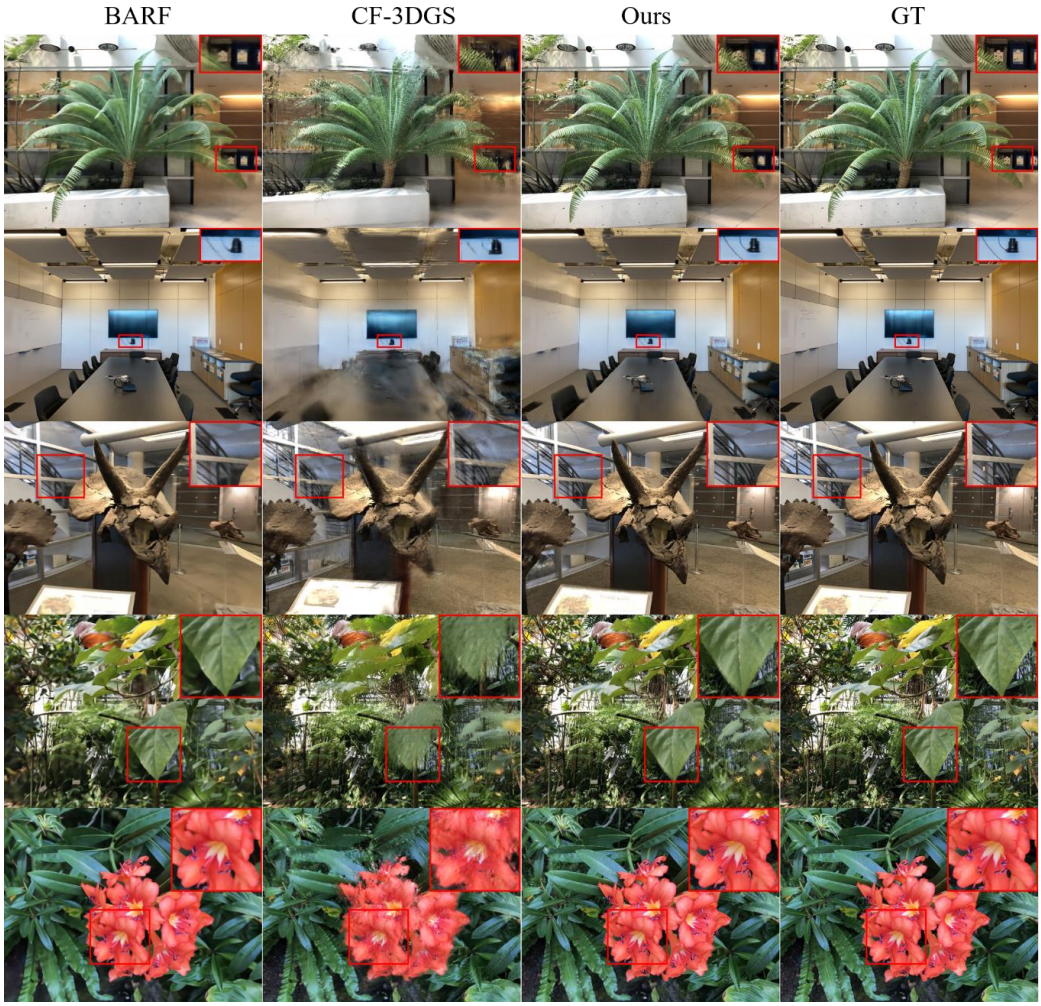

Figure 5. **Qualitative Novel View Synthesis Results on LLFF for Unordered Frame Sequences.** Our method preserves fine details (e.g., leaf veins in Horns, wall textures in Fortress) compared to CF-3DGS (severe distortions) and BARF (pose-known, missing textures), demonstrating robustness to unordered frames.

-0.919 in RPE_r (0.994 vs.1.913), and -0.03 in ATE (0.019 vs.0.049). As shown in our results, it reduces blurriness (e.g., in the Orchids scene) by aligning geometric features for pose optimization—addressing the pose estimation instability in the baseline framework without matching constraints.

Introducing scale-invariant depth loss has minimal impact on PSNR/SSIM but improves depth map accuracy. For example, in the Fortress scene, the proposed method correctly models the depth of distant walls, unlike the ablation variant without depth loss, which confuses foreground and background—this geometric consistency is absent in CF-3DGS, which lacks depth guidance.

Finally, adding local point cloud management further enhances metrics: +0.40 dB in PSNR (22.45 vs.22.05), +0.01 in SSIM (0.70 vs.0.69), and -0.183 in RPE_r (0.985 vs.1.168). This confirms that targeted Gaussian densification/pruning in error regions enhances geometry model-

ing for unordered frames—an improvement over both CF-3DGS and the ablation variant without this strategy, which suffer from under-modeled local details.

5. Conclusion

This paper focuses on pose-free 3D scene reconstruction based on 3DGS to reduce reliance on SfM for camera poses. For ordered frame sequences, we proposes a progressive algorithm with a camera motion-sensitive feature encoder and multi-view consistent depth loss. For unordered sequences, we designs a strategy integrating matching loss for pose estimation, scale-invariant depth loss, and local point cloud management. Experiments on Tanks and Temples, CO3D-V2, and LLFF datasets show the proposed methods outperform baseline methods in novel view synthesis and camera pose estimation. The thesis also notes limitations of the feature encoder's generalization and computational overhead in large-scale scenes, and points out future optimization directions.

Acknowledgement

The work was supported by the Fundamental Research Funds for the Central Universities under grant No. 2024ZYGXZR021, and Guangdong Provincial Basic and Applied Basic Research Fund under Grant No. 2025A1515011884.

References

[1] S. F. Bhat, R. Birkl, D. Wofk, P. Wonka, and M. Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023. 4

[2] W. Bian, Z. Wang, K. Li, J.-W. Bian, and V. A. Prisacariu. Nope-nerf: Optimising neural radiance field with no pose prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4160–4169, 2022. 2, 6, 7, 9, 10

[3] M. Broxton, J. Flynn, R. Overbeck, D. Erickson, P. Hedman, M. DuVall, J. Dourgarian, J. Busch, P. Whalen, and P. Debevec. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 39(4):1–14, 2020. 2, 6

[4] S. Chen, X. Li, Z. Wang, and V. A. Prisacariu. Dfnet: Enhance absolute pose regression with direct feature matching. In *European Conference on Computer Vision*, pages 1–17. Springer, 2022. 4

[5] Z. Fan, W. Cong, K. Wen, K. Wang, J. Zhang, X. Ding, D. Xu, B. Ivanovic, M. Pavone, G. Pavlakos, et al. Instantsplat: Unbounded sparse-view pose-free gaussian splatting in 40 seconds. *arXiv preprint arXiv:2403.20309*, 2(3):4, 2024. 2

[6] G. Fang and B. Wang. Mini-splatting: Representing scenes with a constrained number of gaussians. In *European Conference on Computer Vision*, pages 165–181. Springer, 2024. 2

[7] Y. Fu, S. Liu, A. Kulkarni, J. Kautz, A. A. Efros, and X. Wang. Colmap-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20796–20805, 2024. 2, 6, 7, 9

[8] B. Ji and A. Yao. Sfm-free 3d gaussian splatting via hierarchical training. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21654–21663, 2025. 2, 7, 9

[9] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering., 2023. 1, 6, 9, 10

[10] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 4, 6, 7

[11] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017. 2, 6

[12] J. C. Lee, D. Rho, X. Sun, J. H. Ko, and E. Park. Compact 3d gaussian representation for radiance field, 2024. 2

[13] C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey. Barf: Bundle-adjusting neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2, 6, 7, 9, 10

[14] Y. Liu, C. Luo, L. Fan, N. Wang, J. Peng, and Z. Zhang. Citygaussian: Real-time high-quality large-scale scene rendering with gaussians. In *European Conference on Computer Vision*, pages 265–282. Springer, 2024. 2

[15] A. Meuleman, I. Shah, A. Lanvin, B. Kerbl, and G. Drettakis. On-the-fly reconstruction for large-scale novel view synthesis from unposed images. *ACM Transactions on Graphics (TOG)*, 2025. 3

[16] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis, 2020. 1, 9

[17] J.-I. Pan, C.-H. Lai, Y.-H. Chuang, Y.-Y. Wang, J.-H. Huang, W.-C. Chiu, and M. Sun. Self-calibrating neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5846–5854, 2021. 2, 7

[18] R. Ranftl, A. Bochkovskiy, and V. Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021. 3, 4, 6

[19] J. Reizenstein, R. Shapovalov, P. Henzler, L. Sbordone, P. Labatut, and D. Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10901–10911, 2021. 2, 6

[20] Z. Rosenthal, J. Vertens, M. Wray, R. Liao, A. Zaganidis, L. Liu, and W. Burgard. Nerf–: Neural radiance fields without known camera parameters. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8487–8494, 2022. 2, 7

[21] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Com-*

puter Vision and Pattern Recognition (CVPR), pages 4104–4113, 2016. 6

[22] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8922–8931, 2021. 5, 10

[23] J. Wang, M. Chen, C. Rupprecht, N. Karaev, A. Vedaldi, and D. Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 2, 3

[24] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 6

[25] Y. Yan, H. Lin, C. Zhou, W. Wang, H. Sun, K. Zhan, X. Lang, X. Zhou, and S. Peng. Street gaussians: Modeling dynamic urban scenes with gaussian splatting. In *European Conference on Computer Vision*, pages 156–173. Springer, 2024. 2

[26] Z. Yan, W. F. Low, Y. Chen, and G. H. Lee. Multi-scale 3d gaussian splatting for anti-aliased rendering, 2024. 2

[27] H. Yang, C. Zhang, W. Wang, M. Volino, A. Hilton, L. Zhang, and X. Zhu. Gaussian splatting with localized points management. *CoRR*, 2024. 6

[28] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi. Soundstream: An end-to-end neural audio codec, 2021. 2

[29] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 6