
MS-GS: Multi-Appearance Sparse-View 3D Gaussian Splatting in the Wild

Deming Li

Johns Hopkins University
Baltimore, MD 21218
dli90@jhu.edu

Kaiwen Jiang

University of California, San Diego
La Jolla, CA 92093
k1jiang@ucsd.edu

Yutao Tang

Johns Hopkins University
Baltimore, MD 21218
ytang67@jhu.edu

Ravi Ramamoorthi

University of California, San Diego
La Jolla, CA 92093
ravir@ucsd.edu

Rama Chellappa

Johns Hopkins University
Baltimore, MD 21218
rchella4@jhu.edu

Cheng Peng

Johns Hopkins University
Baltimore, MD 21218
cpeng26@jhu.edu

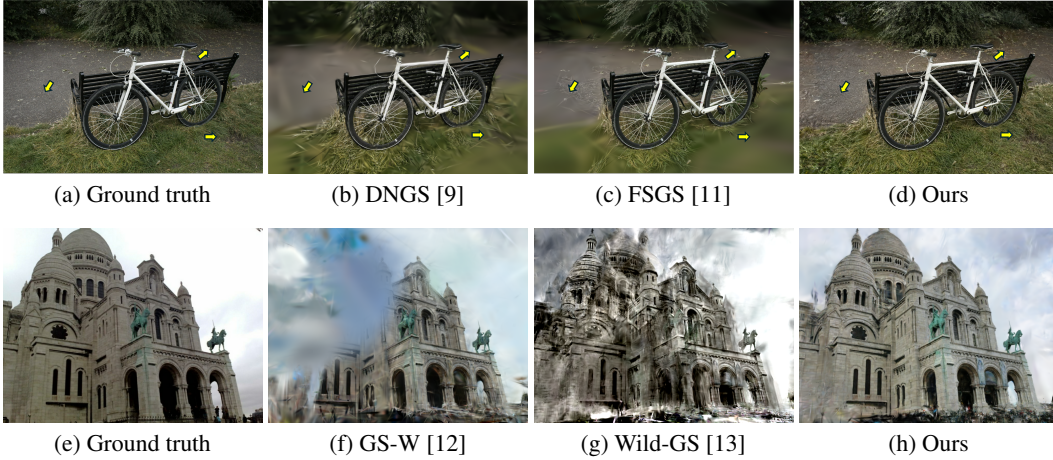
Abstract

In-the-wild photo collections often contain limited volumes of imagery and exhibit multiple appearances, e.g., taken at different times of day or seasons, posing significant challenges to scene reconstruction and novel view synthesis. Although recent adaptations of Neural Radiance Field (NeRF) and 3D Gaussian Splatting (3DGS) have improved in these areas, they tend to oversmooth and are prone to overfitting. In this paper, we present MS-GS, a novel framework designed with Multi-appearance capabilities in Sparse-view scenarios using 3DGS. To address the lack of support due to sparse initializations, our approach is built on the geometric priors elicited from monocular depth estimations. The key lies in extracting and utilizing local semantic regions with a Structure-from-Motion (SfM) points anchored algorithm for reliable alignment and geometry cues. Then, to introduce multi-view constraints, we propose a series of geometry-guided supervision steps at virtual views in pixel and feature levels to encourage 3D consistency and reduce overfitting. We also introduce a dataset and an in-the-wild experiment setting to set up more realistic benchmarks. We demonstrate that MS-GS achieves photorealistic renderings under various challenging sparse-view and multi-appearance conditions, and outperforms existing approaches significantly across different datasets.

1 Introduction

High-quality scene reconstruction and novel view synthesis from images is a long-standing research problem with wide-ranging applications in AR/VR, 3D site modeling, autonomous driving, robotics, etc. Remarkably, neural radiance field (NeRF) [1] and 3D Gaussian Splatting (3DGS) [2] achieve photorealistic novel view synthesis with differentiable rendering pipelines and different scene parameterizations. Both approaches build on the fundamental constraint that a voxel in space projects to similar photometric values across views, a constraint that requires a dense scene coverage and multi-view consistency. In practice, such coverage and consistency are often not guaranteed, largely affecting their performance in unconstrained settings.

Figure 1: With **20 input views**, DNGS and FSGS produce overly smooth rendering in regions lacking support from sparse point cloud initialization. For scenes with multiple appearances and sparse inputs, methods like GS-W and Wild-GS experience large artifacts at novel views. In contrast, our method in Fig. 1d and 1h renders details and provides a coherent reconstruction.



With sparse image sets, overfitting the photometric objectives to an incorrect geometry is a common issue in novel view synthesis. To counter this, semantic constraints [3], depth and novel-view regularization [4, 5, 6], frequency regularization [7], and ray-entropy minimization [8], have been introduced. While effective, these NeRF-based methods incur a heavy computational cost because volumetric rendering requires densely sampling points along camera rays. More recently, 3DGS adaptations have pushed sparse-view synthesis further by exploiting their explicit representation and fast rasterization. Depth regularization [9], floater pruning [10], and proximity-based Gaussian densification [11] regularize the reconstruction and suppress artifacts during training. Despite improved metrics, these methods remain limited by the standard initialization of a sparse Structure-from-Motion (SfM) point cloud when few features are triangulated correctly. Furthermore, applying monocular depth constraints globally to 3DGS is often inaccurate, leading to noisy gradients that prevent proper densification in sparse regions. As shown in Fig. 1, they synthesize overly smooth regions, while our method recovers fine details.

Beyond limited viewpoints, in-the-wild image sets often exhibit photometric inconsistencies across views. These range from subtle exposure shifts to pronounced appearance changes when the same scene is captured at different times of day or in different weather. NeRF-in-the-Wild [14] first demonstrated the ability to model a canonical 3D structure from multi-appearance imagery; since then, various works have followed up to improve synthesis quality [15, 16] and incorporate this capability to 3DGS-based approaches [12, 13, 17]. These methods require more data to disambiguate image-specific radiance compared to appearance-consistent scenarios. As shown in Fig. 1, a moderate number of views still leads to noisy rendering, greatly limiting the application of these methods.

In this paper, we present MS-GS, which improves the robustness of 3DGS in dealing with unconstrained images when limited viewpoints and varying appearances exist, which is underexplored. We find that the performance of 3DGS relies heavily on the initial point cloud: these explicit structures steer the adaptive control of Gaussians and subsequent optimization. To overcome the limitation of the sparse SfM point cloud with limited views, we draw knowledge from the monocular depth estimators [18, 19, 20] that have rapidly progressed. MS-GS aligns the depth prediction with SfM depth, then back-projects pixels into the scene space for a dense point cloud. A key challenge is that monocular depth estimation is often incorrect at relative depth between objects due to single-view ambiguity. We address this with a **Semantic Depth Alignment** approach. A point-prompted segmentation model [21] is leveraged to extract semantically consistent regions using projected SfM points. We design an iterative refinement algorithm to identify each region—expanding or discarding according to the number of enclosed SfM points—and perform alignment inside it before back-projection. The resulting point cloud is denser and better structured than the original sparse SfM output, helping regularize 3DGS structures and promote Gaussian densification.

To enable sparse-view multi-appearance scene modeling, MS-GS decomposes appearance into an image-specific and Gaussian-specific component. The per-image appearance embedding captures the global appearance variations, while each Gaussian’s feature embedding encodes the canonical scene appearance. Under sparse-view conditions, it becomes challenging to disambiguate the image-specific radiance from a consistent scene appearance, resulting in overfitting to training image appearance. Building upon the sufficiently accurate geometry that 3DGS optimizes with our dense initialization, we exploit the use of virtual views for **multi-view** constraints, where a series of **Geometry-guided Supervisions** based on 3D warping is proposed. Specifically, we back-project training images to 3D and then project to virtual views created between training cameras to establish correspondences. Appearance consistency is enforced at pixel and feature levels for precise supervision and handling occluded areas. This approach aims to transfer the well-rendered appearance of training images to multiple views. Coupled with our densified point cloud, this design markedly improves geometric coherence and the rendering quality given sparse and multi-appearance imagery.

In addition, the benchmark dataset Phototoursim [22] is collected through the internet such that each image has a unique appearance. Therefore, methods evaluated on this dataset require access to the test view image to obtain appearance, which is not ideal for train-test separation. To this end, we introduce an unbounded drone dataset that features *multi-view* appearance. By relating to camera metadata, not the pixel information, testing views are rendered with the appearance of training images during evaluation. Similarly, the experimental setting of novel view synthesis in the wild is non-trivial. While methods often assume camera poses are exact, sparse-view and multi-appearance registration are themselves prone to error. Accurate reflection of performance in the wild needs to account for realistic registration noise, especially when the underlying SfM point cloud is the input to the 3DGS-based methods. We therefore advocate a protocol that disentangles training and testing cameras during registration and preserves real-world pose uncertainties.

In summary, the main contributions of our work are:

- We introduce a Semantic Depth Alignment approach, which leverages monocular depths in local semantic regions to construct a dense point cloud initialization and significantly improves fidelity in regions with limited overlap.
- We propose a series of Multi-view Geometry-guided Supervision steps based on 3D warping at pixel and feature levels; such a framework reduces overfitting to limited observation and encourages 3D geometry and appearance consistency.
- We evaluate our overall method, MS-GS, across various benchmark datasets in different evaluation settings. MS-GS demonstrates significant quantitative and visual improvements compared to SoTA methods.

2 Related Work

Sparse-view Novel View Synthesis To improve sparse-view novel view synthesis, DietNeRF [3] uses scene semantics from a pre-trained visual encoder to constrain a 3D representation. RegNeRF [4] regularizes geometry by enforcing smoothness on rendered depth and appearance by a normalizing flow model in patches from unseen viewpoints. FreeNeRF [7] proposes a low-to-high frequency schedule and penalization on density fields near the camera as regularization. Various depth signals were explored to distill depth priors to the training of a NeRF [5, 6]. In the realm of 3DGS, depth regularization [23, 9] with global and local normalization is applied to constrain the 3D radiance field. SparseGS [10] uses depth priors and diffusion loss with a floater-pruning strategy to enhance the quality of renderings from unseen viewpoints. FSGS [11] grows Gaussians with a proximity-based Gaussian unpooling strategy regularized by depth. Previous approaches mainly target 3DGS training. SPARS3R [24] utilizes a pointmap estimator MAST3R [25] for 3DGS initialization, whereas we use monocular depths. Monocular estimates are typically sharper due to less constraints and have been shown to serve as a coarse solution for optimization in unposed reconstruction [26]. In addition, significant appearance variations pose challenges in MAST3R’s view-consistent output. In our paper, we demonstrate the effectiveness of our proposed back-projected point cloud as an improved initialization strategy.

Novel View Synthesis with varying appearances Casual photo collections often include images taken at different times or seasons, resulting in inconsistent appearances. To address the limitations of

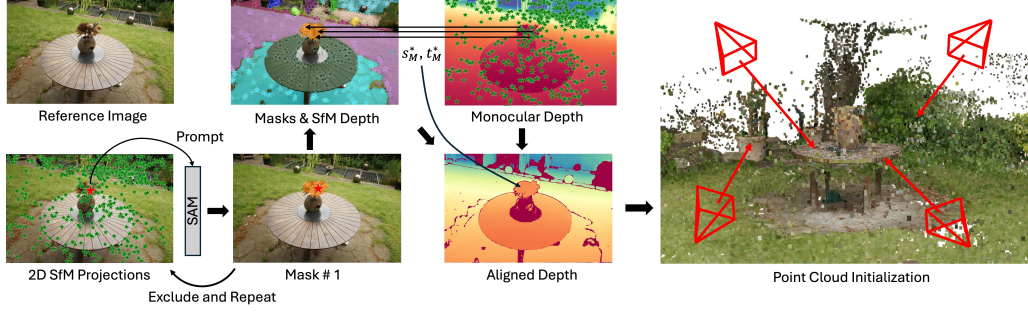


Figure 2: **Overview of our depth prior initialization of MS-GS.** Semantic masks and corresponding SfM point depth within each mask are obtained through our SfM-prompted Semantic module, detailed in Section 3. We then align monocular depth to SfM depth for each mask by computing the optimal scale s_M^* and shift t_M^* . The point cloud is obtained from the back-projection of aligned depths and corresponding image pixel values to construct 3DGS initialization.

vanilla NeRF and 3DGS, which assume static scenes with consistent appearance, subsequent works integrate additional feature representations to account for appearance changes. NeRF in-the-wild (NeRF-W) [14] models static and transient volumes conditioned on image embeddings. Ha-NeRF [15] employs a convolutional neural network (CNN) to extract image appearance features and introduces a view-consistent loss to encourage consistent appearance across viewpoints. CR-NeRF [16] explores cross-ray features and their interaction with global image appearance for better appearance modeling. Recent adaptations of 3DGS for in-the-wild settings include GS-W [12], which uses an adaptive sampling strategy based on 2D feature maps to capture both dynamic and intrinsic appearance for each Gaussian. Wild-GS [13] extends feature representation to 3D as triplane features by incorporating rendered depth with input images. WildGaussians [17] leverages appearance embedding and DINO [27] features to handle appearance changes and occlusions with 3DGS. While these approaches are effective when abundant images are available, their performance severely degrades with sparse inputs, necessitating further advances in this line of research.

3 Method

Our proposed method, MS-GS, builds on the efficient 3DGS framework, in which a scene is represented by a set of Gaussian primitives $\{\mathcal{G}_i\}_{i=1}^N$. Each Gaussian \mathcal{G}_i is defined with the learnable parameters: xyz position $\mu_i \in \mathbb{R}^3$, opacity $\alpha_i \in \mathbb{R}$, color $c_i \in \mathbb{R}^3$, scale $s_i \in \mathbb{R}^3$, and quaternion $q_i \in \mathbb{R}^4$ for rotation. To improve its robustness in sparse-view synthesis and multi-appearance modeling, MS-GS consists of two parts: Semantic Depth Alignment first constructs a dense point cloud by expanding SfM points based on monocular depth and their semantic relevance (Section 3.1), illustrated in Fig. 2. MS-GS then introduces a series of geometry-guided supervisions based on 3D warping at a fine-grained pixel level and coarse feature level. (Section 3.2).

3.1 Semantic depth alignment

Gaussian-Splatting-based methods rely on discrete optimization to densify and prune Gaussians to fit the scene. Such discrete optimization is non-smooth and easily gets stuck at local minima; therefore, a good initial point cloud is crucial and provides anchors from which densification occurs. Conventionally, such a point cloud initialization comes from a prior SfM process and is assumed to be reasonably dense. In sparse-view or view-inconsistent scenarios, this assumption is often invalid due to insufficient correspondences. Therefore, we seek to densify the initial sparse point cloud based on monocular depth estimation.

SfM-anchored alignment After camera calibration, we have a set of \mathcal{N} images $\{I_n | n = 1, 2, \dots, \mathcal{N}\}$, an initial SfM point cloud $X \in \mathbb{R}^{P \times 3}$ and the camera poses. Applying world-to-camera transformation W and camera intrinsics K provides us with projected points $\mathcal{X} \in \mathbb{R}^{\bar{P} \times 2}$ with the pixel coordinate $u_n(i), v_n(i)$ and depth $d_n^{\text{sfm}}(i)$ of $x_i \in \mathcal{X}$ on image I_n . Note that from the

calibration process, we have a visibility function that indicates if x_i is visible in I_n , preventing e.g. points from behind a wall from being projected.

Given a monocular depth estimation model [19], we can obtain a dense depth D_n^{mono} for image n . While D_n^{mono} is not in SfM scale and is not multi-view consistent, d_n^{sfm} can be used to align D_n^{mono} . Specifically, a Least-Squares formulation is solved to find the optimal scale s_n^* and shift t_n^* for such alignment:

$$s_n^*, t_n^* = \arg \min_{s, t} \|s \cdot D_n^{\text{mono}}(u_n(i), v_n(i)) + t - d_n^{\text{sfm}}(i)\|_2 \forall i, \quad (1)$$

where $\|\cdot\|_2$ denotes the Euclidean norm. Once s_n^* and t_n^* are estimated, the aligned monocular depth $D_n^{\text{mono}*} = s_n^* \cdot D_n^{\text{mono}} + t_n^*$ can be back-projected into space to form a dense point cloud with all images by

$$X^{\text{mono}} = \bigcup_{n=1}^{\mathcal{N}} W^{-1} K^{-1} D_n^{\text{mono}*}(u, v) \forall u, v \in I_n. \quad (2)$$

While this formulation can be efficiently computed to construct a very dense point cloud, such a point cloud will be very noisy. Firstly, while the estimated D_n^{mono} may be visually pleasing and detailed, the relative depth between objects within an image is often inaccurate due to its inherent ambiguity. Secondly, in a sparse-view scenario, the number of reliable SfM points is limited. Even if the error in Eq. (1) is minimized, it's unclear whether regions without sufficient constraints, i.e. d_n^{sfm} , are properly aligned. A noisy X^{mono} does not improve NVS quality, as the dense but inaccurate points give rise to artifacts due to noisy gradients and lead to overfitting. To eliminate unreliable depth estimation in the alignment process, we propose an SfM-prompted Semantic Alignment scheme.

SfM-prompted semantic alignment We propose finding semantic regions enclosed by depth discontinuity using projected SfM points and performing individual alignment. As shown in Fig. 2, this is an iterative refinement process. Given a set of visible \bar{P} SfM points $x_i \in \mathcal{X}$ projected on image I_n , we take a point x_i and predict its semantically relevant region through an interactive segmentation model [21] \mathcal{S} , i.e. $M_i = \mathcal{S}(x_i, I_n)$. The intersection SfM points between \mathcal{X} and M_i are represented as $x_{m,i}$, which are semantically related to x_i within the mask. We determine if M_i has enough support by a threshold on $|x_{m,i}|$, the number of SfM points in the mask. To address the situations where insufficient points are caused by partial mask prediction, a second pass is performed $M_i = \mathcal{S}(x_{m,i}, I_n)$ if the threshold is not met the first time. Additionally, masks are checked for merging if they largely overlap to ensure semantic completeness. After each iteration, $x_{m,i}$ are removed from \mathcal{X} , and another x_i is sampled until empty, and we obtain a set of final masks $\{M_{\text{final}}\} \in \mathbb{R}^{\mathcal{M} \times H \times W}$ in the end. The detailed algorithm is presented in the supplement.

Similarly, an optimal scale s_m^* and shift t_m^* are computed to align monocular depth D_m^{mono} and SfM depth d_m^{sfm} for each mask:

$$\begin{aligned} s_m^*, t_m^* &= \arg \min_{s, t} \|s \cdot D_m^{\text{mono}}(u_m(i), v_m(i)) + t - d_m^{\text{sfm}}(i)\|_2 \forall i, \\ D_m^{\text{mono}*} &= s_m^* \cdot D_m^{\text{mono}} + t_m^* \end{aligned} \quad (3)$$

The point cloud is aggregated from all masks in each image through back-projection to initialize 3D Gaussians:

$$X^{\text{mono}} = \bigcup_{j=1}^{\mathcal{N}} \bigcup_{m=1}^{\mathcal{M}} W^{-1} K^{-1} D_m^{\text{mono}*}(u_m, v_m) \forall u_m, v_m \in M_{\text{final}}. \quad (4)$$

3.2 Multi-view geometry-guided supervisions

Modeling multi-appearance scenes under sparse-view constraints is especially difficult: view-specific lighting and weather variations demand more observations to disentangle appearance from structure. Consequently, overfitting, where the model memorizes the sparse training images instead of learning view-invariant geometry, becomes more severe than in the single appearance setting, as the SoTA methods for unconstrained settings exhibit obvious floaters and appearance inconsistencies. To encourage 3D consistency and appearance regularization, our method, illustrated in Fig. 3, exploits virtual cameras for multi-view supervision and utilizes 3D warping to establish correspondences for fine-grained pixel loss and coarse feature loss.

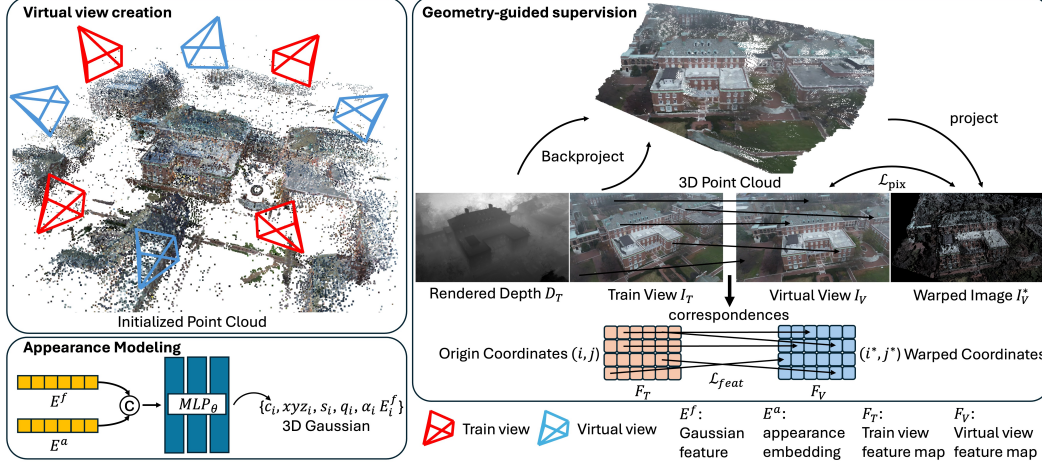


Figure 3: **Overview of our multi-view geometry-guided supervision of MS-GS.** Initialized from our proposed dense point cloud, we first create virtual views between training cameras. A 3D point cloud is back-projected given a training view I_T and its corresponding rendered depth D_T , and then forward-projected onto the virtual view to obtain the warped image I_V^* for a pixel loss. The correspondences from I_T to I_V^* are mapped to feature maps extracted from these two images to form a feature loss.

Appearance modeling To handle appearance variations, MS-GS uses per-image appearance embedding $E^a \in \mathbb{R}^{N \times 32}$ for optimization, where N is the number of images.

Meanwhile, MS-GS models a canonical scene representation through per-Gaussian feature embeddings $E^f \in \mathbb{R}^{N \times 16}$, where N is the number of Gaussians. A feature fusion network MLP_θ takes these two appearance components to decode the RGB colors of 3D Gaussians $c \in \mathbb{R}^{N \times 3}$:

$$c = MLP_\theta(E^a, E^f). \quad (5)$$

Virtual view creation The scene-reconstruction task is under-constrained when limited viewpoints are available. To introduce multi-view regularization, we create virtual views, which enable the additional supervision detailed in the following sections. At each iteration, we interpolate the current training view toward one of its top-k nearest-neighbour views. Camera translations are linearly interpolated with a weight in $[0, 1]$, while rotations are blended with SLERP [28]. The field of view (FOV) is also interpolated to avoid empty regions when moving between near-/far or wide-/narrow FOV pairs. Crucially, each virtual view uses the appearance embedding of the current training image, ensuring that it renders the same appearance and allows the optimization of the same embedding.

Pixel warping supervision Leveraging the geometry optimized from our proposed dense initialization, we create supervision based on 3D warping [29, 30, 31]. We warp one view onto another with known depth and camera matrices. We first render the training and its virtual view to get colors I_T, I_V and depths D_T, D_V . The training view pixels are back-projected to 3D points, then forward-projected onto the virtual view to produce a warped image I_V^* . Furthermore, pixels that have smaller values in the rendered depth D_V than those in the warped depth are removed due to occlusion, forming a mask M_{ocl} . This explicit pixel-wise loss is formulated as:

$$\mathcal{L}_{pix} = \|M_{ocl} \odot (I_V - I_V^*)\|_1. \quad (6)$$

Such a loss allows supervision from multiple views by leveraging reasonably accurate depth to constrain the geometry and appearance along the newly created rays at virtual cameras, as if a "floater" Gaussian exists that will alter the re-projected pixel color.

Semantic feature supervision While the pixel loss provides fine-grained supervision at virtual views, blank pixels are produced due to occlusions and rounding errors when moving viewpoints. Thus, we propose to use a coarse semantic feature supervision at the local patch level, i.e., the receptive field of each feature-map element. Given a feature extractor [32], we can obtain the feature maps of

the training view F_T and virtual view F_V . We make use of the geometry correspondences from 3D warping once again for more effective supervision. Formally, the feature map of the training view F_T is transformed to F_V^* , which is computed using cosine distance loss with F_V :

$$\mathcal{L}_{\text{feat}} = \text{dist}(F_V, F_V^*), \text{ where } F_V^*(i^*, j^*) = F_T(i, j), \quad (7)$$

where (i, j) and (i^*, j^*) are origin pixel coordinates on I_T and warped pixel coordinates on I_V . The correspondences $(i, j) \rightarrow (i^*, j^*)$, illustrated in Fig. 3, are mapped to the feature map resolution from image resolution.

Optimization Incorporating all the aforementioned techniques, the training objective of MS-GS is:

$$\mathcal{L}_{\text{total}} = \lambda_I \|I_T - I_T^*\|_1 + (1 - \lambda_I) \text{SSIM}(I_T, I_T^*) + \lambda_{\text{pix}} \mathcal{L}_{\text{pix}} + \lambda_{\text{feat}} \mathcal{L}_{\text{feat}}, \quad (8)$$

where I_T^* is the ground-truth training image. Notably, the geometry-guided supervisions start after the scene converges to a sufficiently accurate geometry to establish correspondences.

4 Experiments

4.1 Datasets

We evaluate the performance of MS-GS and current SoTA methods on three real-world scenes with sparse inputs—one with single appearance and two with varying appearances. **Sparse Mip-NeRF 360 Dataset** [33] contains 4 outdoor and 4 indoor scenes with a complex central object or area and a detailed background. We sampled 20 images from each scene for training. **Sparse Phototourism Dataset** [22] consists of scenes of well-known monuments. Specifically, we use "Brandenburg Gate", "Sacre Coeur", and "Trevi Fountain", following previous works [14, 12, 13, 17]. We sampled 20 images from the official training set and kept the same testing split for evaluation. Note that these are 2.62%, 2.41%, and 1.18% of the full training set for each scene, respectively. **Sparse Unbounded Drone Dataset**: We collected drone footage of 3 different buildings captured in orbit, creating a dataset featured in multi-view and multi-appearance scenarios. The dataset includes 4 distinct appearances: sunny, cloudy, snowy, and low-light, each captured with a full 360° view. We evenly sampled 5 images from each appearance, resulting in 20 images for training each scene. We aim to establish these benchmarks for sparse-view synthesis in unconstrained settings. Please find a fuller description of our dataset in the supplement.

4.2 Evaluation and Implementation

Most sparse-view synthesis methods [34, 3, 4, 6, 7, 8, 5, 9] assume ground-truth (GT) camera poses, i.e., calibration with dense views, thereby bypassing the challenges of registration and point triangulation. Thus, we propose an in-the-wild evaluation setting to evaluate sparse-view synthesis. A coordinate alignment method, provided in the supplement, is developed to perform separate registrations that disentangle training and testing images and then align them in the same coordinate system. Note that a slight pixel offset occurs during pose alignment, which disturbs pixel-based metrics PSNR and SSIM [35]. Therefore, we evaluate only with perceptual metrics LPIPS [36] and DreamSim (DSIM) [37] in this setting. We strongly encourage the readers to inspect the supplement for more details and analysis on the in-the-wild evaluation, metrics, and implementation.

Table 1: Ablation studies on different components of MS-GS. The metrics are reported as the average on the Sparse Unbounded drone dataset; **bold** numbers are the best, underscored second best.

Dense Init.	\mathcal{L}_{pix}	$\mathcal{L}_{\text{feat}}$	PSNR↑	SSIM↑	LPIPS↓	DSIM↓
×	×	×	18.49	0.492	0.367	0.151
✓	×	×	19.29	0.538	0.336	0.115
✓	✓	×	19.63	0.564	0.330	<u>0.104</u>
✓	×	✓	<u>19.65</u>	<u>0.569</u>	<u>0.328</u>	<u>0.104</u>
✓	✓	✓	19.87	0.580	0.322	0.096

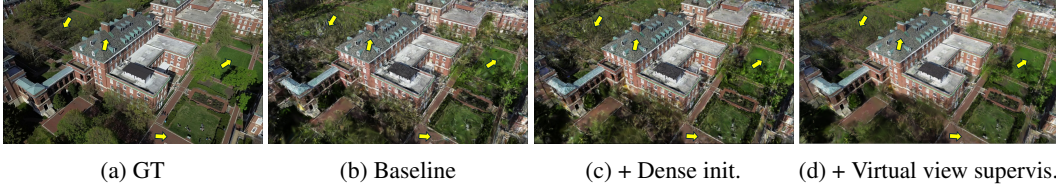


Figure 4: Novel view synthesis results when components are added sequentially. Please zoom in if possible for better visualization.

4.3 Ablation Study

We conduct an ablation study to validate the effectiveness of our method in Table 1 and Fig. 4. We refer to 3DGS augmented with multi-appearance capabilities using per-image embeddings and Gaussian feature embeddings as the baseline and report its metrics in the first row of Table 1. We identify that incorporating our semantic depth alignment initialization significantly improved the metrics with 0.8 dB in PSNR, 0.046 in SSIM, -0.031 in LPIPS, and -0.036 in DSIM. This result proves our hypothesis that optimizing 3DGS directly on a sparse SfM point cloud leaves ambiguities: novel views display incomplete surfaces and artifact Gaussians, shown in Fig. 4b. By contrast, our semantically dense initialization widens the solution space and provides reliable geometric support, enabling more effective Gaussian densification and pruning. This approach regularizes the scene structure and yields a more complete geometry, e.g., filling in the holes on the rooftop.

Next, adding 3D warping pixel loss and geometry-guided feature loss based on virtual views each further boosts the performance by a similar margin. The complete multi-view supervisions enhance the metrics by 0.58 dB in PSNR, 0.042 in SSIM, -0.014 in LPIPS, and -0.019 in DSIM. As visualized in Fig 4d, this strategy suppresses residual artifacts, on regions such as grass, trees, and road markings, and renders a more faithful reconstruction. The multi-view supervision enforces radiance consistency and also refines geometry through synergistic feedback. All proposed components are complementary, and the best results are achieved when combined. The analysis and visualization for semantic scaling validation can be found in the appendix.

4.4 Comparisons

Table 2: Quantitative Comparison on sparse Mip-NeRF 360 dataset; **bold** numbers are the best, underscored second best. We only evaluate LPIPS and DSIM for in-the-wild setting as discussed in the evaluation section and supplement.

Method	In the Wild		GT Pose			
	LPIPS↓	DSIM↓	PSNR↑	SSIM↑	LPIPS↓	DSIM↓
DRGS[23]	0.588	0.273	19.16	0.516	0.544	0.253
DNGS[9]	0.503	0.193	19.79	0.588	0.466	0.185
SparseGS[10]	0.309	0.105	21.37	0.667	0.260	0.093
FSGS[11]	0.327	0.098	<u>21.67</u>	0.637	0.394	0.128
SPARS3R[24]	<u>0.245</u>	<u>0.082</u>	21.48	0.674	<u>0.213</u>	<u>0.081</u>
Ours	0.238	0.080	22.39	0.702	0.211	0.072

Sparse Mip-NeRF 360 Dataset As shown in Table 2 and Fig 5, our approach demonstrates notable improvements over other 3DGS-based methods. DRGS and DNGS yield overly smooth renderings because of the suboptimal depth regularization. Although SparseGS and FSGS improve the rendering quality through floater pruning, score distillation regularization, and the densification strategy. Although SPARS3R renders more details, the global alignment can leave regions where points are incorrectly placed, manifested as artifacts due to the strong initialization bias. In comparison, MS-GS favors more accurate local regions, and our virtual view supervision further improves the results. As in Fig. 5, it preserves coherent geometry and reconstructs fine details, such as the table legs and carpet. These results show the efficacy of our semantic dense initialization in regularizing scene structure and facilitating the optimizations of the 3DGS framework.

Table 3: Quantitative Comparison on sparse unbounded drone dataset. Methods[†] renders each test view with the appearance embedding taken from the training image that is nearest in pose and shares the same appearance. Please refer to the setting described in Sec. A.1.2. Other methods extract appearance from the input image.

Method	Computation		In the Wild		GT Pose			
	GPU hrs.	FPS	LPIPS↓	DSIM↓	PSNR↑	SSIM↑	LPIPS↓	DSIM↓
NeRF-W [†] [14]	5.79	<1	0.659	0.451	16.95	0.453	0.621	0.402
Ha-NeRF [15]	11..66	<1	0.669	0.405	16.27	0.470	0.622	0.361
CR-NeRF [16]	7.31	<1	0.694	0.489	16.59	0.467	0.612	0.370
GS-W [12]	2.75	63	<u>0.446</u>	<u>0.213</u>	<u>17.33</u>	<u>0.491</u>	<u>0.487</u>	<u>0.279</u>
Wild-GS [13]	<u>1.01</u>	74	0.526	0.425	14.13	0.345	0.547	0.487
WildGaussians [†] [17]	2.57	<u>170</u>	0.502	0.302	15.60	0.388	0.546	0.428
Ours[†]	0.29	373	0.331	0.105	19.87	0.580	0.322	0.096

Table 4: Quantitative Comparison on sparse Phototourism dataset. Methods[‡] optimize appearance embedding on the left half of the test image and evaluate on the other half. Other methods extract appearance from the input image.

Method	Computation		In the Wild		GT Pose			
	GPU hrs.	FPS	LPIPS↓	DSIM↓	PSNR↑	SSIM↑	LPIPS↓	DSIM↓
NeRF-W [‡] [14]	8.28	<1	0.325	0.180	<u>17.93</u>	0.619	0.439	0.269
Ha-NeRF [15]	14.49	<1	0.310	0.167	16.33	0.663	0.446	<u>0.231</u>
CR-NeRF [16]	8.83	<1	0.299	<u>0.145</u>	16.98	<u>0.668</u>	0.422	<u>0.232</u>
GS-W [12]	2.32	57	<u>0.289</u>	<u>0.161</u>	16.74	<u>0.637</u>	<u>0.365</u>	0.245
Wild-GS [13]	<u>0.95</u>	70	0.331	0.216	16.21	0.592	0.385	0.330
WildGaussians [‡] [17]	2.71	<u>162</u>	0.317	0.195	14.33	0.577	0.433	0.248
Ours[‡]	0.32	351	0.258	0.138	18.99	0.684	0.269	0.154

Sparse unbounded drone and Phototourism Datasets Tables 3 and 4 together with Fig. 5 present results on these benchmarks. On the sparse unbounded-drone dataset, our approach significantly outperforms the SoTA methods with improvements of 2.54 dB in PSNR, 0.089 in SSIM, and cuts LPIPS and DSIM by 33.8% and 65.6%, respectively, with respect to the best prior method. The added capacity of U-Net in GS-W and Wild-GS is hard to train with sparse views and worsens the overfitting issue, manifesting as artifacts and blurred structure, e.g., the building surfaces. Without sufficient constraints, the appearance-affine head and uncertainty weighting in WildGaussians can absorb photometric error instead of correcting structures, leaving as off-view aliasing and texture drift. All these methods exhibit inconsistent appearances and floaters, e.g., the statue and the dome in Trevi Fountain and Brandenburg Gate. In contrast, MS-GS reconstructs a more coherent structure with fine-grained details and consistent appearance rendering, thanks to the synergy of our proposed components. Furthermore, our design is lightweight, requiring >3× less GPU time for training over Wild-GS and rendering at 300+ FPS.

5 Limitations

First, MS-GS is not designed for handling transient objects, which is especially difficult under sparse views due to increased uncertainty and ambiguities in scene reconstruction. While recent methods leverage uncertainty masks to remove transients and allow other observations to fill in the blank, often no other observations exist under a sparse setting; therefore, the transient regions remain under-constrained. Insufficient learning of transient masks often leads to worse results. Second, generalizing to non-Lambertian surfaces is challenging and requires more complex modeling. As MS-GS targets 3D consistency between views, the specular highlights can be smoothed or averaged out (see Fig. 10). Combining our framework with techniques, such as explicitly modeling of light [38] and surface reconstruction [39], remains an open research area. Specific techniques have to be developed to solve these limitations, which we leave as future work.



Figure 5: Qualitative comparison of novel view synthesis across different datasets. MS-GS (ours) excels at capturing detailed structures and preserving consistent appearance.

6 Conclusion

MS-GS establishes a strong baseline for multi-appearance sparse-view 3D Gaussian Splatting, significantly improving over existing methods. We identify that one of the limitations of 3DGS-based methods in sparse-view synthesis is the sparse point cloud initialization. To address this, our proposed method constructs a dense point cloud by performing individual alignment and back-projection in local semantic regions. The geometric prior steers the 3DGS optimization and acts as the cornerstone for our multi-view geometry-guided supervision. We further introduce virtual views to provide supervision along newly created camera rays as self-regularization to suppress floaters and encourage consistency, which aligns with the fundamental constraint of 3D reconstruction. Jointly, MS-GS offers a robust solution under challenges of limited viewpoints and varying appearances that naturally arise in real-world data.

7 Acknowledgement

This research is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA R&D Contract No. 140D0423C0076. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. This research and/or curriculum was supported by grants from NVIDIA and utilized NVIDIA RTX A5500 and A100 GPUs.

References

- [1] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.
- [2] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.
- [3] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5885–5894, 2021.
- [4] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5480–5490, 2022.
- [5] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12882–12891, 2022.
- [6] Guangcong Wang, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9065–9076, 2023.
- [7] Jiawei Yang, Marco Pavone, and Yue Wang. Freenerf: Improving few-shot neural rendering with free frequency regularization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8254–8263, 2023.
- [8] Mijeong Kim, Seonguk Seo, and Bohyung Han. Infonerf: Ray entropy minimization for few-shot neural volume rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12912–12921, 2022.
- [9] Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20775–20785, 2024.
- [10] Haolin Xiong, Sairisheek Muttukuru, Rishi Upadhyay, Pradyumna Chari, and Achuta Kadambi. Sparsegs: Real-time 360 $\{\deg\}$ sparse view synthesis using gaussian splatting. *arXiv preprint arXiv:2312.00206*, 2023.
- [11] Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. Fsgs: Real-time few-shot view synthesis using gaussian splatting. In *European conference on computer vision*, pages 145–163. Springer, 2025.
- [12] Dongbin Zhang, Chuming Wang, Weitao Wang, Peihao Li, Minghan Qin, and Haoqian Wang. Gaussian in the wild: 3d gaussian splatting for unconstrained image collections. *arXiv preprint arXiv:2403.15704*, 2024.

- [13] Jiacong Xu, Yiqun Mei, and Vishal M Patel. Wild-gs: Real-time novel view synthesis from unconstrained photo collections. *arXiv preprint arXiv:2406.10373*, 2024.
- [14] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7210–7219, 2021.
- [15] Xingyu Chen, Qi Zhang, Xiaoyu Li, Yue Chen, Ying Feng, Xuan Wang, and Jue Wang. Hallucinated neural radiance fields in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12943–12952, 2022.
- [16] Yifan Yang, Shuhai Zhang, Zixiong Huang, Yubing Zhang, and Minghui Tan. Cross-ray neural radiance fields for novel-view synthesis from unconstrained image collections. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15901–15911, 2023.
- [17] Jonas Kulhanek, Songyou Peng, Zuzana Kukelova, Marc Pollefeys, and Torsten Sattler. Wildgaussians: 3d gaussian splatting in the wild. *arXiv preprint arXiv:2407.08447*, 2024.
- [18] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3), 2022.
- [19] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9492–9502, 2024.
- [20] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2024.
- [21] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.
- [22] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM siggraph 2006 papers*, pages 835–846. 2006.
- [23] Jaeyoung Chung, Jeongtaek Oh, and Kyoung Mu Lee. Depth-regularized optimization for 3d gaussian splatting in few-shot images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 811–820, 2024.
- [24] Yutao Tang, Yuxiang Guo, Deming Li, and Cheng Peng. Spars3r: Semantic prior alignment and regularization for sparse 3d reconstruction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26810–26821, 2025.
- [25] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision*, pages 71–91. Springer, 2024.
- [26] Kaiwen Jiang, Yang Fu, Mukund Varma T, Yash Belhe, Xiaolong Wang, Hao Su, and Ravi Ramamoorthi. A construct-optimize approach to sparse view synthesis without camera pose. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024.
- [27] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.
- [28] Ken Shoemake. Animating rotation with quaternion curves. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 245–254, 1985.
- [29] William R Mark, Leonard McMillan, and Gary Bishop. Post-rendering 3d warping. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 7–ff, 1997.

- [30] Leonard McMillan and Gary Bishop. Plenoptic modeling: An image-based rendering system. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 433–440. 2023.
- [31] Yiqun Mei, Jiacong Xu, and Vishal Patel. Regs: Reference-based controllable scene stylization with gaussian splatting. *Advances in Neural Information Processing Systems*, 37:4035–4061, 2024.
- [32] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [33] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5470–5479, 2022.
- [34] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4578–4587, 2021.
- [35] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4): 600–612, 2004.
- [36] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.
- [37] Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. Dreamsim: Learning new dimensions of human visual similarity using synthetic data. *arXiv preprint arXiv:2306.09344*, 2023.
- [38] Jia-Mu Sun, Tong Wu, Yong-Liang Yang, Yu-Kun Lai, and Lin Gao. Sol-nerf: Sunlight modeling for outdoor scene decomposition and relighting. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–11, 2023.
- [39] Kaiwen Jiang, Venkataram Sivaram, Cheng Peng, and Ravi Ramamoorthi. Geometry field splatting with gaussian surfels. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5752–5762, 2025.
- [40] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019.
- [41] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018.
- [42] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. Roma: Robust dense feature matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19790–19800, 2024.
- [43] Shuzhe Wang, Vincent Leroy, Yann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024.
- [44] John C Gower. Generalized procrustes analysis. *Psychometrika*, 40:33–51, 1975.
- [45] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–12, 2023.
- [46] Yuechen Zhang, Zexin He, Jinbo Xing, Xufeng Yao, and Jiaya Jia. Ref-npr: Reference-based non-photorealistic radiance fields for controllable scene stylization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4242–4251, 2023.

- [47] Kai Zhang, Nick Kolkin, Sai Bi, Fujun Luan, Zexiang Xu, Eli Shechtman, and Noah Snavely. Arf: Artistic radiance fields. In *European Conference on Computer Vision*, pages 717–733. Springer, 2022.
- [48] Chen Quei-An. Nerf pl: a pytorch-lightning implementation of nerf. URL https://github.com/kwea123/nerf_pl, 5, 2020.
- [49] Keyan Ding, Kede Ma, Shiqi Wang, and Eero P Simoncelli. Image quality assessment: Unifying structure and texture similarity. *IEEE transactions on pattern analysis and machine intelligence*, 44(5):2567–2581, 2020.
- [50] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.
- [51] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.
- [52] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2818–2829, 2023.
- [53] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: The abstract and introduction (Section 1) clearly state the paper's main contributions, including the MS-GS framework for multi-appearance sparse-view 3D Gaussian Splatting, the Semantic Depth Alignment approach using geometric priors from monocular depth, and the Geometry-guided Supervisions. These claims accurately reflect the detailed methodology (Section 3) and experimental results (Section 4).

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: We discuss the limitations in Section 5.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The paper details its methodology and mathematical formulations, such as those for 3D Gaussian Splatting (Section 4.1), Semantic Depth Alignment (Section 4.1, Equations 5-8), and the multi-view geometry-guided supervisions including appearance modeling and optimization (Section 4.2, Equations 9-12). These are presented with their underlying assumptions and components. Foundational concepts are also appropriately cited.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Section 5 provides detail on the datasets (Section 5.1), the proposed "in-the-wild" evaluation setting (Section 5.2), and implementation details including the base model and hyperparameters (Supplement).

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

- (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We will open-source the code and data upon acceptance.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Section 5 states the details of the training and test details.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report the average metrics for each dataset and note the sample size of each dataset.

Guidelines:

- The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: "Results are obtained using an NVIDIA RTX A5500 GPU" is stated in the supplement. Furthermore, Tables 3 and 4 include "Computation" columns detailing "GPU hrs." for training and "FPS" for rendering.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: We conform with the NeurIPS Code of Ethics when preparing our submission.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [No]

Justification: Our paper contributes to the wide-ranging applications in AR/VR, 3D site modeling, autonomous driving, robotics but the topic of our paper is not closely related to societal influence.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [No]

Justification: We will constrain our data for research use only but dataset is not the main theme in our paper.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All concepts, methods, datasets are appropriately cited as demonstrated across the entire paper.

Guidelines:

- The answer NA means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#)

Justification: We provide detailed description of our method and the dataset we provide in Section 4. More details can be found in Supplementary Materials.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[NA\]](#)

Justification: We do not have crowdsourcing experiments or experiments with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [\[NA\]](#)

Justification: IRB approval or an equivalent review would not be applicable as the research methodology does not involve human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: We do not use LLM in our core method.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

A Technical Appendices and Supplementary Material

A.1 Sparse unbounded drone dataset

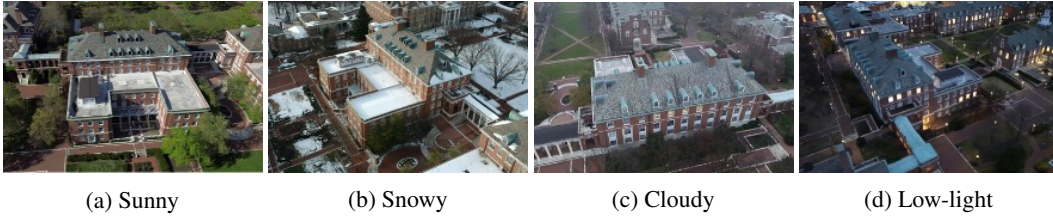


Figure 6: Dataset visualizations

A.1.1 Overview and registration

As shown in Fig. 6, our dataset consists of 4 unique appearances: sunny, snowy, cloudy, and low-light, captured in drone footage from different seasons and time of the day. We sampled 5 images per appearance evenly from the circular camera trajectory, resulting in 20 training images, and similarly for 12 testing images. We used hloc [40] registration pipeline with SuperPoint features [41] and RoMa [42] matcher.

A.1.2 Comparison with prior benchmark

To obtain the appearance embeddings of a test image from the Phototourism [22] dataset, prior setups either optimize on half of the test image or use a network to extract features from the entire test image. Both of these approaches involve the use of test images during evaluation. For our dataset, the multi-view, multi-appearance setup enables the correct appearance to be rendered based on metadata/timestamp, not pixel-wise information from the test image. For example, a snowy test image can be rendered with the appearance embedding of a snowy training image by relating their timestamps. This setup is also faster than optimizing half of the test image. Additionally, our dataset contains scenes with 360-degree coverage by perspective cameras, whereas Phototourism is covered by face-forward images.

A.2 Experiments and visualizations

A.2.1 Initialization comparisons

Table 5: Experiments on the effectiveness of different initializations. The metrics are reported as the average on the Sparse Mip-NeRF 360 dataset.

Method	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	DSIM \downarrow
sparse init.	20.83	0.627	0.267	0.109
image-level alignment	21.06	0.631	0.253	0.094
semantic alignment	21.96	0.690	0.216	0.080
DUST3R [43]	19.89	0.585	0.270	0.118

In this section, we compare different initialization strategies, namely the sparse SfM point cloud, dense initialization with image-level alignment as introduced in Section 3.1, our proposed dense initialization with semantic alignment, and DUST3R [43] point cloud. In Figure 7, the sparse SfM point cloud contains only a few thousand points and covers a small fraction of the scene. Even though initialization with image-level alignment is much denser, it also introduces more errors in the point cloud, leading to noisy structures. In contrast, our method, which favors more accurate local alignments, achieves cleaner and semantically meaningful scene components.

Quantitatively (Table 5), initialization from image-level alignment offers only marginal benefit compared to the baseline, as misplaced Gaussians that are not pruned or densified correctly can produce noisy structures, as shown in Fig. 8. DUST3R is a two-view pointmap estimator. When the number of images is greater than two, it aggregates all pairwise pointmap predictions into a very

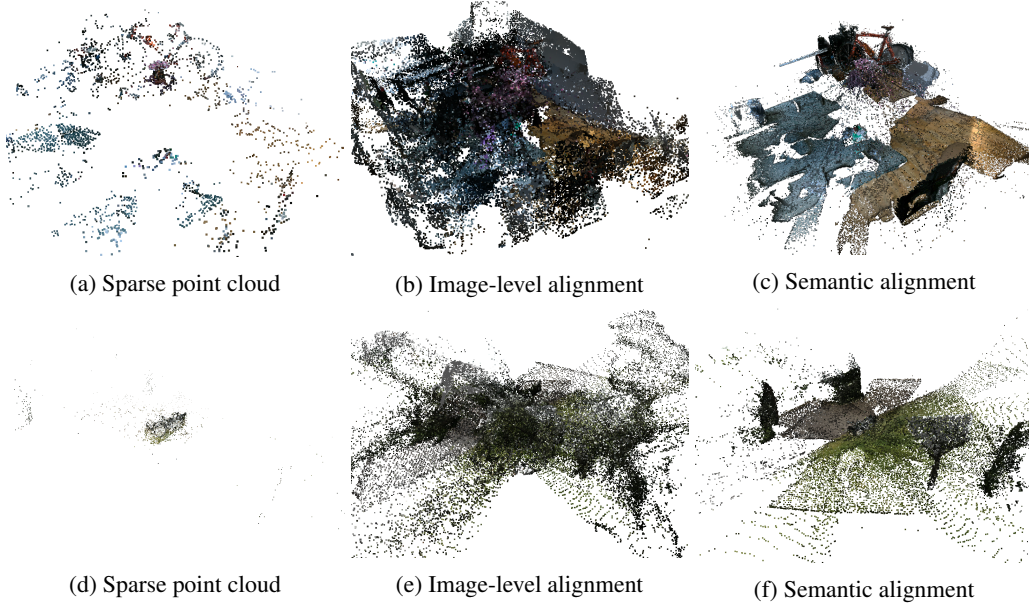


Figure 7: Visualizations of different point cloud initializations.

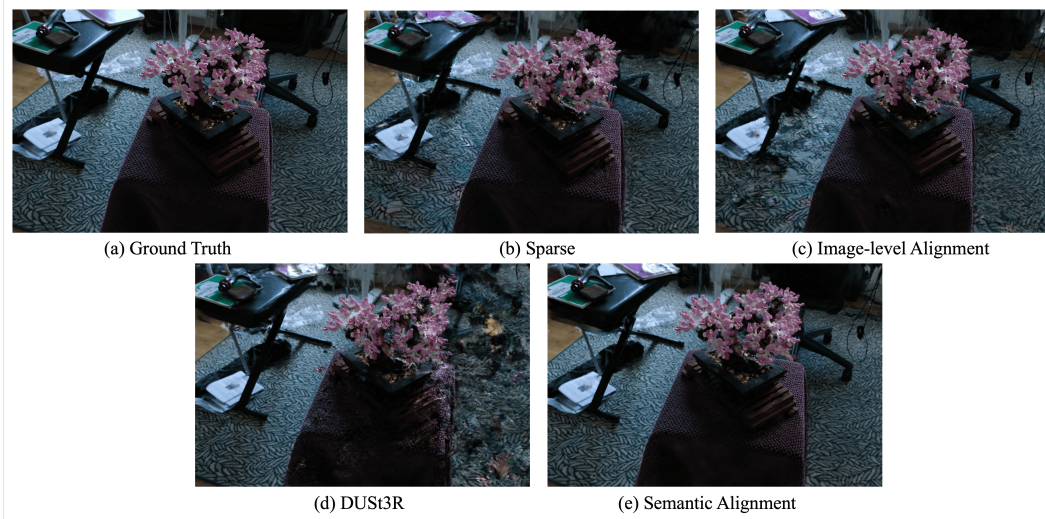


Figure 8: Visualizations of rendering with different point cloud initializations.

dense point cloud, usually millions of points. To utilize DUST3R points, we align them to the SfM points based on corresponding pixels using Procrustes Alignment [44]. While the output of DUST3R is visually pleasing, it still suffers from depth ambiguities, leading to incorrect placement of objects. As shown in Fig. 8, it produces ghosting artifacts due to the strong initialization bias. Notably, our approach improves the PSNR by 1.13, SSIM by 0.063, LPIPS by 0.051, and DSIM by 0.029. In addition, the time complexity of DUST3R to run N images is $\mathcal{O}(N^2)$ compared to $\mathcal{O}(N)$ for the monocular depth estimator, which makes it harder to scale. This analysis highlights the importance of semantic depth alignment, which guides 3DGS to converge to a better scene reconstruction.

We also validate the semantic alignment accuracy on the bonsai scene. Specifically, we use the Multi-View Stereo (MVS) depth from dense views as GT depth estimates. An image-level estimate of scale results in a mean absolute error of 1.94, and our per-semantic-region scaling reduces this to 1.07. The error maps are visualized in Fig. 9, which shows that the depth continuities around object boundaries exhibit higher error, and the error inside objects is lower/darker with our piece-wise alignment approach.

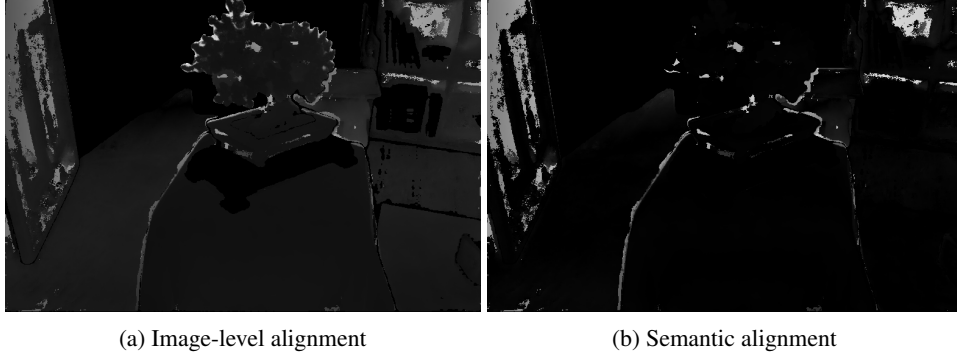


Figure 9: Error maps after alignment. Brighter means higher error.

A.2.2 Edge cases

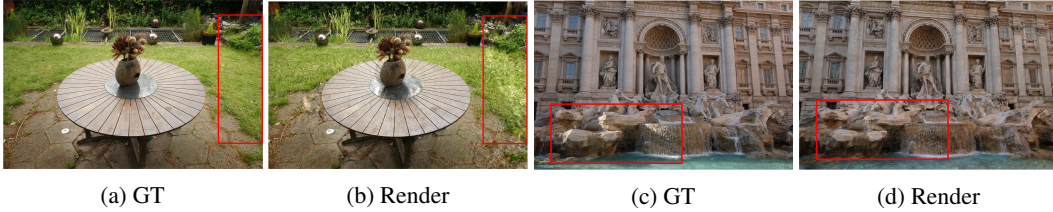


Figure 10: As MS-GS favors more accurate local alignment, areas without dense initialization can introduce artifacts in (a) and (b). Specular highlights can be smoothed out due to the multi-view consistency and limited capacity of appearance embedding, as seen in (c) and (d).

A.3 Implementation details

We develop MS-GS based on the 3DGS implementation from NeRFStudio, called Splatfacto [45]. The baseline introduced in our ablation study Section 4.3 uses the same Splatfacto model. In Semantic Depth Alignment, the minimum number of SfM points threshold within a valid mask is 10. The intersection of two masks for merging is 0.7. We use both back-projected point cloud and MVS points for our initialization. The appearance MLP consists of 3 layers of 64 hidden units. The embedding sizes for the Gaussian feature and per-image appearance embeddings are 16 and 32, respectively. Virtual views are generated by interpolating toward one of the $k=4$ nearest training cameras. We use features extracted from blocks 3 and 4 of VGG-16 [32, 46, 47] for feature loss at different resolutions and receptive fields. We set $\lambda_I = 0.8$, $\lambda_{\text{pix}} = 1.0$, and $\lambda_{\text{feat}} = 0.04$. The total number of training iterations is 16,500, with the geometry-guided supervision enabled after 15,000 iterations. The same hyperparameters are maintained throughout the experiments. Results are obtained with the NVIDIA RTX A5500 GPU.

A.4 Appearance embedding initialization

Table 6: Experiments on the appearance embedding initializations. The metrics are reported as the average on the Sparse Unbounded Drone dataset.

Method	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	DSIM \downarrow
normal distribution in $[0,1]$	18.77	0.524	0.352	0.132
near-zero initialization	19.29	0.538	0.336	0.115

Appearance embeddings are typically initialized with a normal distribution in $[0, 1]$ [48, 14]. We find that this initialization introduces view-specific biases. Instead, we initialize them near zero, i.e., uniform distribution in $[-1 \times 10^{-4}, 1 \times 10^{-4}]$, which shows improved metrics and yields meaningful clusters after training, as shown in Table 6 and Fig. 11. We attribute this result to the near-zero

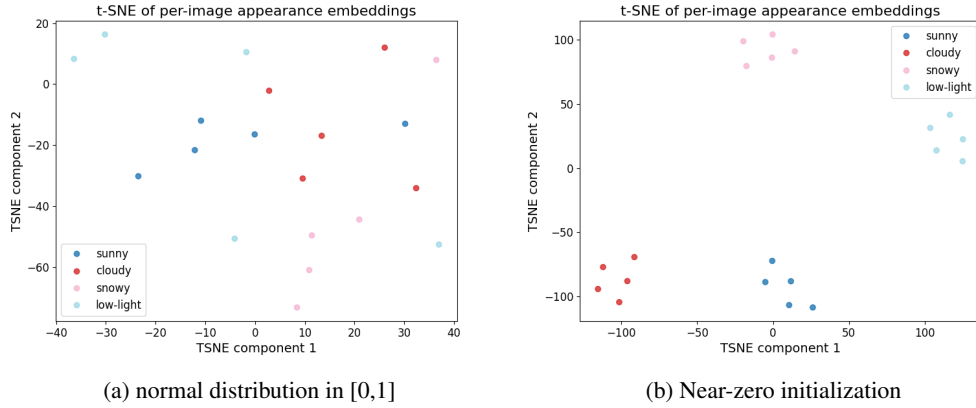


Figure 11: t-SNE visualizations of per-image appearance embeddings after training with different initializations

initialization: it delays the expressive power of the per-image appearance embeddings, minimally influencing the MLP training in the early stages, so the network first learns a shared color basis and later allocates capacity to disentangle appearances.

A.4.1 Sparser setting

Table 7: Experiments in 12-view setting, where each appearance has 3 images. MS-GS continues to outperform other methods. The metrics are reported as the average on the Sparse unbounded drone dataset.

Method	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	DSIM \downarrow
GS-W [12]	14.83	0.371	0.560	0.457
Wild-GS [13]	13.66	0.289	0.587	0.583
WildGaussians [17]	12.47	0.278	0.612	0.664
Ours	17.78	0.477	0.412	0.180

A.4.2 Dense init. for other in-the-wild methods

In this section, we investigate the performance of other in-the-wild methods using our proposed dense initialization. Based on Table 8, on one hand, all methods achieve significantly better metrics compared to their original baseline. For example, GS-W gains 0.9 dB in PSNR, 0.054 in SSIM, and reduces 0.11 in LPIPS and 0.069 in DSIM. This experiment confirms that our initialization is a drop-in enhancement for 3DGS-based pipeline. On the other hand, the improved performance of other methods is still inferior to MS-GS by a large margin, validating the effectiveness of our appearance modules and multi-view geometry-guided supervision in this challenging setting.

Table 8: Experiments on the effectiveness of our dense initialization applied to other methods for multi-appearance synthesis. The metrics are reported as the average on the Sparse unbounded drone dataset.

Method	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	DSIM \downarrow
GS-W [12]	17.33	0.491	0.487	0.279
+ dense init.	18.23	0.545	0.377	0.210
Wild-GS [13]	14.13	0.345	0.547	0.487
+ dense init.	14.35	0.395	0.544	0.443
WildGaussians [17]	15.60	0.388	0.546	0.428
+ dense init.	16.50	0.449	0.482	0.316
Ours	19.87	0.580	0.322	0.096

A.5 Semantic alignment algorithm

Algorithm 1: Semantic Masks Prediction

Input : Image I_n , a set of visible 2D SfM points \mathcal{X} on I_n , segmentation model \mathcal{S} , threshold TH_{sfm} , threshold TH_{IoU} .

Output : Final set of masks M_{final} .

```

1 Def append_mask( $M_i, M_{final}, TH_{IoU}$ ):
2   merged = False;
3   for  $M \in M_{final}$  do
4     if  $M_i \cap M > TH_{IoU}$  then
5        $M = M \cup M_i$ ;                                /* Merge the masks */
6       merged = True;
7       break;
8     end
9   end
10  if not merged then
11     $M_i \rightarrow M_{final}$ ;                            /* Append the mask to set */
12  end
13   $M_{final} = \emptyset$ ;
14  while  $\mathcal{X}$  is not empty do
15     $x_i \sim \mathcal{X}$ ;                                       /* Sample a point */
16     $M_i = \mathcal{S}(x_i, I_n)$ ;                             /* Prompt a mask */
17     $x_{m,i} = \mathcal{X} \cap M_i$ ;                         /* Find points within the mask */
18    if  $|x_{m,i}| > TH_{sfm}$  then
19      append_mask( $M_i, M_{final}, TH_{IoU}$ );           /* Enough points */
20    else
21       $M_i = \mathcal{S}(x_{m,i}, I_n)$ ;                       /* Re-prompt with points within the mask */
22       $x_{m,i} = \mathcal{X} \cap M_i$ ;
23      if  $|x_{m,i}| > TH_{sfm}$  then
24        append_mask( $M_i, M_{final}, TH_{IoU}$ );
25      else
26        continue;
27      end
28    end
29    Exclude  $x_{m,i}$  from  $\mathcal{X}$ ;                            /* Remove points from set */
30  end

```

The iterative refinement algorithm is detailed in Algorithm 1. This is an automatic process to find the semantic masks anchored by SfM points, which are back-projected individually to form a dense point cloud for 3DGS initialization.

A.6 In-the-wild evaluation

Sparse-view and multi-appearance registration is challenging because of limited overlap and view inconsistency; Fewer reliable feature matches result in suboptimal pose estimation and point triangulation. Sparse-view methods [34, 3, 4, 6, 7, 8, 5, 9] commonly assume ground-truth camera poses, i.e., calibration from dense views. However, only training views should be available in an in-the-wild setting. 3DGS-based methods rely heavily on the SfM point cloud, further necessitating the separation of training and testing views from the registration stage. A previous approach [11] has tried to perform re-triangulation based on known train poses, but does not account for pose inaccuracy. Therefore, we propose a coordinate alignment method, illustrated in Fig. 12, to disentangle training and testing images in registration.

A.6.1 Coordinate alignment

In coordinate alignment, we seek to register training and testing views separately and align them together. Therefore, we perform two registrations: 1. training images only, resulting in the input

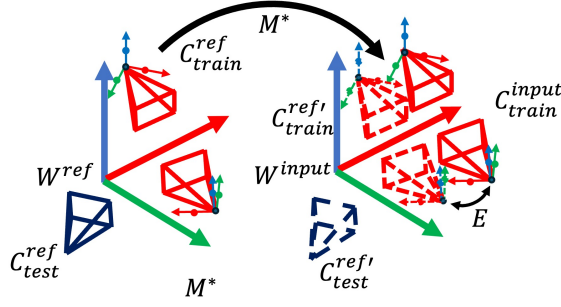


Figure 12: Illustration of Coordinate Alignment. We first compute the transformation M^* between train cameras in two coordinate systems $C_{\text{train}}^{\text{ref}}$ and $C_{\text{train}}^{\text{input}}$; each camera corresponds to 4 points: one position and three rotation points, displayed as small black, red, green, and blue points in the figure. The transformed $C_{\text{train}}^{\text{ref}}$ is denoted as $C_{\text{train}}^{\text{ref}'}$ in dashed lines, which is used to compute the error E between $C_{\text{train}}^{\text{input}}$. Finally, M^* transforms test camera poses $C_{\text{test}}^{\text{ref}}$ to $C_{\text{test}}^{\text{ref}'}$ in the input coordinate system.

coordinate system $C_{\text{train}}^{\text{input}}$. 2. training and testing images in the reference coordinate system, $C_{\text{train}}^{\text{ref}}$ and $C_{\text{test}}^{\text{ref}}$, as SfM reconstructs the scene in a different coordinate system each time. A transformation M^* is computed between $C_{\text{train}}^{\text{input}}$ and $C_{\text{train}}^{\text{ref}}$ using Procrustes Alignment [44] to transform test cameras $C_{\text{test}}^{\text{ref}}$ to the input coordinate system $C_{\text{test}}^{\text{ref}'}$. Conventionally, only camera positions/centers are considered during alignment. To leverage rotation information, we additionally sample three points along the camera’s local rotation axes to form a small local frame around each camera center. Formally, the point set of each camera, represented as $P_{\text{cam}} \in \mathbb{R}^{4 \times 3}$, is defined as:

$$\begin{aligned} R &= [r_x, r_y, r_z] \in \mathbb{R}^{3 \times 3}, \\ s &= \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2}, \\ P_{\text{cam}} &= \begin{bmatrix} T^\top \\ (T + s r_x)^\top \\ (T + s r_y)^\top \\ (T + s r_z)^\top \end{bmatrix} \in \mathbb{R}^{4 \times 3}, \end{aligned} \quad (9)$$

where R is the camera rotation matrix, $T \in \mathbb{R}^3$ is the translation (camera center), and s is a scalar scaling factor approximated by the per-dimension standard deviation σ . Each row of P_{cam} corresponds to a 3D point: the camera center followed by three axis-offset points derived from the camera’s orientation. Finally, we use $C_{\text{train}}^{\text{input}}$, $C_{\text{test}}^{\text{ref}'}$, and points triangulated from $C_{\text{train}}^{\text{input}}$ for 3DGS input. In this way, we simulate the real-world scenario, where camera poses and 3D points are estimated from training views while having the testing camera poses in the same coordinate system for evaluation.

As shown in Table 9, our rotation-aware alignment reduces the rotation error E_R , in degrees, by more than 10 times and the position error E_T , in an arbitrary unit as in the SfM, by 4 times. This improvement results in accurately aligned test cameras and, consequently, more reliable evaluations.

Table 9: Experiments on the effectiveness of our rotation-aware camera alignment. The metrics are reported as the average on the Sparse Unbounded Drone dataset.

Method	$E_R(\text{med})$	$E_R(\mu)$	$E_T(\text{med})$	$E_T(\mu)$
w/o rotation points	0.791	0.793	0.0397	0.0377
ours	0.063	0.066	0.0067	0.0085

A.6.2 Evaluation metrics

We evaluate the novel view rendering quality based on the image and perceptual metrics, including PSNR, SSIM [35], and LPIPS [36]. We also propose to use DreamSim (DSIM) [37] as an additional

metric, which is an ensemble method of different perceptual metrics [36, 49, 50, 51, 52, 53] fine-tuned for human visual perspectives.

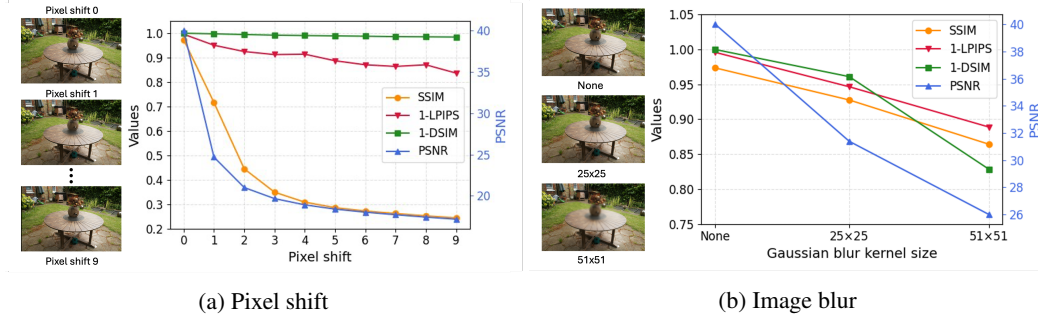


Figure 13: Evaluation of DSIM as metric

Our coordinate alignment method is accurate but not perfect, leaving small residual pose shifts. However, this slight pixel offset should not reflect a significant difference in metrics, dominating the quality assessment. As Fig. 13a shows, PSNR and SSIM drop steeply with a few pixel offsets, whereas DSIM remains almost flat. When images are dissimilar, where we add a blob of Gaussian blur at different kernel sizes in Fig. 13b to simulate semi-transparent Gaussians, DSIM shows a consistent decline as other metrics. This analysis indicates that DSIM is an appropriate metric for in-the-wild evaluations: it avoids over-penalising inevitable alignment errors while still capturing real perceptual degradation.