GaussianObject: High-Quality 3D Object Reconstruction from Four Views with Gaussian Splatting

CHEN YANG*, MoE Key Lab of Artificial Intelligence, AI Institute, SJTU, China
SIKUANG LI*, MoE Key Lab of Artificial Intelligence, AI Institute, SJTU, China
JIEMIN FANG†, Huawei Inc., China
RUOFAN LIANG, University of Toronto, Canada
LINGXI XIE, Huawei Inc., China
XIAOPENG ZHANG, Huawei Inc., China
WEI SHEN‡, MoE Key Lab of Artificial Intelligence, AI Institute, SJTU, China
QI TIAN, Huawei Inc., China

(a) 4 Input Views (b) Ground Truth (c) DietNeRF (d) FreeNeRF (e) 3DGS (f) Ours

Fig. 1. We introduce GaussianObject, a framework capable of reconstructing high-quality 3D objects from only 4 images with Gaussian splatting. GaussianObject demonstrates superior performance over previous state-of-the-art (SOTA) methods on challenging objects.

Reconstructing and rendering 3D objects from highly sparse views is of critical importance for promoting applications of 3D vision techniques and improving user experience. However, images from sparse views only contain very limited 3D information, leading to two significant challenges: 1) Difficulty in building multi-view consistency as images for matching are too few; 2) Partially omitted or highly compressed object information as view coverage is insufficient. To tackle these challenges, we propose GaussianObject, a framework to represent and render the 3D object with Gaussian splatting that achieves high rendering quality with only 4 input images. We first introduce techniques of visual hull and floater elimination, which explicitly inject structure priors into the initial optimization process to help build multi-view consistency, yielding a coarse 3D Gaussian representation. Then we construct a Gaussian repair model based on diffusion models to supplement the omitted object information, where Gaussians are further refined. We design a self-generating strategy to obtain image pairs for training the repair model. We further design a COLMAP-free variant, where pre-given accurate camera poses are not required, which achieves competitive quality and facilitates wider applications. GaussianObject is evaluated on several challenging datasets, including MipNeRF360, OmniObject3D, OpenIllumination, and our-collected unposed images, achieving superior performance from only four views and significantly outperforming previous SOTA methods. Our demo is available at *https://gaussianobject.github.io/*, and the code has been released at *https://github.com/GaussianObject/GaussianObject*.

CCS Concepts: • **Computing methodologies** → **Reconstruction**; **Rendering**; **Point-based models**.

Additional Key Words and Phrases: Sparse view reconstruction, 3D Gaussian Splatting, ControlNet, Visual hull, Novel view synthesis

ACM Reference Format:
Chen Yang, Sikuang Li, Jiemin Fang, Ruofan Liang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. 2024. GaussianObject: High-Quality 3D Object

*Equal contributions.
†Project lead.
‡Corresponding author.

Authors' addresses: Chen Yang, MoE Key Lab of Artificial Intelligence, AI Institute, SJTU, Shanghai, China, ycyangchen@sjtu.edu.cn; Sikuang Li, MoE Key Lab of Artificial Intelligence, AI Institute, SJTU, Shanghai, China, uranusits@sjtu.edu.cn; Jiemin Fang, Huawei Inc., Wuhan, China, jaminfong@gmail.com; Ruofan Liang, University of Toronto, Toronto, Canada, ruofan@cs.toronto.edu; Lingxi Xie, Huawei Inc., Beijing, China, 198808xc@gmail.com; Xiaopeng Zhang, Huawei Inc., Shanghai, China, zxphistory@gmail.com; Wei Shen, MoE Key Lab of Artificial Intelligence, AI Institute, SJTU, Shanghai, China, wei.shen@sjtu.edu.cn; Qi Tian, Huawei Inc., Shenzhen, China, tian.qi1@huawei.com.

2024. ACM 0730-0301/2024/12-ART
https://doi.org/10.1145/3687759

Reconstruction from Four Views with Gaussian Splatting. *ACM Trans. Graph.* 43, 6 (December 2024), 28 pages. https://doi.org/10.1145/3687759

1 INTRODUCTION

Reconstructing and rendering 3D objects from 2D images has been a long-standing and important topic, which plays critical roles in a vast range of real-life applications. One key factor that impedes users, especially ones without expert knowledge, from widely using these techniques is that usually dozens of multi-view images need to be captured, which is cumbersome and sometimes impractical. Efficiently reconstructing high-quality 3D objects from highly sparse captured images is of great value for expediting downstream applications such as 3D asset creation for game/movie production and AR/VR products.

In recent years, a series of methods [Guangcong et al. 2023; Jain et al. 2021; Niemeyer et al. 2022; Shi et al. 2024b; Song et al. 2023b; Yang et al. 2023; Zhou and Tulsiani 2023; Zhu et al. 2024] have been proposed to reduce reliance on dense captures. However, it is still challenging to produce high-quality 3D objects when the views become **extremely sparse**, *e.g.* only 4 images in a 360° range, as shown in Fig. 1. We delve into the task of sparse-view reconstruction and discover two main challenges behind it. The first one lies in the difficulty of building multi-view consistency from highly sparse input. The 3D representation is easy to overfit the input images and degrades into fragmented pixel patches of training views without reasonable structures. The other challenge is that with sparse captures in a 360° range, some content of the object can be inevitably omitted or severely compressed when observed from extreme views[1]. The omitted or compressed information is impossible or hard to be reconstructed in 3D only from the input images.

To tackle the aforementioned challenges, we introduce GaussianObject, a novel framework designed to reconstruct high-quality 3D objects from as few as 4 input images. We choose 3D Gaussian splatting (3DGS) [Kerbl et al. 2023] as the basic representation as it is fast and, more importantly, explicit enough. Benefiting from its point-like structure, we design several techniques for introducing object structure priors, *e.g.* the basic/rough geometry of the object, to help build multi-view consistency, including visual hull [Laurentini 1994] to locate Gaussians within the object outline and floater elimination to remove outliers. To erase artifacts caused by omitted or highly compressed object information, we propose a Gaussian repair model driven by 2D large diffusion models [Rombach et al. 2022], translating corrupted rendered images into high-fidelity ones. As normal diffusion models lack the ability to repair corrupted images, we design self-generating strategies to construct image pairs to tune the diffusion models, including rendering images from leave-one-out training models and adding 3D noises to Gaussian attributes. Images generated from the repair model can be used to refine the 3D Gaussians optimized with structure priors, where the rendering quality can be further improved. To further extend GaussianObject to practical applications, we introduce a COLMAP-free variant of GaussianObject (CF-GaussianObject), which achieves competitive reconstruction performance on challenging datasets with only four input images without inputting accurate camera parameters.

Our contributions are summarized as follows:

- We optimize 3D Gaussians from highly sparse views using explicit structure priors, introducing techniques of visual hull for initialization and floater elimination for training.
- We propose a Gaussian repair model based on diffusion models to remove artifacts caused by omitted or highly compressed information, where the rendering quality can be further improved.
- The overall framework GaussianObject consistently outperforms current SOTA methods on several challenging real-world datasets, both qualitatively and quantitatively. A COLMAP-free variant is further presented for wider applications, weakening the requirement of accurate camera poses.

2 RELATED WORK

Vanilla NeRF struggles in sparse settings. Techniques like Deng et al. [2022]; Roessle et al. [2022]; Somraj et al. [2024, 2023]; Somraj and Soundararajan [2023] use Structure from Motion (SfM) [Schönberger and Frahm 2016] derived visibility or depth and mainly focus on closely aligned views. Xu et al. [2022] uses ground truth depth maps, which are costly to obtain in real-world images. Some methods [Guangcong et al. 2023; Song et al. 2023b] estimate depths with monocular depth estimation models [Ranftl et al. 2021, 2022] or sensors, but these are often too coarse. Jain et al. [2021] uses a vision-language model [Radford et al. 2021] for unseen view rendering, but the semantic consistency is too high-level to guide low-level reconstruction. Shi et al. [2024b] combines a deep image prior with factorized NeRF, effectively capturing overall appearance but missing fine details in input views. Priors based on information theory [Kim et al. 2022], continuity [Niemeyer et al. 2022], symmetry [Seo et al. 2023], and frequency regularization [Song et al. 2023a; Yang et al. 2023] are only effective for specific scenarios, limiting their further applications. Besides, there are some methods [Jang and Agapito 2024; Jiang et al. 2024; Xu et al. 2024c; Zou et al. 2024] that employ Vision Transformer (ViT) [Dosovitskiy et al. 2021] to reduce the requirements for constructing NeRFs and Gaussians.

The recent progress in diffusion models has spurred notable advancements in 3D applications. Dreamfusion [Poole et al. 2023] proposes Score Distillation Sampling (SDS) for distilling NeRFs with 2D priors from a pre-trained diffusion model for 3D object generation from text prompts. It has been further refined for text-to-3D [Chen et al. 2023; Lin et al. 2023; Metzer et al. 2023; Shi et al. 2024a; Tang et al. 2024b; Wang et al. 2023a,b; Yi et al. 2024] and 3D/4D editing [Haque et al. 2023; Shao et al. 2024] by various studies, demonstrating the versatility of 2D diffusion models in 3D contexts. Burgess et al. [2024]; Chan et al. [2023]; Liu et al. [2023c]; Müller et al. [2024]; Pan et al. [2024]; Zhu and Zhuang [2024] have adapted these methods for 3D generation and view synthesis from a single image, while they often have strict input requirements and can produce overly saturated images. In sparse reconstruction, approaches like DiffusioNeRF [Wynn and Turmukhambetov 2023], SparseFusion [Zhou and Tulsiani 2023], Deceptive-NeRF [Liu et al. 2023b], ReconFusion [Wu et al. 2024] and CAT3D [Gao et al. 2024] integrate diffusion models with NeRFs. Recently, Large reconstruction

[1]When the view is orthogonal to the surface of the object, the observed information attached to the surface can be largely preserved; On the contrary, the information will be severely compressed.

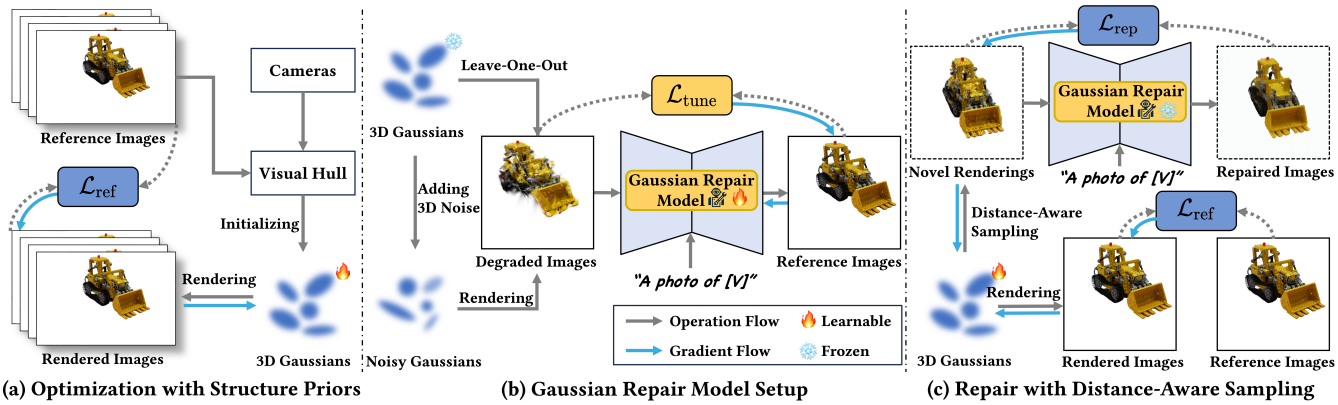

Fig. 2. Overview of GaussianObject. (a) We initialize 3D Gaussians by constructing a visual hull with camera parameters and masked images, which are optimized with $\mathcal{L}_{\text{ref}}$ and refined through floater elimination. (b) We use a novel 'leave-one-out' strategy and add 3D noise to Gaussians to generate corrupted Gaussian renderings. These renderings, paired with their corresponding reference images, facilitate the training of the Gaussian repair model employing $\mathcal{L}_{\text{tune}}$. For details please refer to Fig. 3. (c) Once trained, the Gaussian repair model is frozen and used to correct views that need to be rectified. These views are identified through distance-aware sampling. The repaired images and reference images are used to further optimize 3D Gaussians with $\mathcal{L}_{\text{rep}}$ and $\mathcal{L}_{\text{ref}}$.

models (LRMs) [Hong et al. 2024; Li et al. 2024; Tang et al. 2024a; Wang et al. 2024b; Wei et al. 2024; Weng et al. 2023; Xu et al. 2024a,b; Zhang et al. 2024] also achieve 3D reconstruction from highly sparse views. Though effective in generating images fast, these methods encounter issues with large pretraining, strict requirements on view distribution and object location, and difficulty in handling real-world captures.

While 3DGS shows strong power in novel view synthesis, it struggles with sparse 360° views similar to NeRF. Inspired by few-shot NeRFs, methods [Charatan et al. 2024; Chung et al. 2023; Paliwal et al. 2024; Xiong et al. 2023; Zhu et al. 2024] have been developed for sparse 360° reconstruction. However, they still severely rely on the SfM points. Our GaussianObject proposes structure-prior-aided Gaussian initialization to tackle this issue, drastically reducing the required input views to only 4, a significant improvement compared with over 20 views required by FSGS [Zhu et al. 2024].

3 METHOD

The subsequent sections detail the methodology: Sec. 3.1 reviews foundational techniques; Sec. 3.2 introduces our overall framework; Sec. 3.3 describes how we apply the structure priors for initial optimization; Sec. 3.4 details the setup of our Gaussian repair model; Sec. 3.5 illustrates the repair of 3D Gaussians using this model and Sec. 3.6 elucidates the COLMAP-free version of GaussianObject. To facilitate a better understanding, all key mathematical symbols and their corresponding meanings are listed in Table 1.

3.1 Preliminary

3D Gaussian Splatting. 3D Gaussian Splatting [Kerbl et al. 2023] represents 3D scenes with 3D Gaussians. Each 3D Gaussian is composed of the center location μ, rotation quaternion q, scaling vector s, opacity σ, and spherical harmonic (SH) coefficients sh. Thus, a scene is parameterized as a set of Gaussians $\mathcal{G} = \{G_i : \mu_i, q_i, s_i, \sigma_i, sh_i\}_{i=1}^{P}$.

ControlNet. Diffusion models are generative models that sample from a data distribution $q(X_0)$, beginning with Gaussian noise ϵ and using various sampling schedulers. They operate by reversing a discrete-time stochastic noise addition process $\{X_t\}_{t=0}^{T}$ with a diffusion model $p_\theta(X_{t-1}|X_t)$ trained to approximate $q(X_{t-1}|X_t)$, where $t \in [0, T]$ is the noise level and θ is the learnable parameters. Substituting X_0 with its latent code Z_0 from a Variational Autoencoder (VAE) [Kingma and Welling 2014] leads to the development of Latent Diffusion Models (LDM) [Rombach et al. 2022]. ControlNet [Zhang et al. 2023a] further enhances the generative process with additional

Table 1. List of Key Mathematical Symbols

Symbol	Meaning
$X^{\text{ref}} = \{x_i\}_{i=1}^{N}$	Reference images
$K^{\text{ref}} = \{k_i\}_{i=1}^{N}$	Intrinsics of X^{ref}
$\hat{K}^{\text{ref}}$	Estimated intrinsics of X^{ref}
$\hat{K}$	Estimated shared intrinsics of X^{ref}
$\Pi^{\text{ref}} = \{\pi_i\}_{i=1}^{N}$	Extrinsics of X^{ref}
Π^{nov}	Extrinsics of viewpoints in repair path
$\hat{\Pi}^{\text{ref}}$	Estimated extrinsics of X^{ref}
$M^{\text{ref}} = \{m_i\}_{i=1}^{N}$	Masks of X^{ref}
μ	Center location of Gaussian
q	Rotation quaternion of Gaussian
s	Scale vector of Gaussian
σ	Opacity of Gaussian
sh	Spherical harmonic coefficients of Gaussian
$\mathcal{G}_c$	Coarse 3D Gaussians
$\mathcal{R}$	Diffusion based Gaussian repair model
$\mathcal{E}$	Latent diffusion encoder of $\mathcal{R}$
$\mathcal{D}$	Latent diffusion decoder of $\mathcal{R}$
x'	Degraded rendering
$\hat{x}$	Image repaired by $\mathcal{R}$
ϵ_s	3D Noise added to attributes of $\mathcal{G}_c$
ϵ	2D Gaussian noise for fine-tuning
ϵ_θ	2D Noise predicted by $\mathcal{R}$
c^{tex}	Object-specific language prompt
$\mathcal{P}$	Coarse point cloud predicted by DUSt3R

image conditioning by integrating a network structure similar to the diffusion model, optimized with the loss function:

$$\mathcal{L}_{Cond} = \mathbb{E}_{Z_0,t,\epsilon}[\|\epsilon_\theta(\sqrt{\bar{\alpha}_t}Z_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t, c^{\text{tex}}, c^{\text{img}}) - \epsilon\|_2^2], \quad (1)$$

where c^{tex} and c^{img} denote the text and image conditioning respectively, and ϵ_θ is the Gaussian noise inferred by the diffusion model with parameter θ, $\bar{\alpha}_{1:T} \in (0,1]^T$ is a decreasing sequence associated with the noise-adding process.

3.2 Overall Framework

Given a sparse collection of N reference images $X^{\text{ref}} = \{x_i\}_{i=1}^N$, captured within a 360° range and encompassing one object, along with the corresponding camera intrinsics[2] $K^{\text{ref}} = \{k_i\}_{i=1}^N$, extrinsics $\Pi^{\text{ref}} = \{\pi_i\}_{i=1}^N$ and masks $M^{\text{ref}} = \{m_i\}_{i=1}^N$ of the object, our target is to obtain a 3D representation $\mathcal{G}$, which can achieve photo-realistic rendering $x = \mathcal{G}(\pi|\{x_i, \pi_i, m_i\}_{i=1}^N)$ from any viewpoint. To achieve this, we employ the 3DGS model for its simplicity for structure priors embedding and fast rendering capabilities. The process begins with initializing 3D Gaussians using a visual hull [Laurentini 1994], followed by optimization with floater elimination, enhancing the structure of Gaussians. Then we design self-generating strategies to supply sufficient image pairs for constructing a Gaussian repair model, which is used to rectify incomplete object information. The overall framework is shown in Fig. 2.

3.3 Initial Optimization with Structure Priors

Sparse views, especially for only 4 images, provide limited 3D information for reconstruction. In this case, SfM points, which are the key for 3DGS initialization, are often absent. Besides, insufficient multi-view consistency leads to ambiguity among shape and appearance, resulting in many floaters during reconstruction. We propose two techniques to initially optimize the 3D Gaussian representation, which take full advantage of structure priors from the limited views and result in a satisfactory outline of the object.

Initialization with Visual Hull. To better leverage object structure information from limited reference images, we utilize the view frustums and object masks to create a visual hull as a geometric scaffold for initializing our 3D Gaussians. Compared with the limited number of SfM points in extremely sparse settings, the visual hull provides more structure priors that help build multiview consistency by excluding unreasonable Gaussian distributions. The cost of the visual hull is just several masks derived from sparse 360° images, which can be easily acquired using current segmentation models such as SAM [Kirillov et al. 2023]. Specifically, points are randomly initialized within the visual hull using rejection sampling: we project uniformly sampled random 3D points onto image planes and retain those within the intersection of all image-space masks. Point colors are averaged from bilinearly interpolated pixel colors across reference image projections. Then we transform these 3D points into 3D Gaussians. For each point, we assign its position as μ and convert its color into sh. The mean distance between adjacent points forms the scale s, while the rotation q is set to a unit quaternion as default. The opacity σ is initialized to a constant value. This initialization strategy relies on the initial masks. Despite potential inaccuracies in these masks or unrepresented concavities by the visual hull, we observed that subsequent optimization processes reliably yield high-quality reconstructions.

Floater Elimination. While the visual hull builds a coarse estimation of the object geometry, it often contains regions that do not belong to the object due to the inadequate coverage of reference images. These regions usually appear to be floaters, damaging the quality of novel view synthesis. These floaters are problematic as the optimization process struggles to adjust them due to insufficient observational data regarding their position and appearance.

To mitigate this issue, we utilize the statistical distribution of distances among the 3D Gaussians to distinguish the primary object and the floaters. This is implemented by the K-Nearest Neighbors (KNN) algorithm, which calculates the average distance to the nearest $\sqrt{P}$ Gaussians for each element in $\mathcal{G}_c$. We then establish a normative range by computing the mean and standard deviation of these distances. Based on statistical analysis, we exclude Gaussians whose mean neighbor distances exceed the adaptive threshold $\tau = \text{mean} + \lambda_e \text{std}$. This thresholding process is repeated periodically throughout optimization, where λ_e is linearly decreased to 0 to refine the scene representation progressively.

Initial Optimization The optimization of $\mathcal{G}_c$ incorporates color, mask, and monocular depth losses. The color loss combines L1 and D-SSIM losses from 3D Gaussian Splatting:

$$\mathcal{L}_1 = \|x - x^{\text{ref}}\|_1, \quad \mathcal{L}_{\text{D-SSIM}} = 1 - \text{SSIM}(x, x^{\text{ref}}), \quad (2)$$

where x is the rendering and x^{ref} is the corresponding reference image. A binary cross entropy (BCE) loss [Jadon 2020] is applied as mask loss:

$$\mathcal{L}_{\text{m}} = -(m^{\text{ref}} \log m + (1 - m^{\text{ref}}) \log(1 - m)), \quad (3)$$

where m denotes the object mask. A shift and scale invariant depth loss is utilized to guide geometry:

$$\mathcal{L}_{\text{d}} = \|D^* - D^*_{\text{pred}}\|_1, \quad (4)$$

where D^* and D^*_{pred} are per-frame rendered depths and monocularly estimated depths [Bhat et al. 2023] respectively. The depth values are computed following a normalization strategy [Ranftl et al. 2020]:

$$D^* = \frac{D - \text{median}(D)}{\frac{1}{M}\sum_{i=1}^M |D - \text{median}(D)|}, \quad (5)$$

where M denotes the number of valid pixels. The overall loss combines these components:

$$\mathcal{L}_{\text{ref}} = (1 - \lambda_{\text{SSIM}})\mathcal{L}_1 + \lambda_{\text{SSIM}}\mathcal{L}_{\text{D-SSIM}} + \lambda_{\text{m}}\mathcal{L}_{\text{m}} + \lambda_{\text{d}}\mathcal{L}_{\text{d}}, \quad (6)$$

where λ_{SSIM}, λ_{m}, and λ_{d} control the magnitude of each term. Thanks to the efficient initialization, our training speed is remarkably fast. It only takes 1 minute to train a coarse Gaussian representation $\mathcal{G}_c$ at a resolution of 779×520.

3.4 Gaussian Repair Model Setup

Combining visual hull initialization and floater elimination significantly enhances 3DGS performance for NVS in sparse 360° contexts. While the fidelity of our reconstruction is generally passable, $\mathcal{G}_c$

[2]Given that the camera intrinsics are known and fixed, we exclude them from the rendering function for simplicity.

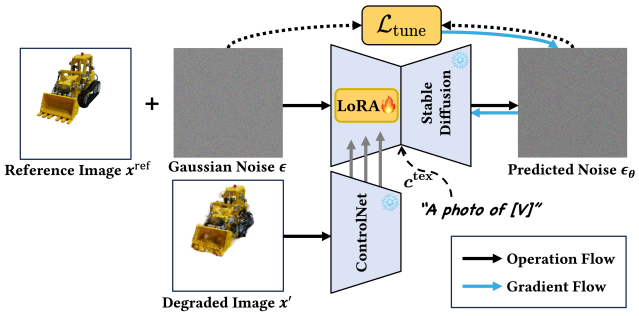

Fig. 3. Illustration of Gaussian repair model setup. First, we add Gaussian noise ϵ to a reference image x^{ref} to form a noisy image. Next, this noisy image along with x^{ref}'s corresponding degraded image x' are passed to a pre-trained fixed ControlNet with learnable LoRA layers to predict a noise distribution ϵ_θ. We use the differences among ϵ and ϵ_θ to fine-tune the parameters in LoRA layers.

still suffers in regions that are poorly observed, regions with occlusion, or even unobserved regions. These challenges loom over the completeness of the reconstruction, like the sword of Damocles.

To mitigate these issues, we introduce a Gaussian repair model $\mathcal{R}$ designed to correct the aberrant distribution of $\mathcal{G}_c$. Our $\mathcal{R}$ takes corrupted rendered images $x'(\mathcal{G}_c, \pi^{\text{nov}})$ as input and outputs photo-realistic and high-fidelity images $\hat{x}$. This image repair capability can be used to refine the 3D Gaussians, leading to learning better structure and appearance details.

Sufficient data pairs are essential for training $\mathcal{R}$ but are rare in existing datasets. To this end, we adopt two main strategies for generating adequate image pairs, *i.e.*, **leave-one-out training** and **adding 3D noises**. For leave-one-out training, we build N subsets from the N input images, each containing $N-1$ reference images and 1 left-out image x^{out}. Then we train N 3DGS models with reference images of these subsets, termed as $\{\mathcal{G}_c^i\}_{i=0}^{N-1}$. After specific iterations, we use the left-out image x^{out} to continue training each Gaussian model $\{\mathcal{G}_c^i\}_{i=0}^{N-1}$ into $\{\hat{\mathcal{G}}_c^i\}_{i=0}^{N-1}$. Throughout this process, the rendered images from the left-out view at different iterations are stored to form the image pairs along with left-out image x^{out} for training the repair model. Note that training these left-out models costs little, with less than N minutes in total. The other strategy is to add 3D noises ϵ_s onto Gaussian attributes. The ϵ_s are derived from the mean μ_Δ and variance σ_Δ of attribute differences between $\{\mathcal{G}_c^i\}_{i=0}^{N-1}$ and $\{\hat{\mathcal{G}}_c^i\}_{i=0}^{N-1}$. This allows us to render more degraded images $x'(\mathcal{G}_c(\epsilon_s), \pi^{\text{ref}})$ at all reference views from the created noisy Gaussians, resulting in extensive image pairs (X', X^{ref}).

We inject LoRA weights and fine-tune a pre-trained ControlNet [Zhang et al. 2023b] using the generated image pairs as our Gaussian repair model. The training procedure is shown in Fig. 3. The loss function, based on Eq. 1, is defined as:

$$\mathcal{L}_{\text{tune}} = \mathbb{E}_{x^{\text{ref}},t,\epsilon,x'}\left[\|(\epsilon_\theta(x_t^{\text{ref}}, t, x', c^{\text{tex}}) - \epsilon)\|_2^2\right], \tag{7}$$

where c^{tex} denotes an object-specific language prompt, defined as "a photo of [V]," as per Dreambooth [Ruiz et al. 2023]. Specifically, we inject LoRA layers into the text encoder, image condition branch, and U-Net for fine-tuning. Please refer to the Appendix for details.

3.5 Gaussian Repair with Distance-Aware Sampling

After training $\mathcal{R}$, we distill its target object priors into $\mathcal{G}_c$ to refine its rendering quality. The object information near the reference views is abundant. This observation motivates designing distance as a criterion in identifying views that need rectification, leading to distance-aware sampling.

Specifically, we establish an elliptical path aligned with the training views and focus on a central point. Arcs near Π^{ref}, where we assume $\mathcal{G}_c$ renders high-quality images, form the reference path. The other arcs, yielding renderings, need to be rectified and define the repair path, as depicted in Fig. 4. In each iteration, novel viewpoints, $\pi_j \in \Pi^{\text{nov}}$, are randomly sampled among the repair path. For each π_j, we render the corresponding image $x_j(\mathcal{G}_c, \pi_j)$, encode it to be $\mathcal{E}(x_j)$ by the latent diffusion encoder $\mathcal{E}$ and pass $\mathcal{E}(x_j)$ to the image conditioning branch of $\mathcal{R}$. Simultaneously, a cloned $\mathcal{E}(x_j)$ is disturbed into a noisy latent z_t:

$$z_t = \sqrt{\bar{\alpha}_t}\mathcal{E}(x_j) + \sqrt{1-\bar{\alpha}_t}\epsilon, \text{ where } \epsilon \sim \mathcal{N}(0, I), t \in [0, T], \tag{8}$$

which is similar to SDEdit [Meng et al. 2022]. We then generate a sample $\hat{x}_j$ from $\mathcal{R}$ by running DDIM sampling [Song et al. 2021] over $k = \lfloor 50 \cdot \frac{t}{T} \rfloor$ steps and forwarding the diffusion decoder $\mathcal{D}$:

$$\hat{x}_j = \mathcal{D}(\text{DDIM}(z_t, \mathcal{E}(x_j))), \tag{9}$$

where $\mathcal{E}$ and $\mathcal{D}$ are from the VAE model used by the diffusion model. The distances from π_j to Π^{ref} is used to weight the reliability of $\hat{x}_j$, guiding the optimization with a loss function:

$$\mathcal{L}_{\text{rep}} = \mathbb{E}_{\pi_j,t}\left[w(t)\lambda(\pi_j)\left(\|x_j - \hat{x}_j\|_1 + \|x_j - \hat{x}_j\|_2 + L_p(x_j, \hat{x}_j)\right)\right],$$
$$\text{where } \lambda(\pi_j) = \frac{2 \cdot \min_{i=1}^{N}(\|\pi_j - \pi_i\|_2)}{d_{\max}}. \tag{10}$$

Here, L_p denotes the perceptual similarity metric LPIPS [Zhang et al. 2018], $w(t)$ is a noise-level modulated weighting function from DreamFusion [Poole et al. 2023], $\lambda(\pi_j)$ denotes a distance-based weighting function, and $d_{\max}$ is the maximal distance among neighboring reference viewpoints. To ensure coherence between 3D Gaussians and reference images, we continue training $\mathcal{G}_c$ with $\mathcal{L}_{\text{ref}}$ during the whole Gaussian repair procedure.

3.6 COLMAP-Free GaussianObject (CF-GaussianObject)

Current SOTA sparse view reconstruction methods rely on precise camera parameters, including intrinsics and poses, obtained through an SfM pipeline with dense input, limiting their usability in daily applications. This process can be cumbersome and unreliable in sparse-view scenarios where matched features are insufficient for accurate reconstruction.

To overcome this limitation, we introduce an advanced sparse matching model, DUSt3R [Wang et al. 2024a], into GaussianObject to enable COLMAP-free sparse 360° reconstruction. Given reference input images X^{ref}, DUSt3R is formulated as:

$$\mathcal{P}, \hat{\Pi}^{\text{ref}}, \hat{K}^{\text{ref}} = \text{DUSt3R}(X^{\text{ref}}), \tag{11}$$

where $\mathcal{P}$ is an estimated coarse point cloud of the scene, and $\hat{\Pi}^{\text{ref}}$, $\hat{K}^{\text{ref}}$ are the predicted camera poses and intrinsics of X^{ref}, respectively. For CF-GaussianObject, we modify the intrinsic recovery module within DUSt3R, allowing $x_i \in X^{\text{ref}}$ to share the same intrinsic $\hat{K}$. This adaption enables the retrieval of $\mathcal{P}$, $\hat{\Pi}^{\text{ref}}$, and $\hat{K}$. Besides,