# TrackGS: Optimizing COLMAP-Free 3D Gaussian Splatting with Global Track Constraints

**Dongbo Shi**[1*], **Shen Cao**[2*], **Lubin Fan**[2✉], **Bojian Wu**[2], **Jinhui Guo**[2], **Ligang Liu**[1], **Renjie Chen**[1✉],

[1]University of Science and Technology of China
[2]Independent Researcher
✉lubinfan@gmail.com, renjiec@ustc.edu.cn

### Abstract

We present TrackGS, a novel method to integrate global feature tracks with 3D Gaussian Splatting (3DGS) for COLMAP-free novel view synthesis. While 3DGS delivers impressive rendering quality, its reliance on accurate precomputed camera parameters remains a significant limitation. Existing COLMAP-free approaches depend on local constraints that fail in complex scenarios. Our key innovation lies in leveraging feature tracks to establish global geometric constraints, enabling simultaneous optimization of camera parameters and 3D Gaussians. Specifically, we: (1) introduce track-constrained Gaussians that serve as geometric anchors, (2) propose novel 2D and 3D track losses to enforce multi-view consistency, and (3) derive differentiable formulations for camera intrinsics optimization. Extensive experiments on challenging real-world and synthetic datasets demonstrate state-of-the-art performance, with much lower pose error than previous methods while maintaining superior rendering quality. Our approach eliminates the need for COLMAP preprocessing, making 3DGS more accessible for practical applications.

## Introduction

Given a collection of images from a 3D scene along with the corresponding camera intrinsic and extrinsic parameters, 3D Gaussian Splatting (3DGS) (Kerbl et al. 2023) can effectively represent the scene with a series of 3D Gaussians, and generate high-quality images from novel viewpoints. Due to its efficiency in training and superior performance in testing, 3DGS has become popular for a variety of applications including reconstruction, editing, and AR/VR etc. However, the effectiveness of 3DGS training relies on accurately pre-determined camera poses (i.e., camera extrinsics) and camera focal lengths (i.e., camera intrinsics). These parameters are typically derived using COLMAP (Schonberger and Frahm 2016) in advance. This preprocessing step is not only time-consuming but also impacts the training performance of 3DGS, particularly when dealing with complex camera movements and scenes.

Recent COLMAP-Free approaches (Yen-Chen et al. 2021; Jeong et al. 2021; Bian et al. 2023; Mai et al. 2024; Fan et al. 2024; Fu et al. 2024; Ji and Yao 2024) have tried to eliminate COLMAP dependencies by adding local constraints, such as photometric consistency or sequential frame alignment. While these methods work well for simple scenes with smooth camera trajectories, they struggle with more challenging cases—such as wide-baseline views, rapid camera movements, or unordered image collections—due to their reliance on incremental optimization and local geometric cues. This often leads to pose drift, scale ambiguity, and degraded rendering quality.

To address this problem, we propose **TrackGS**, a novel framework that integrates global track information into 3DGS to enforce multi-view geometric consistency robustly. Our key insight is that feature tracks provide strong, long-range constraints across disparate views, enabling stable optimization of both camera parameters and 3D Gaussians even in complex scenarios. Specifically, we introduce: 1. **Track Gaussians**, where a subset of Gaussians are anchored to 3D track points, ensuring their spatial accuracy through reprojection and backprojection losses. 2. **A fully differentiable pipeline**, that jointly optimizes camera intrinsics, extrinsics, and 3DGS parameters. We derive gradients for intrinsic parameters (e.g., focal length), enabling end-to-end training without any precomputed camera priors. Experimental results on both public and synthetic datasets demonstrate that our joint optimization framework with global track constraints outperforms previous methods.

In summary, our contributions are as follows:

- We propose a method to integrate track information with 3DGS, using global geometric constraints to simultaneously optimize camera parameters and 3DGS. To achieve this, we introduce 2D and 3D track losses to constrain reprojection and backprojection errors.
- We propose a joint optimization framework for all camera parameters and 3D Gaussians. Without relying on any known camera parameters, we achieve full differentiability for the entire pipeline, seamlessly integrating camera parameters estimation, including both intrinsics and extrinsics, with 3DGS training.
- On both challenging public and synthetic datasets, our approach outperforms previous methods on both camera parameters estimation and novel view synthesis.

[*]These authors contributed equally.

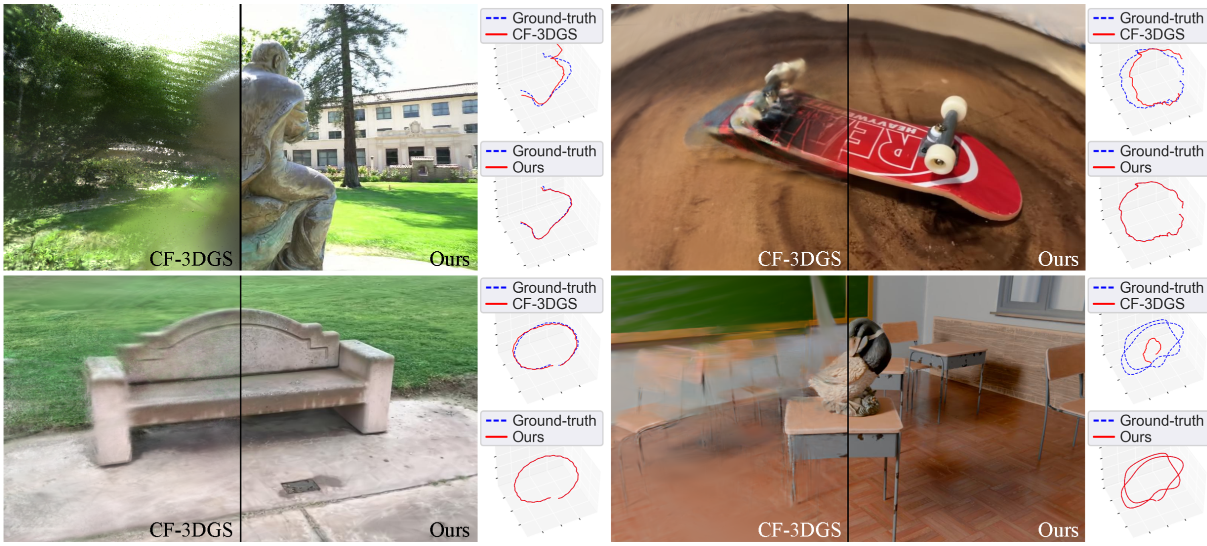


Figure 1: Comparisons on novel view synthesis and camera poses. We propose a novel 3DGS model without any known camera parameters by leveraging global track information. Compared with state-of-the-art methods, we provide higher rendering quality in novel view synthesis, and more accurate estimation of camera poses on benchmark datasets, including the challenging real-world indoor and outdoor or synthetic scenes with *complicated* camera movements (the right column).

# Related Work

## Novel View Synthesis

Novel view synthesis is a foundational task in the computer vision and graphics, which aims to generate unseen views of a scene from a given set of images. Numerous methods have been developed to address this problem by approaching it as 3D geometry-based rendering, such as using meshes (Hu et al. 2021; Riegler and Koltun 2020, 2021), MPI (Li et al. 2021; Tucker and Snavely 2020; Zhou et al. 2018), point clouds (Zhang et al. 2022; Xu et al. 2022), etc.

Recently, Neural Radiance Fields (NeRF) (Mildenhall et al. 2020) provide a novel solution to this problem by representing scenes as implicit radiance fields using neural networks, achieving photo-realistic rendering quality. Although having some works in improving efficiency (Müller et al. 2022; Lin et al. 2022), the time-consuming training and rendering still limit its practicality. Alternatively, 3D Gaussian Splatting (3DGS) (Kerbl et al. 2023) models the scene as explicit Gaussian kernels, with differentiable splatting for rendering. Its improved real-time rendering performance, lower storage and efficiency, quickly attract more attentions.

## Optimizing Camera Poses in NeRFs and 3DGS

Although NeRF and 3DGS can provide impressive scene representation, these methods all need accurate camera parameters (both intrinsic and extrinsic) as additional inputs, which are mostly obtained by COLMAP (Schonberger and Frahm 2016). When the prior is inaccurate or unknown, accurately estimating camera parameters and scene representations becomes crucial.

In earlier studies, scene training and camera pose estimation relied solely on photometric constraints. iNeRF (Yen-Chen et al. 2021) refines the camera poses using a pre-trained NeRF model. NeRFmm (Wang et al. 2021) introduces a joint optimization approach that simultaneously estimates camera poses and trains the NeRF model. BARF (Lin et al. 2021) and GARF (Chng et al. 2022) propose a new positional encoding strategy to address the gradient inconsistency issues in positional embedding, achieving promising results. However, these methods only yield satisfactory optimization when the initial pose is very close to the ground truth, as photometric constraints alone can only enhance camera estimation quality within a limited range. Subsequently, SC-NeRF (Jeong et al. 2021) minimizes a projected ray distance loss based on correspondence between adjacent frames. NoPe-NeRF (Bian et al. 2023) utilizes monocular depth maps as geometric priors and defines undistorted depth loss and relative pose constraints.

Regarding 3D Gaussian Splatting, CF-3DGS (Fu et al. 2024) utilizes mono-depth information to refine the optimization of local 3DGS for relative pose estimation and subsequently learns a global 3DGS in a sequential manner. InstantSplat (Fan et al. 2024) targets sparse view scenes, initially employing MASt3R (Leroy, Cabon, and Revaud 2024) to create a dense, pixel-aligned point set for initializing 3D Gaussian models and implements a parallel grid partitioning strategy to accelerate joint optimization. Jiang et al. (Jiang et al. 2024) develops an incremental method for reconstructing camera poses and scenes, but struggles with complex scenes and unordered images. SFGS (Ji and Yao 2024) interpolates frames for training and splits the scene into local clips, using a hierarchical strategy to build 3DGS model. It works well for simple scenes, but fails with dramatic motions due to unstable interpolation and low efficiency.

However, most existing methods generally depend on sequentially ordered image inputs and incrementally optimize camera parameters and 3DGS, which often leads to drift er-

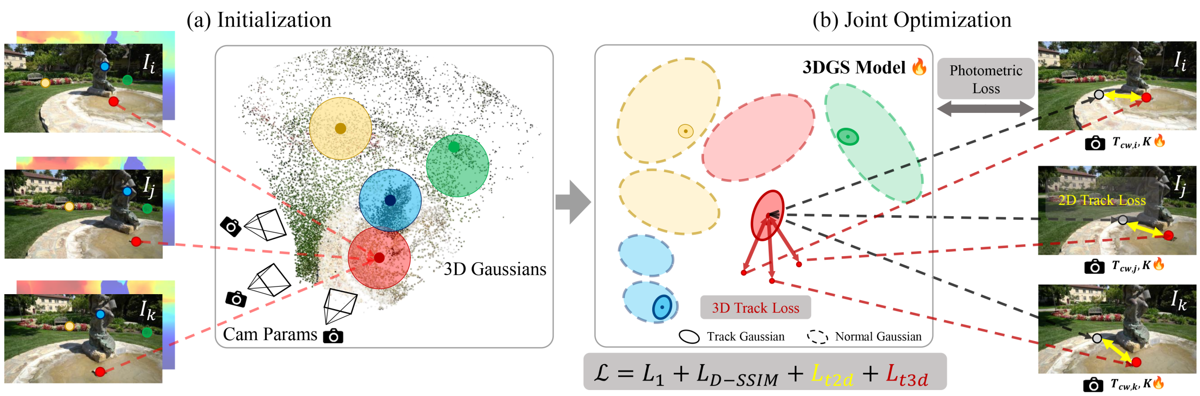


Figure 2: Overview. Given a set of images, our method obtains both camera intrinsics and extrinsics, as well as a 3DGS model. During the initialization, we extract the global tracks, and initialize camera parameters and Gaussians from image correspondences and monodepth. We determine Gaussian kernels with 3D track points, and then jointly optimize the parameters $K$, $T_{cw}$, 3DGS through the proposed global track constraints ($L_{t2d}$, $L_{t3d}$) and original photometric losses ($L_1$, $L_{D-SSIM}$).

rors and hinders achieving globally consistent results. Our work seeks to overcome these limitations.

## 3D Gaussian Splatting

3DGS models a scene using a set of 3D anisotropic Gaussians. Each Gaussian is parameterized by a centroid $\mu \in \mathbb{R}^3$, a quaternion factor $q \in \mathbb{R}^4$, a scale factor $s \in \mathbb{R}^3$, spherical harmonics (SH) coefficients of color $c \in \mathbb{R}^k$, and opacity $\alpha \in \mathbb{R}$. Donating the rotation matrix of quaternion $q$ and scale matrix of $s$ by $R \in \mathbb{R}^{3\times3}$ and $S = \text{diag}(s)$, the covariance matrix $\Sigma$ and Gaussian function $G(x)$ are:

$$\Sigma = RSS^\top R^\top, G(x) = \exp\left(-\frac{1}{2}(x-\mu)^\top\Sigma^{-1}(x-\mu)\right). \tag{1}$$

Denoting projection matrix $T_{cw} = [R_{cw}|t_{cw}]$, which transforms points from the *world* to *camera* coordinate space, an image rendered from the specified view can be obtained as follows. First, the covariance matrix in camera coordinates $\Sigma^{\text{2D}}$ is obtained by approximating the projection of 3D Gaussian in pixel coordinates, and can be expressed as:

$$\Sigma^{\text{2D}} = JR_{cw}\Sigma R_{cw}^\top J^\top, \tag{2}$$

where $J$ is the Jacobian of the affine approximation of the projective transformation. The final rendered color $\hat{C}$ can be denoted as the alpha-blending of $N$ ordered Gaussians:

$$\hat{C} = \sum_i^N c_i\alpha_i \prod_j^{i-1}(1-\alpha_j), \tag{3}$$

where $c_i$ and $\alpha_i$ are the color and opacity of the Gaussians. Similarly, the depth of the scene perceived of a pixel is,

$$\hat{D} = \sum_i^N d_i\alpha_i \prod_j^{i-1}(1-\alpha_j), \tag{4}$$

where $d_i$ denotes the z-axis coordinate for the transformed Gaussian centers in the camera space.

Usually, the parameters of 3D Gaussians are optimized by rendering and comparing the rendered images with the ground-truths. The loss function $\mathcal{L}$ is defined as follows:

$$\mathcal{L} = (1-\lambda)L_1 + \lambda L_{\text{D-SSIM}}. \tag{5}$$

Typically, 3D Gaussians are initialized with Structure from Motion (SfM) point clouds obtained from the input images.

## Method

**Overview.** Given a set of images $\mathcal{I} = \{I_i\}_{i=1}^M$, with unknown extrinsic matrix $T_{cw,i}$ at each view and unknown intrinsic matrix denoted by $K$, our method aims to build a 3D Gaussian Splatting (3DGS) model while simultaneously estimating both the extrinsic and intrinsic matrices, as shown in Fig. 2. To achieve this goal, our key approach is to leverage the global track constraint to explicitly capture and enforce multi-view geometric consistency, which serves as the foundation for accurately estimating both the 3DGS model and the camera parameters. During initialization, we extract 2D feature correspondences and derive global tracks using off-the-shelf methods. We then initialize both the camera parameters and the original 3D Gaussians with the estimated 3D track points. Subsequently, we extract a subset of Gaussian kernels from the vanilla model, referred to as *track Gaussians*, to integrate the track-based geometric prior into 3DGS. Building on this, we further propose a joint optimization framework with two additional loss terms: a 2D track loss and a 3D track loss. The 2D track loss enforces multi-view geometric consistency by minimizing the error between the reprojected track pixels and their references. The 3D track loss constrains the track Gaussians to remain aligned with the scene surface by penalizing the distance between the back-projected 3D points (from rendered depth) and the track Gaussians. We derive and implement the differentiable components of the camera parameters, including both the extrinsic and intrinsic matrices. This allows us to apply the chain rule, enabling seamless joint optimization of the 3DGS model and the camera parameters.

## Initialization

Since the training set contains only images, we need to extract essential information during initialization. To initialize the 3D Gaussians, we estimate monocular depth maps $D$ for each image $I$ using depth estimators such as DPT (Ranftl, Bochkovskiy, and Koltun 2021), following the setups in CF-3DGS (Fu et al. 2024) and SFGS (Ji and Yao 2024). In addition, to construct global track constraints for later optimization, we detect 2D feature points $\{p_i\}$ in each image $I$ and compute feature correspondences across all images using off-the-shelf methods (DeTone, Malisiewicz, and Rabinovich 2018; Sarlin et al. 2020).

**Camera Parameters.** We assume all cameras share a standard pinhole model with no distortion, and the principal point locates at the center of the image, then the intrinsic matrix $K$ of camera is:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{6}$$

where $(c_x, c_y)$ is the principal point and $(f_x, f_y)$ is focal length. Empirically, we initialize the focal length with a field of view (FoV) of $60^\circ$ as:

$$f_x = f_y = \frac{\sqrt{c_x^2 + c_y^2}}{\tan(\mathrm{FoV}/2)}. \tag{7}$$

Given the estimated mono-depth maps $D_i, D_j$ and 2D correspondences $p_i, p_j$, the relative transformation $T_{ij}$ between frames $i$ and $j$ is computed by aligning point clouds $p_i^*, p_j^*$, where $p^* = D(p) \cdot K^{-1} \cdot p$. These alignments provide a coarse initialization of both camera intrinsics and extrinsics.

**Global Tracks.** By using Union-Find algorithm over feature points, We extract a set of tracks $\mathcal{P}$, where each element $(P, \{p_i\}_{i=1}^l) \in \mathcal{P}$ represents a 3D track point $P$ and its corresponding matching points $\{p_i\}_{i=1}^l$ associated with the training images. With the estimated camera parameters and mono-depth maps, the 3D point $P$ is initialized as the average position of $\{p_i^*\}_i$. Notably, we use track points solely to initialize 3D Gaussians, as their positions will be refined by global optimization and constraints to accurately represent object surfaces.

## Joint Optimization

**Track Gaussians and Global Track Constraints.** During joint optimization, we leverage the global tracks obtained during initialization to enforce multi-view geometric consistency in both 2D and 3D space. Our method is built on two key components. First, to associate the tracks with the 3DGS model, we select a subset of 3D Gaussians associated with each track, referred to as *track Gaussians*, whose centroids coincide with the 3D track points and serve as geometric anchors within the model. Second, we introduce two additional loss terms to impose global track constraints over these track Gaussians. The reprojection loss (2D track loss) enforces that the reprojections of 3D track points on each image closely aligned with the corresponding 2D feature points. The backprojection loss (3D track loss) enforces that the matched 2D feature points, when back-projected using the 3DGS rendering depth, remain close to the same corresponding 3D track points across different views. Beyond enforcing multi-view consistency, the backprojection loss further ensures that the track Gaussians treated as geometric anchors, as the projected feature points with rendering depth are considered to lie on the scene surface. These points serve as key elements for optimizing camera parameters and enhancing the global geometric consistency.

**2D Track Loss.** We reproject the 3D track point $P$ into the corresponding images using the associated camera parameters and compute the reprojection loss, which will be summed to calculate the total 2D track loss:

$$L_{t2d} = \sum_{P \in \mathcal{P}} \frac{1}{l} \sum_{i=1}^{l} \|p_i - K \cdot T_{cw,i} \cdot P\|. \tag{8}$$

**3D Track Loss.** We backproject the 2D feature points into 3D scene using the rendered depth and camera parameters associated with each point. The backprojection error is then computed with respect to the 3D track point $P$. Then errors are aggregated to calculate the overall 3D track loss:

$$L_{t3d} = \sum_{P \in \mathcal{P}} \frac{1}{l} \sum_{i=1}^{l} \|d(p_i) \cdot T_{cw,i}^{-1} \cdot K^{-1} \cdot p_i - P\|, \tag{9}$$

where $d(p_i)$ denotes the depth perceived from $p_i$ according to Eq. 4. Note, the 2D track loss relates to the track Gaussians that are mainly used for the optimization of camera parameters, whereas the 3D track loss requires the 3DGS rendered depth values during computation. This indirectly ties the optimization of the 3D track loss to the optimization of the 3DGS model and enhances the capability of multi-view geometric consistency. They are fundamentally different.

**Overall Objectives.** Combined with Eq. 5, our joint optimization can be formulated as:

$$\mathcal{L} = (1-\lambda)L_1 + \lambda L_{\mathrm{D-SSIM}} + \lambda_{t2d} L_{t2d} + \lambda_{t3d} L_{t3d}. \tag{10}$$

**Optimizing Camera Parameters.** To optimize the camera parameters of 3D Gaussians simultaneously, the gradient of the loss function $\mathcal{L}$ with respect to the camera parameters are needed. We derive these gradients accordingly, where the gradient of extrinsic parameters is:

$$\frac{\partial \mathcal{L}}{\partial T_{cw}} = \frac{\partial \mathcal{L}}{\partial t} q^\top, \tag{11}$$

where $q = [\mu, 1]^T$ and $t = T_{cw} q = [t_x, t_y, t_z, t_w]^T$. Further, let $(\mu^{'}, \Sigma^{'})$ be the 2D projection of the centroid and covariance $(\mu, \Sigma)$, the gradient of $\mathcal{L}$ respect to focal length $F = (f_x, f_y)$ can be computed, where $T = J R_{cw}$, as:

$$\begin{cases} \dfrac{\partial \mathcal{L}}{\partial f_x} = \dfrac{t_x}{t_z} \dfrac{\partial \mathcal{L}}{\partial \mu_x^{'}} + < \dfrac{\partial \mathcal{L}}{\partial T} R_{cw}^\top, \dfrac{\partial J}{\partial f_x} >, \\ \dfrac{\partial \mathcal{L}}{\partial f_y} = \dfrac{t_y}{t_z} \dfrac{\partial \mathcal{L}}{\partial \mu_y^{'}} + < \dfrac{\partial \mathcal{L}}{\partial T} R_{cw}^\top, \dfrac{\partial J}{\partial f_y} > . \end{cases} \tag{12}$$

Please refer to the supplementary materials for more details.


Figure 3: Qualitative comparison for NVS on Tanks and Temples. We achieve better rendering results on details.

| T&T | Church | | | Horse | | | Ignatius | | | Ballroom | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS |
| CF-3DGS | 21.96 | 0.71 | 0.27 | 17.36 | 0.64 | 0.33 | 14.96 | 0.31 | 0.59 | 22.20 | 0.72 | 0.25 |
| SFGS | 22.46 | 0.72 | 0.22 | 20.86 | 0.75 | 0.19 | 12.16 | 0.32 | 0.72 | 17.42 | 0.49 | 0.27 |
| Ours | 25.56 | 0.84 | 0.16 | 27.60 | 0.90 | 0.12 | 22.12 | 0.71 | 0.22 | 25.94 | 0.86 | 0.12 |
| CO3D V2 | 46_2587_7531 | | | 110_13051_23361 | | | 245_26182_52130 | | | 415_57112_110099 | | |
| | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS |
| CF-3DGS | 25.44 | 0.80 | 0.21 | 29.69 | 0.89 | 0.29 | 27.24 | 0.85 | 0.30 | 22.14 | 0.64 | 0.34 |
| SFGS | 30.65 | 0.91 | 0.13 | 29.95 | 0.87 | 0.19 | 28.59 | 0.87 | 0.27 | 27.23 | 0.78 | 0.30 |
| Ours | 31.83 | 0.92 | 0.12 | 33.44 | 0.94 | 0.11 | 33.82 | 0.93 | 0.20 | 30.37 | 0.88 | 0.22 |
| Synthetic | classroom | | | lego_c2 | | | livingroom | | | bedroom | | |
| | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS |
| CF-3DGS | 19.69 | 0.69 | 0.46 | 15.93 | 0.31 | 0.55 | 16.63 | 0.57 | 0.57 | 16.98 | 0.65 | 0.45 |
| SFGS | 16.79 | 0.67 | 0.55 | 13.64 | 0.29 | 0.68 | 15.11 | 0.49 | 0.60 | 12.39 | 0.54 | 0.54 |
| Ours | 36.26 | 0.94 | 0.15 | 29.36 | 0.90 | 0.12 | 33.52 | 0.88 | 0.24 | 31.17 | 0.93 | 0.13 |

Table 1: NVS results on Tanks and Temples, CO3D-V2 and Synthetic. Each baseline method is trained with its public code under the original settings and evaluated with the same evaluation protocol. The best results are gray background.

# Experiments

## Experimental Setup

**Datasets.** We conduct experiments on two real-world datasets, *CO3D-V2* (Reizenstein et al. 2021) and *Tanks and Temples* (Knapitsch et al. 2017), and a *Synthetic Dataset* created by ourselves. **CO3D-V2** includes thousands of videos of various objects. Following CF-3DGS (Fu et al. 2024), we select scenes with significant camera movements to demonstrate our robustness. **Tanks and Temples** used in CF-3DGS is overly smooth and lacks camera motions which are common in real-world scenarios. We reduce the original 20 fps to 4 fps and perform sampling along a longer camera trajectory to obtain more challenging cases with large camera motions. **Synthetic Dataset** comprises 4 scenes with about 150 frames each created using Blender (Community 2018), showcasing complex roaming and object-centric camera motions. It's used to assess camera parameter estimation, providing ground truth for intrinsic and extrinsic parameters. For additional details on Tanks and Temples and Synthetic Dataset, see the supplementary material.

**Metrics.** We use standard evaluation metrics, including PSNR, SSIM (Wang et al. 2004), and LPIPS (Zhang et al. 2018) to evaluate the quality of novel view synthesis (NVS). For pose estimation, we rely on the Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) (Lin et al. 2021; Bian et al. 2023; Fu et al. 2024). $\mathrm{RPE}_r$ and $\mathrm{RPE}_t$ are utilized to measure the accuracy of rotation and translation. To ensure the metrics are comparable on the same scale, we align the camera poses using Umeyama's method (Umeyama 1991) for both estimation and evaluation. For camera focal length, we convert it to the field of view (FoV) and calculate the angular error, following (Zhu et al. 2023).

**Implementation Details.** Our implementation is primarily based on *gsplat* (Ye et al. 2024), an accelerated 3DGS library. We modify the CUDA operator to backpropagate gradients for camera parameters. All parameters are optimized using Adam. For initialization, we optimize the relative pose between frames, and focal length. In joint process, the 3DGS parameters, absolute poses of cameras, and focal length are optimized. As vanilla 3DGS framework demonstrates inherent limitations in simultaneous optimization of all cameras within a single step. We introduce a gradient accumulation strategy (Hermans, Spanakis, and Möckel 2017) that enables co-optimization of the track Gaussians and all camera parameters in a bundle adjustment manner. The camera pose is represented as a combination of an axis-angle representation $\mathrm{q} \in \mathfrak{so}(3)$ and a translation vector $\mathrm{t} \in \mathbb{R}^3$. During training, we will clone new Gaussians from those links to the track

| CO3D V2 | 46_2587_7531 | | | 110_13051_23361 | | | 245_26182_52130 | | | 415_57112_110099 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\mathrm{RPE}_t \downarrow$ | $\mathrm{RPE}_r \downarrow$ | ATE$\downarrow$ | $\mathrm{RPE}_t$ | $\mathrm{RPE}_r$ | ATE | $\mathrm{RPE}_t$ | $\mathrm{RPE}_r$ | ATE | $\mathrm{RPE}_t$ | $\mathrm{RPE}_r$ | ATE |
| CF-3DGS | 0.095 | 0.447 | 0.009 | 0.140 | 0.401 | 0.021 | 0.239 | 0.472 | 0.017 | 0.110 | 0.424 | 0.014 |
| SFGS | 0.025 | 0.275 | 0.004 | 0.093 | 0.331 | 0.020 | 0.064 | 0.438 | 0.017 | 0.049 | 0.351 | 0.024 |
| Ours | 0.013 | 0.080 | 0.001 | 0.012 | 0.052 | 0.001 | 0.005 | 0.029 | 0.001 | 0.004 | 0.024 | 0.001 |

Table 2: Quantitative comparison of pose accuracy on CO3D-V2. The unit of $\mathrm{RPE}_r$ is in degrees, ATE is in the ground truth scale and $\mathrm{RPE}_t$ is scaled by 100. See the supplementary material for additional results.


Figure 4: Qualitative comparison for NVS and pose estimation on CO3D-V2. Benefit from the accuracy of the camera pose estimation, the rendering quality of novel view synthesis obtained by our method is higher than CF-3DGS.

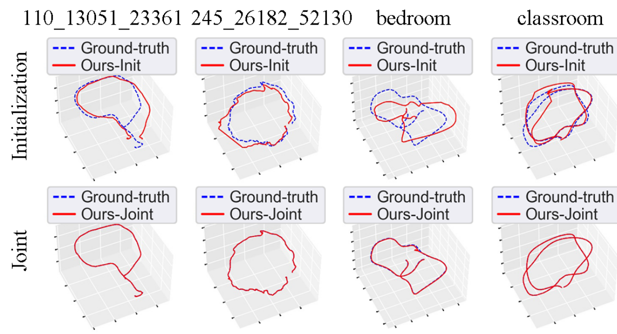


Figure 5: The trajectory of initial stage and joint stage. Our joint stage significantly improved the accuracy of camera pose.

points and apply the same training strategy as the original 3DGS (including clone, split, and delete). The number of track Gaussians is kept constant throughout training to ensure consistency with the imposed track constraints. We set $\lambda = 0.2$, $\lambda_{t2d} = 0.01$, $\lambda_{t3d} = 0.01$ in Eq. 10 for training.

## Experimental Results and Analysis

**Novel View Synthesis.** Since the camera poses of test views are unknown, we need to estimate them for rendering. Following CF-3DGS (Fu et al. 2024), we obtain these test-view poses by minimizing the photometric error between the synthesized images and the test views using the pre-trained 3DGS model. We apply the same procedure to all baseline methods to maintain a consistent bias for a fair comparison.

We report the results on CO3D-V2 and Tanks and Temples in Tab. 1. Our method consistently outperforms all baselines on these real-world datasets with complex camera motions. Both CF-3DGS and SFGS are less effective on these challenging datasets. The main reason is that these methods rely heavily on local constraints. When the training sequences are sparsely sampled (e.g., 20 fps $\rightarrow$ 4 fps), photometric loss and relative pose estimation are highly prone to failure and become ineffective. However, our proposed track loss effectively imposes global geometric constraints under such conditions. As illustrated in Fig. 3 and 4, the advantages of our algorithm are well demonstrated, especially with large camera motions. Due to global joint optimization, multi-view geometric consistency is better maintained in the trained 3DGS model, leading to high-quality images.

For further comparison, we evaluated our method on the Synthetic Dataset in Tab. 1, which features extremely complex camera motions. One result is shown in the bottom-right of Fig. 1. In this case, the camera not only moves in multiple circles around the object but also changes significantly in the vertical direction. Our synthesized image from the novel view remains clear and sharp, whereas the CF-3DGS result is blurry with obvious artifacts.

**Camera Parameter Estimation.** First, we compare the camera pose estimation with baseline methods. In the comparison, our method only assumes a constant camera focal length across frames, while others additionally input the camera focal length. The estimated camera poses are analyzed by Procrusts as in CF-3DGS and compared with the ground-truth of training views. The quantitative results of camera pose estimation on CO3D-V2 datasets are summarized in Tab. 2. The results show that our estimated camera parameters achieve the smallest error among all methods, with the Absolute Trajectory Error (ATE) being only one-tenth of that of the second-best method. This demonstrates that our algorithm excels in scenes with complex camera motions. Compared to the baselines, the global tracking information we use eliminates accumulated errors, leading to more accurate camera pose estimation. Additionally, joint optimization enhances the stability of the estimation results.

Next, we evaluate camera parameter estimation on our Synthetic Dataset. Tab. 3 shows the estimation errors from CF-3DGS, SFGS, COLMAP, and ours. Note that CF-3DGS and SFGS use the camera FoV estimated by COLMAP. We find that our estimated camera FoVs and poses are comparable to those of COLMAP, and the camera pose error is 50 times smaller than baselines. Fig. 5 visualizes our estimated poses in different stages. Thanks to the joint optimization based on global track and the back-propagation of the gradient of the camera parameters, our approach is able to combine these two tasks, reducing the input requirements.

| Scenes | classroom | | | | lego_c2 | | | | livingroom | | | | bedroom | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | FoV(°) | $RPE_t$ | $RPE_r$ | ATE | FoV(°) | $RPE_t$ | $RPE_r$ | ATE | FoV(°) | $RPE_t$ | $RPE_r$ | ATE | FoV(°) | $RPE_t$ | $RPE_r$ | ATE |
| CF-3DGS | 0.993 | 0.588 | 2.436 | 0.07412 | 0.021 | 1.126 | 4.946 | 0.10795 | 0.029 | 0.425 | 2.104 | 0.07653 | 0.042 | 0.366 | 1.103 | 0.06284 |
| SFGS | 0.993 | 0.605 | 2.148 | 0.07811 | 0.021 | 1.101 | 4.519 | 0.10904 | 0.029 | 0.400 | 1.817 | 0.07356 | 0.042 | 0.343 | 1.075 | 0.06492 |
| COLMAP | 0.993 | 0.004 | 0.018 | 0.00023 | 0.021 | 0.009 | 0.026 | 0.00019 | 0.029 | 0.008 | 0.026 | 0.00014 | 0.042 | 0.009 | 0.035 | 0.00023 |
| Ours | 0.012 | 0.002 | 0.013 | 0.00008 | 0.031 | 0.002 | 0.015 | 0.00011 | 0.012 | 0.002 | 0.013 | 0.00009 | 0.003 | 0.013 | 0.062 | 0.00044 |

Table 3: Quantitative comparison of parameter accuracy on our Synthetic Dataset. We convert the estimated camera intrinsics focal to FoV and perform the errors of FoV with ground truth (provided by our synthetic datasets). As CF-3DGS and SFGS require the camera intrinsic parameters as fixed inputs, we set them the same as COLMAP+3DGS.

| ID | Variant | PSNR↑ | SSIM↑ | LPIPS↓ | ATE↓ | FoV↓ |
|---|---|---|---|---|---|---|
| 1 | COLMAP + 3DGS | 32.26 | 0.91 | 0.18 | 0.00020 | 0.271 |
| 2 | CF-3DGS | 17.30 | 0.55 | 0.51 | 0.08036 | 0.271 |
| 3 | SFGS | 14.48 | 0.50 | 0.59 | 0.08141 | 0.271 |
| 4 | w.o. 2D track | 18.18 | 0.56 | 0.47 | 0.02020 | 2.617 |
| 5 | w.o. 3D track | 32.38 | 0.91 | 0.17 | 0.00275 | 0.063 |
| 6 | Ours | 32.58 | 0.92 | 0.16 | 0.00018 | 0.015 |

Table 4: Ablation study on our Synthetic Dataset.

| Scenes | Ours | | Ours(FoV=60°) | | COLMAP+3DGS | |
|---|---|---|---|---|---|---|
| | PSNR | LPIPS | PSNR | LPIPS | PSNR | LPIPS |
| classroom | 36.26 | 0.15 | 34.25 | 0.16 | 35.81 | 0.15 |
| lego_c2 | 29.36 | 0.12 | 23.82 | 0.26 | 28.77 | 0.15 |
| livingroom | 33.52 | 0.24 | 26.31 | 0.32 | 32.74 | 0.27 |
| bedroom | 31.17 | 0.13 | 24.01 | 0.25 | 31.73 | 0.13 |

Table 5: NVS results on our Synthetic Dataset compared with Fixed FoV and COLMAP-assisted 3DGS.

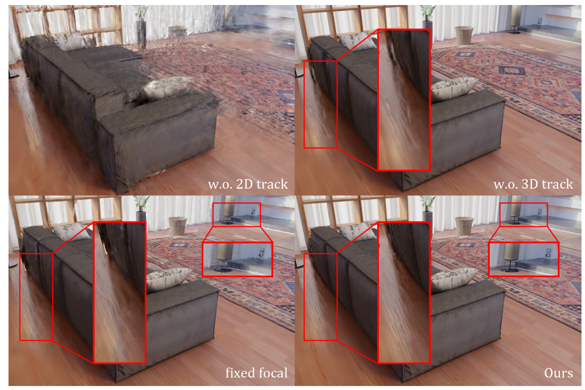


Figure 6: Visualization of ablation study on Synthetic Dataset. Without 2D/3D track loss or intrinsic optimization, the result appears blurry in novel view synthesis.

## Ablation Study

**Effectiveness of Different Losses.** We ablate each loss of the algorithm on Synthetic Dataset, since it has ground-truth camera parameters. Tab. 4 reports the average synthesis quality and camera parameter errors across different algorithm variants (see supplementary material for details). First, any variant of our algorithm (Variant 4, 5, 6) is better than the baseline methods (Variant 2, 3) in synthesis quality and absolute camera position. Second, Variant 4 shows that 2D track loss plays a crucial role in the entire joint optimization. When 2D track loss is not used, compared with the final method (Variant 4 vs. 6), there is a significant decrease in synthesis quality (18.18 vs. 32.58), and the camera parameter error is significantly larger (2.617 vs. 0.015). This shows that the reprojection error constrained by global consistency can significantly enhance the camera parameter estimation, thereby improving the 3DGS training effect and improving the new perspective synthesis ability, as illustrated in Fig. 6. In addition, the results of Variant 5 vs. 6 show that 3D track loss can further enhance the geometric consistency of 3DGS. Using 3D track loss, the PSNR of NVS can be further improved by 0.2 dB, and the ATE of the camera can be reduced by an order of magnitude.

**Effectiveness of Intrinsic Optimization.** Accurate camera intrinsics resolve scale ambiguity in 3DGS models, leading to improved NVS performance. As shown in Tab. 3, our method produces more accurate intrinsics (i.e., FoV) compared to COLMAP with vanilla 3DGS. We also performed an ablation study with a fixed camera FoV of 60° and without further optimization. The results, shown in Tab. 5, indicate a 21.4% average decrease in PSNR, due to scale ambiguity introduced by inaccurate camera intrinsics.

**Comparison with COLMAP-Assisted 3DGS.** We compare the NVS results generated by our method against vanilla 3DGS, where the camera intrinsics and extrinsics are estimated using COLMAP on Synthetic Dataset. Tab. 5 shows that our method outperforms COLMAP-assisted 3DGS in most scenes. Unlike the original 3DGS, which uses a fixed camera pose for training, our method seamlessly integrates 3DGS training with camera parameter estimation, allowing the two tasks to complement each other and ultimately achieve high-quality novel view synthesis.

## Conclusion

We presented TrackGS, a novel framework that successfully integrates global feature tracks with 3D Gaussian Splatting to achieve COLMAP-free novel view synthesis. By establishing geometric constraints through track Gaussians and introducing novel 2D/3D consistency losses, our method jointly optimizes camera parameters and scene representation with high accuracy. The differentiable formulation of camera intrinsics completes the pipeline, removing all dependencies on precomputed parameters. Experiments demonstrate significant improvements over state-of-the-art methods in both rendering quality and pose estimation accuracy across diverse scenarios. Future extensions include extending the framework to per-view intrinsics, distortion modeling, and dynamic scene support.

## Acknowledgements

This work was partly supported by the Anhui Provincial Natural Science Foundation (2508085MA001).

## References

Bian, W.; Wang, Z.; Li, K.; Bian, J.; and Prisacariu, V. A. 2023. NoPe-NeRF: Optimising Neural Radiance Field with No Pose Prior. In *CVPR*.

Chng, S. F.; Ramasinghe, S.; Sherrah, J.; and Lucey, S. 2022. GARF: Gaussian Activated Radiance Fields for High Fidelity Reconstruction and Pose Estimation.

Community, B. O. 2018. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam.

DeTone, D.; Malisiewicz, T.; and Rabinovich, A. 2018. Superpoint: Self-supervised interest point detection and description. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 224–236.

Fan, Z.; Cong, W.; Wen, K.; Wang, K.; Zhang, J.; Ding, X.; Xu, D.; Ivanovic, B.; Pavone, M.; Pavlakos, G.; Wang, Z.; and Wang, Y. 2024. InstantSplat: Sparse-view SfM-free Gaussian Splatting in Seconds. arXiv:2403.20309.

Fu, Y.; Liu, S.; Kulkarni, A.; Kautz, J.; Efros, A. A.; and Wang, X. 2024. COLMAP-Free 3D Gaussian Splatting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 20796–20805.

Hermans, J. R.; Spanakis, G.; and Möckel, R. 2017. Accumulated gradient normalization. In *Asian Conference on Machine Learning*, 439–454. PMLR.

Hu, R.; Ravi, N.; Berg, A. C.; and Pathak, D. 2021. Worldsheet: Wrapping the World in a 3D Sheet for View Synthesis from a Single Image. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Jeong, Y.; Ahn, S.; Choy, C.; Anandkumar, A.; Cho, M.; and Park, J. 2021. Self-Calibrating Neural Radiance Fields. In *ICCV*.

Ji, B.; and Yao, A. 2024. SfM-Free 3D Gaussian Splatting via Hierarchical Training. arXiv:2412.01553.

Jiang, K.; Fu, Y.; Varma T, M.; Belhe, Y.; Wang, X.; Su, H.; and Ramamoorthi, R. 2024. A Construct-Optimize Approach to Sparse View Synthesis without Camera Pose. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers '24*, volume 2022 of *SIGGRAPH '24*, 1–11. ACM.

Kerbl, B.; Kopanas, G.; Leimkühler, T.; and Drettakis, G. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4).

Knapitsch, A.; Park, J.; Zhou, Q.-Y.; and Koltun, V. 2017. Tanks and temples: benchmarking large-scale scene reconstruction. *ACM Trans. Graph.*, 36(4).

Leroy, V.; Cabon, Y.; and Revaud, J. 2024. Grounding Image Matching in 3D with MASt3R. In *ECCV*, 71–91. Berlin, Heidelberg: Springer-Verlag. ISBN 978-3-031-73219-5.

Li, J.; Feng, Z.; She, Q.; Ding, H.; Wang, C.; and Lee, G. H. 2021. MINE: Towards Continuous Depth MPI with NeRF for Novel View Synthesis. In *ICCV*.

Lin, C.-H.; Ma, W.-C.; Torralba, A.; and Lucey, S. 2021. BARF: Bundle-Adjusting Neural Radiance Fields. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 5721–5731.

Lin, H.; Peng, S.; Xu, Z.; Yan, Y.; Shuai, Q.; Bao, H.; and Zhou, X. 2022. Efficient Neural Radiance Fields for Interactive Free-viewpoint Video. In *SIGGRAPH Asia Conference Proceedings*.

Mai, J.; Zhu, W.; Rojas, S.; Zarzar, J.; Hamdi, A.; Qian, G.; Li, B.; Giancola, S.; and Ghanem, B. 2024. TrackNeRF: Bundle Adjusting NeRF from Sparse and Noisy Views via Feature Tracks. In *European Conference on Computer Vision*.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *European Conference on Computer Vision*.

Müller, T.; Evans, A.; Schied, C.; and Keller, A. 2022. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Trans. Graph.*, 41(4): 102:1–102:15.

Ranftl, R.; Bochkovskiy, A.; and Koltun, V. 2021. Vision Transformers for Dense Prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 12179–12188.

Reizenstein, J.; Shapovalov, R.; Henzler, P.; Sbordone, L.; Labatut, P.; and Novotny, D. 2021. Common Objects in 3D: Large-Scale Learning and Evaluation of Real-life 3D Category Reconstruction. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 10881–10891.

Riegler, G.; and Koltun, V. 2020. Free View Synthesis. In *European Conference on Computer Vision*.

Riegler, G.; and Koltun, V. 2021. Stable View Synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Sarlin, P.-E.; DeTone, D.; Malisiewicz, T.; and Rabinovich, A. 2020. SuperGlue: Learning Feature Matching with Graph Neural Networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4938–4947.

Schonberger, J. L.; and Frahm, J.-M. 2016. Structure-From-Motion Revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tucker, R.; and Snavely, N. 2020. Single-view View Synthesis with Multiplane Images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Umeyama, S. 1991. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4): 376–380.

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

Wang, Z.; Wu, S.; Xie, W.; Chen, M.; and Prisacariu, V. A. 2021. NeRF−−: Neural Radiance Fields Without Known Camera Parameters. *arXiv preprint arXiv:2102.07064*.

Xu, Q.; Xu, Z.; Philip, J.; Bi, S.; Shu, Z.; Sunkavalli, K.; and Neumann, U. 2022. Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5438–5448.

Ye, V.; Li, R.; Kerr, J.; Turkulainen, M.; Yi, B.; Pan, Z.; Seiskari, O.; Ye, J.; Hu, J.; Tancik, M.; and Kanazawa, A. 2024. gsplat: An Open-Source Library for Gaussian Splatting. arXiv:2409.06765.

Yen-Chen, L.; Florence, P.; Barron, J. T.; Rodriguez, A.; Isola, P.; and Lin, T.-Y. 2021. iNeRF: Inverting Neural Radiance Fields for Pose Estimation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1323–1330.

Zhang, Q.; Baek, S.-H.; Rusinkiewicz, S.; and Heide, F. 2022. Differentiable Point-Based Radiance Fields for Efficient View Synthesis. *arXiv preprint arXiv:2205.14330*.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *2018 IEEE/CVF 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. In *SIGGRAPH*.

Zhu, S.; Kumar, A.; Hu, M.; and Liu, X. 2023. Tame a Wild Camera: In-the-Wild Monocular Camera Calibration. In *NeurIPS*.