Efficient Panoramic 3D Gaussian Splatting with Gradient Optimization and Scene Regularization

Ruopeng Zhao Yafeng Zhao Zhijie Song Gang Shi*

Xinjiang University

zhaoruopeng@stu.xju.edu.cn, shigang@xju.edu.cn

Abstract

Panoramic cameras capture a full 360° field of view and are increasingly used in immersive 3D applications. Although 3D Gaussian Splatting (3DGS) has demonstrated remarkable success in real-time rendering, its formulation is primarily suited to perspective projection. This paper presents a high-fidelity real-time rendering framework specifically designed for panoramic imagery. To address gradient instability caused by spherical distortion and reconstruction degradation due to Gaussian degeneracy, we analyze gradient propagation under panoramic projection and propose a consistency-driven optimization strategy. A cross-boundary Gaussian continuity mechanism mitigates discontinuities at the longitude seam, enhancing stability and geometric accuracy. In addition, a geometry-aware isotropy regularization combined with a quantile-guided scale ratio constraint effectively suppresses extreme ellipsoidal degeneracy. A learnable compression module further reduces model size while preserving visual quality. Experimental results on public datasets demonstrate that the proposed method achieves superior or comparable performance to state-of-the-art approaches in both accuracy and efficiency.

***Keywords:* Panoramic Scene Reconstruction; 3D Gaussian Splatting; Novel View Synthesis**

1. Introduction

3D scene reconstruction is a fundamental research area in computer vision, with extensive applications in fields such as autonomous driving and robotic perception. Panoramic (360°) images offer a distinct advantage for large-scale environmental modeling and immersive navigation tasks by capturing a comprehensive field of view in a single shot. However, the spherical projection inherent to them introduces severe distortion and accumulated reprojection errors in high-latitude regions, which leads to unstable gradient propagation and ultimately compromises the optimization process and final reconstruction quality. In the context of deep learning-based methods, Neural Radiance Fields (NeRF) [17] achieve high-quality novel view synthesis through implicit coordinate-based networks. To adapt this paradigm to panoramic scenes, the research community has proposed several variants [8, 12, 4, 6], which optimize the NeRF framework from different perspectives to address various characteristics of panoramic imagery. Although these methods demonstrate outstanding visual quality, their reliance on extensive ray sampling for volume rendering results in high computational costs for both training and inference, limiting their real-time applicability.

To overcome this efficiency bottleneck, 3D Gaussian Splatting (3DGS) [10] has emerged. This method utilizes explicit 3D Gaussian primitives as the scene representation and achieves real-time rendering through differentiable rasterization. In standard perspective camera scenarios, 3DGS not only significantly reduces training time but also delivers rendering quality comparable to, or even surpassing, NeRF. However, when applied directly to panoramic images, its core assumption of perspective projection becomes invalid, leading to a dramatic degradation in quality. To bridge this gap, recent studies [15, 13] have focused on adapting 3DGS for the panoramic domain. While these methods strike a good balance between speed and quality, they still suffer from limitations such as a reliance on local affine approximations, susceptibility to geometric artifacts, and significant high-latitude errors, making the accurate recovery of scene details challenging.

While existing methods have made notable progress in panoramic scene reconstruction, several critical limitations persist. The projection distortion in high-latitude regions causes unstable gradient propagation, which complicates the optimization process. Concurrently, scenes often feature a large number of redundant or geometrically degenerated Gaussian primitives, leading to unnecessary storage and rendering overhead. To address these issues, this paper proposes an efficient and robust method for panoramic 3D re-

*Corresponding author

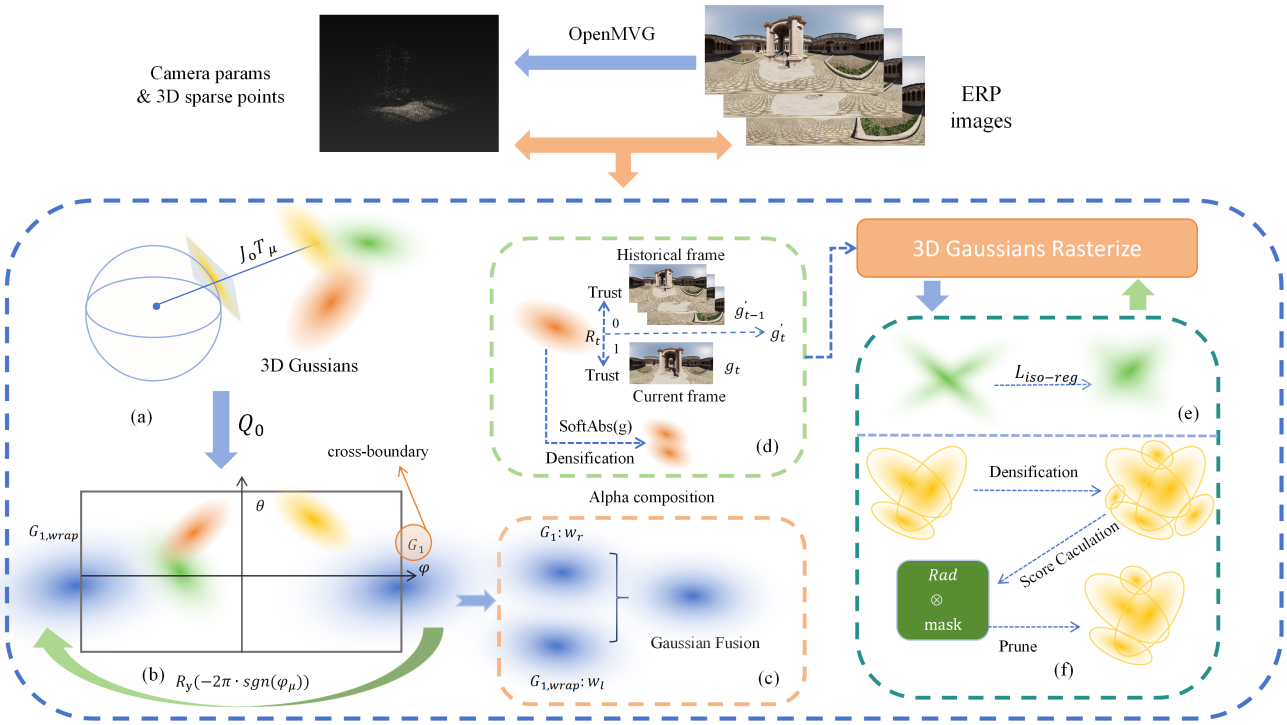

Figure 1. Overall schematic diagram of our method. (a) ODGS tangent plane projection. (b) and (c) Cross-boundary Gaussian Fusion. (d) Gradient Optimization for Continuous Panoramic Projection. (e) Isotropic Regularization. (f) LP pruning strategy based on Radsplat.

construction. We first tackle the core challenge of gradient propagation by designing a gradient optimization method under continuous panoramic projection. By stabilizing and correcting gradients, we ensure a reliable optimization process, thereby significantly enhancing geometric accuracy. Building on this, we introduce a geometry-aware isotropic regularization method that dynamically constrains the morphology of Gaussian primitives, effectively suppressing the extreme ellipsoidal degeneracy caused by projection distortion and thus recovering more scene details. Finally, to enhance efficiency while preserving fidelity, we integrate a learnable compression strategy that achieves a better trade-off between reconstruction quality and model overhead.

2. Related Work

2.1. Panoramic Image Reconstruction

In contrast to conventional perspective images, 360° panoramic images can capture the complete surrounding environment in a single shot, greatly reducing the burden of multi-view data acquisition. However, this advantage is accompanied by severe geometric distortion introduced by the Equirectangular Projection (ERP). Unlike the local linear projection of perspective cameras, ERP causes non-uniform stretching in the polar regions, which hinders the maintenance of geometric consistency across different views. Furthermore, occluders such as the camera rig, support structures, or the capturing platform often appear in panoramic images. These elements, when viewed from different positions, can be misinterpreted as consistent structures during reconstruction, leading to artifacts or "floaters". These challenges make it difficult to achieve satisfactory results by directly applying conventional multi-view reconstruction or stereo matching methods.

Early research often relied on structured light or Structure-from-Motion (SfM) techniques to infer camera poses and scene geometry through feature point matching. However, as scene scale increases, such methods become computationally intensive and highly dependent on the stability of feature extraction and matching. To enhance accuracy and efficiency, researchers have explored combining panoramic images with deep learning, improving dense reconstruction through Convolutional Neural Networks[11] (CNNs) or multi-plane image representations. Sun et al.[20] proposed a divide-and-conquer planar reconstruction method for panoramic images, which utilizes a CNN to predict pixel-wise plane attributes for horizontal and vertical plane clustering. They also designed a yaw-invariant reparameterization mechanism to improve robustness to varying camera orientations. Validated on the PanoH&V dataset, this method more accurately extracts the primary geometric structures of indoor scenes compared to tradi-

tional SfM and plane fitting methods. Hyeonjoong Jang et al. [9] proposed an efficient method for spherical stereo matching by adapting the RAFT optical flow network to spherical disparity estimation, and fused the results using a spherical binoctree data structure to obtain per-frame depth. Beyond specific panoramic methods, the broader field of 3D scene representation has seen significant progress in enhancing geometric fidelity. Recent approaches have leveraged diffusion models to achieve high-quality point cloud upsampling [5], and employed adversarial learning for unsupervised 3D shape completion from multi-view [22] or single-view [23] inputs. While these methods effectively address data sparsity and incompleteness in object-level or perspective scenarios, directly adapting high-fidelity reconstruction to the panoramic domain remains challenging due to the severe non-linear distortions inherent in Equirectangular Projection (ERP).

This approach achieves high-precision scene geometry even from short, handheld camera trajectories. Despite their potential in indoor scenes and regular geometric environments, these studies still face challenges in complex real-world scenarios, including texture blurring, boundary distortion, and high computational overhead. Consequently, the development in panoramic image reconstruction has gradually shifted towards integrating novel rendering and representation techniques to overcome structural inconsistencies caused by distortion while ensuring global coverage.

2.2. NeRF-based Panoramic Reconstruction

Neural Radiance Fields (NeRF)[17] have established a new paradigm for 3D reconstruction, implicitly modeling a scene's color and density with a multi-layer perceptron (MLP) to synthesize high-fidelity novel views. To adapt this framework for panoramic imagery, various NeRF variants have been proposed to address the non-linear distortions introduced by Equirectangular Projection (ERP).

For representing unbounded panoramic scenes, MipNeRF360[2] introduced anti-aliasing mip-mapping and a specialized spatial transformation to mitigate resolution inconsistencies and boundary detail loss. Similarly, EgoNeRF[6] employed spherical feature grids and a hierarchical density mechanism, enabling efficient representation of large-scale egocentric scenes. For large-scale navigation, 360Roam[8] proposed a complete system that first trains a global model and then builds a probabilistic occupancy map to accelerate rendering, achieving real-time panoramic roaming.

Despite achieving excellent rendering quality, these methods are fundamentally limited by the computational expense of volume rendering. The requisite extensive ray sampling for training results in prolonged optimization times, often spanning hours to days, which severely restricts their use in real-time applications. This inherent inefficiency has motivated a shift in the research community towards more efficient explicit representations, most notably 3D Gaussian Splatting (3DGS).

2.3. 3DGS-based Panoramic Reconstruction

3D Gaussian Splatting (3DGS) provides an efficient explicit alternative to neural representations, using a set of Gaussian primitives to achieve real-time, high-fidelity rendering through a differentiable rasterization pipeline. However, the original 3DGS is designed exclusively for perspective projection and suffers severe quality degradation when applied to panoramic Equirectangular Projection (ERP) images, necessitating specialized adaptations.

Initial research addressed general non-perspective camera models. Fisheye-GS[16]analytically maps 3D Gaussians to the image plane using the fisheye projection equation and its Jacobian, preserving kernel shape consistency without pre-undistortion. Extending this concept, 3DGUT[24] employs the Unscented Transform (UT) to statistically propagate Gaussian parameters through arbitrary non-linear camera models, improving shape recovery under strong distortion.

For the complexities of panoramic cameras, methods are broadly categorized into two approaches. The first involves direct projection into the equirectangular space. A prominent example is OmniGS[15], which introduced a GPU-accelerated rasterizer that directly maps 3D Gaussians into the ERP space. While versatile and fast, this approach can produce significant errors in the highly distorted polar regions. Building on this, SPaGS[14] introduced a bounding box constraint mechanism to confine Gaussians to target regions, preventing the unbounded diffusion caused by projection distortion. The second category relies on local tangent plane approximations. An early extension, 360-GS[1], utilized a two-step projection (tangent plane to sphere) and incorporated structural priors from panoramic depth to suppress artifacts. In comparison, ODGS[13] presents a more rigorous and unified formulation. For each Gaussian, it explicitly constructs a local perspective imaging plane tangent to the unit sphere and then analytically maps the 2D projection to the ERP domain via a final Jacobian matrix. While its dynamic densification strategy alleviates over-splitting in polar regions, its gradient mechanism remains relatively coarse. It fails to fully account for the deformation and non-uniformity of the projection, which can lead to under- or over-densification in complex regions and motivates our work on a more refined optimization strategy.

3. Method

3.1. Preliminaries

3DGS represents scenes using anisotropic 3D Gaussians as primitives. Each Gaussian is defined by a mean

$\boldsymbol{\mu}$ representing its 3D position and a covariance matrix $\boldsymbol{\Sigma} = \mathbf{RSS}^\top\mathbf{R}^\top$. The parameters for position, scale, rotation, color, and opacity are jointly optimized through a differentiable rasterizer and a photometric loss. Under a perspective camera model, the projection function $\pi(\boldsymbol{\mu}) = \mathbf{K}1 : 2[\mu_x/\mu_z, \mu_y/\mu_z, 1]^\top$ for a point $\boldsymbol{\mu}$ onto the pixel plane and its first-order Jacobian $\mathbf{J}$ are given by:

$$\mathbf{J} = \frac{\partial \pi(\mu)}{\partial \mu} = \begin{bmatrix} \frac{f_x}{\mu_z} & 0 & -\frac{f_x \mu_x}{\mu_z^2} \\ 0 & \frac{f_y}{\mu_z} & -\frac{f_y \mu_y}{\mu_z^2} \end{bmatrix}, \tag{1}$$

This yields the 2D covariance matrix in the image plane:

$$\boldsymbol{\Sigma}_{2D} = \mathbf{JW\Sigma W}^\top\mathbf{J}^\top, \tag{2}$$

where $\mathbf{W}$ is the world-to-camera transformation matrix. The color of each pixel is then computed via α-blending in depth order:

$$C = \sum_j c_j \alpha_j T_j, \tag{3}$$

where T_j is the transmittance and α_j is the effective opacity:

$$T_j = \prod_{k<j}(1-\alpha_k), \quad \alpha_j = o_j \exp(-\frac{1}{2}\Delta \mathrm{x}^{\mathrm{T}}\Sigma_{\mathrm{2D}}^{-1}\Delta \mathrm{x}), \tag{4}$$

For a non-perspective spherical camera model, they adopt the tangent plane projection approximation from ODGS, as shown in Figure 1(a). The camera coordinate system is rotated to a local tangent plane whose normal aligns with the current Gaussian's center direction $\hat{\boldsymbol{\mu}} = \boldsymbol{\mu}/|\boldsymbol{\mu}|$, with the rotation matrix denoted as $\mathbf{T}_\mu$. The final panoramic projection Jacobian, $\mathbf{J}_{\mathbf{omni}}$, is composed as:

$$\mathbf{J}_{\mathbf{omni}} = \mathbf{S}_o\mathbf{Q}_o\mathbf{J}_o\mathbf{T}_\mu, \tag{5}$$

where $\mathbf{J}_o = \mathrm{diag}(1/|\boldsymbol{\mu}|, 1/|\boldsymbol{\mu}|)$ is the local perspective Jacobian, $\mathbf{Q}_o = \mathrm{diag}(\sec\theta_\mu, 1)$ is a horizontal stretching matrix, and $\mathbf{S}_o = \mathrm{diag}(W/(2\pi), H/\pi)$ maps angular coordinates to pixel coordinates. The corresponding 2D covariance is:

$$\boldsymbol{\Sigma}_{\text{2D,omni}} = \mathbf{J}_{\text{omni}}\mathbf{W\Sigma W}^\top\mathbf{J}_{\text{omni}}^\top, \tag{6}$$

3.2. Gradient Optimization for Continuous Panoramic Projection

Consistency-Driven Gradient Stabilization. In the standard 3D Gaussian Splatting (3DGS) framework, both the adaptive densification and parameter optimization processes rely on positional gradients in view space. While this design is effective for handling perspective projection images, it has inherent limitations. Specifically, the screen space projections of a single 3D Gaussian primitive from different camera views can produce 2D positional gradients with opposing directions. Under the conventional gradient accumulation and update mechanism, these opposing gradients can cancel each other out. This phenomenon, known as gradient collision, hinders the optimization process. To address this issue, AbsGS [25] and the concurrent work, GOF [26], propose accumulating the absolute values of the gradients rather than the gradient vectors themselves.

They decompose the native gradient ($g_{i,x}$) of the pixel-wise loss with respect to the mean, as formulated in Eq. (7):

$$g_{i,x} = \frac{\partial L_j}{\partial \mu_{i,x}} = \sum_{l=1}^{3} \frac{\partial L_j}{\partial c_l^j} \times \frac{\partial c_l^j}{\partial \alpha_i} \times \frac{\partial \alpha_i}{\partial \mu_{i,x}}, \tag{7}$$

where the gradient can be viewed as the combined effect of the pixel-wise loss, the rendered color, opacity, and the mean. Therefore, the gradient collision, which occurs during accumulation when sub-gradients have opposite signs, can be avoided by accumulating the absolute value of each component. This modified gradient, denoted as $\widehat{g}_{i,x}$, is defined in Eq. (8):

$$\widehat{g}_{i,x} = \sum_{l=1}^{3} \left| \frac{\partial L_j}{\partial c_l^j} \times \frac{\partial c_l^j}{\partial \alpha_i} \times \frac{\partial \alpha_i}{\partial \mu_{i,x}} \right|, \tag{8}$$

The motivation for this approach is that the mere existence of a gradient, regardless of its direction, is a signal of under-convergence and should be treated as a valid driver for densification. Experiments have also demonstrated that their method achieves a slight improvement in the quality of scenes rendered under perspective projection compared to the original 3DGS.

For panoramic images employing Equirectangular Projection (ERP), the problem of gradient collision is far more severe. Unlike the linear transformation of perspective projection, ERP is a non-linear mapping that induces significant geometric distortion in high-latitude regions. Consequently, the shape and size of the 2D projection of the same 3D Gaussian primitive can vary dramatically depending on the viewing angle and latitude. This instability in the projected shape leads to inconsistent screen-space gradients, as shown in Figure 2. As a result, gradients from different views are more likely to point in conflicting directions, triggering more frequent collisions. Concurrently, the projection distortion itself can generate misleading gradients that are unrelated to the true scene geometry.

To address this issue, we first employ a smooth and differentiable SoftAbs function to replace the standard absolute value function. It is defined as:

$$\mathrm{softAbs}(t) = \sqrt{t^2 + \beta^2} - \beta, \tag{9}$$

where β is a small positive number that ensures the function is smoothly differentiable across its entire domain, providing a smooth transition around the origin to maintain the stability of the gradient descent process.

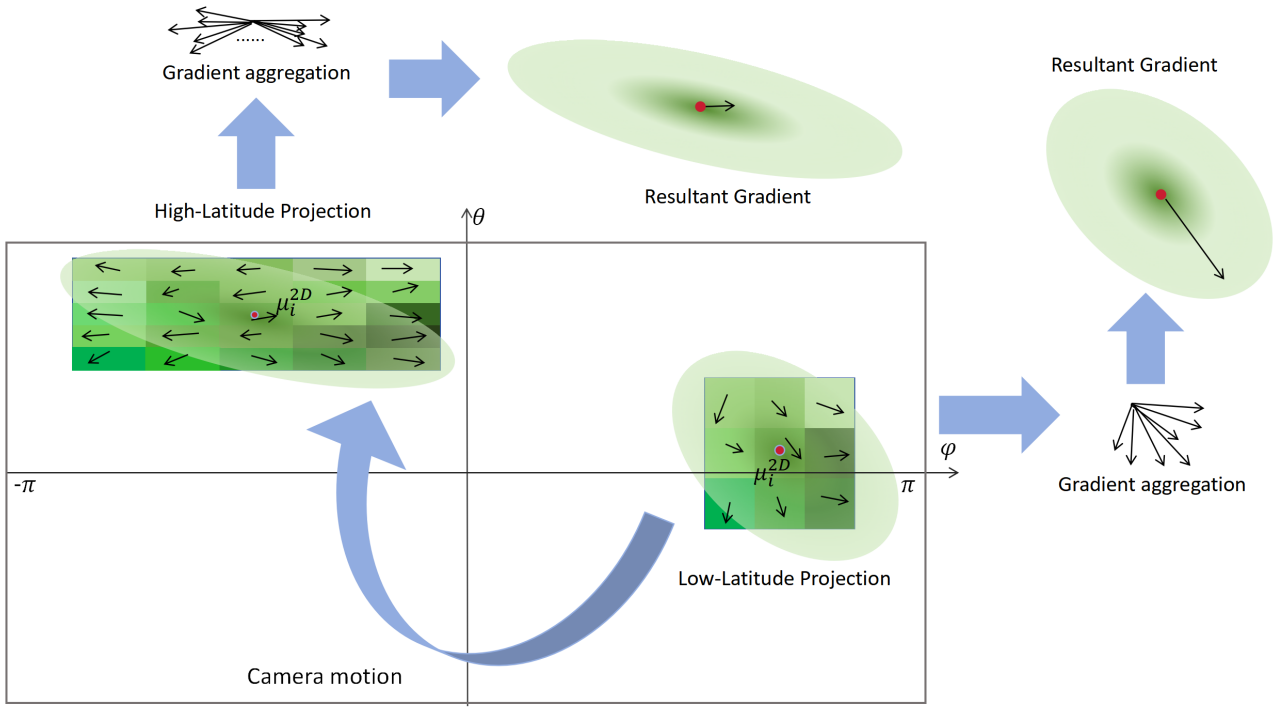

Figure 2. At high latitudes, Gaussian primitives are subject to projection distortion and greater pixel density, which causes their gradients to skew towards a horizontal orientation. Consequently,this leads to severe gradient collision when gradients from multiple views are aggregated.

For the splitting strategy of 3D Gaussian primitives, ODGS proposed a dynamic densification approach that replaces a fixed threshold with a latitude-dependent one to trigger splitting. This adaptive mechanism helps suppress the splitting of high-latitude Gaussians. Nevertheless, the ODGS strategy still fundamentally relies on a single, potentially noisy gradient magnitude. It mitigates the symptoms caused by distortion but fails to address the core issue of the gradient itself: its cross-view consistency. To this end, we propose a more profound, decoupled optimization paradigm that separates the gradient used for triggering geometric splitting from the backpropagation gradient used for updating Gaussian parameters. Specifically, while we retain the dynamic densification strategy of ODGS, we only feed it the SoftAbs-processed gradients, which are used exclusively to determine whether to perform a split. Concurrently, we design a more sophisticated mechanism, the Gradient Consistency Ratio (R), is defined in Eq. (10) and Eq. (11), to dynamically evaluate the stability and consistency of gradient directions. This ratio quantifies the degree of consistency by comparing the magnitude of the conventional gradient sum with the sum of its directional components.

$$R_x = \frac{|\sum_{i=1}^{N} g_{i,x}|}{\sum_{i=1}^{N} \text{softAbs}(g_{i,x}) + \epsilon}, \tag{10}$$

$$R_y = \frac{|\sum_{i=1}^{N} g_{i,y}|}{\sum_{i=1}^{N} \text{softAbs}(g_{i,y}) + \epsilon}, \tag{11}$$

The ratio R precisely quantifies the directional consistency of the historically accumulated gradients. A value approaching 0 indicates severe gradient collision, while a value approaching 1 suggests strong directional agreement. Based on this metric, we construct an R-weighted exponentially-mixed gradient to completely replace the original backpropagated gradient of the Gaussian primitive:

$$g'_t = R_t \cdot g_t + (1 - R_t) \cdot g'_{t-1}, \tag{12}$$

When the gradient direction is highly consistent ($R \to 1$), the parameter update is predominantly driven by the information-rich gradient from the current signal g_t, enabling rapid convergence. Conversely, when the gradient direction exhibits significant conflict ($R \to 0$), the update tends to rely on the historically accumulated and more directionally stable componentg'_{t-1}, as shown in Figure 1(d),effectively filtering out the destructive noise introduced by view inconsistencies or projection distortion.

Continuous Cross-Boundary Gaussian Fusion. ODGS employs tangent planes to project 3D Gaussians from the spherical angular domain onto a rectangular plane in the

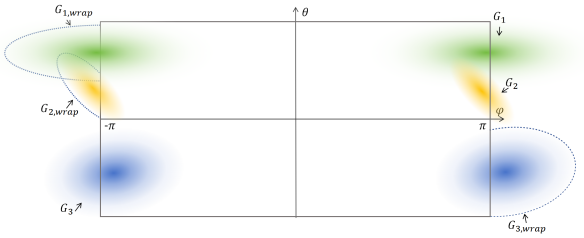

Figure 3. For the identified cross-boundary Gaussian primitives G_1, G_2, G_3, corresponding "wrapped" instances $(G_{1,\text{wrap}}, G_{2,\text{wrap}}, G_{3,\text{wrap}})$ are generated on the opposite side of the boundary. These wrapped instances share all parameters with their originals except for their 3D center positions. The dashed lines indicate the non-cross-boundary regions of the projections.

pixel domain. However, a discontinuity arises when a Gaussian's azimuth is located near the longitude boundary ($\pm\pi$). If the Gaussian primitive is large, its 2D projection is forcibly split across the left and right edges of the image. This boundary discontinuity introduces optimization ambiguity; as the azimuth of a Gaussian's center, ϕ_μ, minutely crosses the $\pm\pi$ boundary, the set of pixels covered by its projection changes drastically and discontinuously, which hinders the stable convergence of the model. To ensure that the optimizer can find the correct shortest angular distance during optimization, we have designed a complete processing pipeline that encompasses both forward rendering and backpropagation.

During the forward pass, we perform boundary detection on each Gaussian primitive. We determine its standard deviation along the azimuth axis, σ_ϕ, by analyzing the 2D covariance matrix of its projection, and define its effective angular radius as three times this standard deviation, $3\sigma_\phi$. A Gaussian is classified as "cross-boundary" if the absolute value of its central azimuth plus its effective angular radius exceeds the π boundary:

$$|\phi_\mu| + 3\sigma_\phi > \pi, \tag{13}$$

Once a cross-boundary Gaussian G_i is identified, we create a virtual "wrapped" instance G'_i in 3D space, as shown in Figure 3 or Figure 1(b), which shares all parameters with the original except for its 3D mean. We compute the new 3D mean vector $\boldsymbol{\mu}'_{3\text{D},i}$ for the wrapped instance by applying a full periodic rotation to the original 3D mean vector $\boldsymbol{\mu}_{3\text{D},i}$:

$$\boldsymbol{\mu}'_{3\text{D},i} = \mathbf{R}_y(-2\pi \cdot \text{sgn}(\phi_\mu)) \cdot \boldsymbol{\mu}_{3\text{D},i}, \tag{14}$$

where ϕ_μ is the azimuth determined by the original 3D mean vector and $\mathbf{R}_y(\theta)$ is the standard 3D rotation matrix around the y-axis. Both G_i and G'_i are then sent to the rasterizer for rendering.

To prevent rendering artifacts from the direct superposition of the two instances, we designed an edge-aware Alpha fusion strategy. The core is to define a smooth transition zone of width ϵ_w near the image boundaries and to weight the contributions. We use the cubic function $s(t) = 3t^2 - 2t^3$ as the base weighting function for its smooth gradients. We first define two general and position-bound weight functions, which are $w_r(u)$ for the right boundary transition and $w_l(u)$ for the left boundary transition. They are obtained based on the horizontal coordinate u_x of the pixel and $s(t)$. For the Gaussian instance responsible for rendering the right part of the image, its weight function $w_r(u)$ is defined as:

$$w_r(u) = \begin{cases} s((W - u_x)/\epsilon_w), & \text{if } W - \epsilon_w < u_x \le W \\ 1, & \text{otherwise} \end{cases}, \tag{15}$$

Correspondingly, for the instance responsible for rendering the left part of the image, its weight function $w_l(u)$ is defined as:

$$w_l(u) = \begin{cases} s(u_x/\epsilon_w), & \text{if } 0 \le u_x < \epsilon_w \\ 1, & \text{otherwise} \end{cases}, \tag{16}$$

The base opacities of the original Gaussian G_i and the wrapped instance G'_i are determined based on the Mahalanobis distance, which measures the deviation of a pixel position u_x from the projected Gaussian center while considering the covariance structure of the distribution. For each Gaussian, the squared Mahalanobis distance is defined as:

$$d_i^2(u) = (u - \boldsymbol{\mu}_{2\text{D},i})^\top \boldsymbol{\Sigma}_{2\text{D},i}^{-1} (u - \boldsymbol{\mu}_{2\text{D},i}), \tag{17}$$

and for the wrapped Gaussian instance:

$$d_{i,\text{wrap}}^2(u) = (u - \boldsymbol{\mu}'_{2\text{D},i})^\top \boldsymbol{\Sigma}'^{-1}_{2\text{D},i} (u - \boldsymbol{\mu}'_{2\text{D},i}), \tag{18}$$

Using this distance measure, the base opacities of the two Gaussians are then computed as:

$$\alpha_i(u) = o_i \cdot \exp\left(-\tfrac{1}{2} d_i^2(u)\right), \tag{19}$$

$$\alpha_{i,\text{wrap}}(u) = o_i \cdot \exp\left(-\tfrac{1}{2} d_{i,\text{wrap}}^2(u)\right), \tag{20}$$

where o_i is the shared opacity parameter. Assuming the original Gaussian's center is on the right half ($\phi_\mu > 0$), the opacities are weighted to ensure a smooth transition:

$$\alpha'_i(u) = \alpha_i(u) \cdot w_r(u), \tag{21}$$

$$\alpha'_{i,\text{wrap}}(u) = \alpha_{i,\text{wrap}}(u) \cdot w_l(u), \tag{22}$$

The final effective opacity $a_{\text{effective},i}(u)$ for Alpha composition is obtained by combining(Figure 1(c)) the contributions:

$$a_{\text{effective},i}(u) = 1 - (1 - \alpha'_i(u)) \cdot (1 - \alpha'_{i,\text{wrap}}(u)), \tag{23}$$

Figure 4. An abundance of highly elongated, needle-like Gaussian primitives produced by ODGS within the Bricks scene from the Ricoh360 dataset.

This Alpha fusion strategy ensures the total gradient signal remains smooth and continuous across the $\pm\pi$ boundary. As it is only activated for a small number of cross-boundary Gaussians, the computational overhead is minimal. This approach fundamentally resolves the boundary discontinuity problem in panoramic reconstruction at a minimal cost.

3.3. Geometry-Aware Isotropic Regularization

In 3DGS, the projection is based on a local affine approximation. As the field of view increases or the focal length decreases, the error of this approximation grows, which in turn induces larger and more elongated ellipsoids to compensate for projection errors. The Equirectangular Projection (ERP) features an extremely wide field of view, and its panoramic projection Jacobian, $\mathbf{J}_{\text{omni}}(\phi, \theta)$, explicitly includes a latitude-dependent scaling term in $\mathbf{Q}_o = \text{diag}(\sec\theta_\mu, 1)$. The horizontal scale and orientation of $\mathbf{J}_{\text{omni}}$ change rapidly with position, exhibiting strong anisotropic distortion. This causes the same 3D Gaussian primitive to be projected into 2D ellipses of varying shapes and orientations across different frames. This issue is particularly pronounced in roaming scenarios. For a fixed point in the scene, as the camera translates, its line of sight traces a long arc on the unit sphere. ERP approximates this long arc as a nearly horizontal pixel trajectory. Since a single 2D ellipse cannot consistently fit this trajectory across frames due to differing projection directions and scales, the optimizer adopts the most direct compensation strategy: it enlarges the major axis of the covariance and elongates it along the trajectory, like Figure4. This results in the formation of numerous thin, elongated, and disparately rotated Gaussians that erase high-frequency texture details.

To suppress this geometric degeneracy, we introduce a Geometrically-Aware Isotropic Regularization loss term. The core objective of this method is not to simply enforce isotropy for all Gaussian primitives, but rather to constrain their morphology within a dynamic and healthy range, thereby simultaneously suppressing both extremely anisotropic shapes (overly elongated) and extremely isotropic shapes (nearly spherical) that are inefficient for representation. First, we define a metric for Gaussian morphology—the anisotropy ratio, denoted as r. For the i-th Gaussian, its scales along the three principal axes are represented by $\mathbf{s}_i = (s_{i,1}, s_{i,2}, s_{i,3})$. Its anisotropy ratio r_i is then calculated as follows:

$$r_i = \frac{\max(\mathbf{s}_i)}{\min(\mathbf{s}_i) + \epsilon}, \tag{24}$$

where ϵ is a small constant to ensure numerical stability. Based on this metric, our designed regularization loss, $\mathcal{L}_{\text{iso-reg}}$, aims to constrain the r_i value of every Gaussian within a dynamically determined target interval $[r_{\min}, r_{\max}]$. Its mathematical form is defined as:

$$\mathcal{L}_{\text{iso-reg}} = \lambda \sum_{i=1}^{N} w_i \left(\text{ReLU}(r_{\min} - r_i) + \text{ReLU}(r_i - r_{\max})\right), \tag{25}$$

The design of this loss function incorporates several key adaptive mechanisms. At its core is a dual-sided interval penalty, implemented via two ReLU functions: the term $\text{ReLU}(r_{\min} - r_i)$ penalizes inefficient, nearly spherical representations, while the term $\text{ReLU}(r_i - r_{\max})$ suppresses the formation of extremely elongated shapes. Crucially, the target interval $[r_{\min}, r_{\max}]$ is not a fixed hyperparameter but is dynamically computed in each training iteration based on the overall distribution of the current anisotropy ratios r_i. In our method, we use the 10th and 90th quantiles of this distribution as the bounds, ensuring a minimum gap between them. Furthermore, considering that ERP distortion is most severe in the polar regions, we introduce a spatial weighting scheme w_i based on the latitude θ_i of the Gaussian's center to impose a stronger geometric constraint in high-latitude areas. This weight w_i is defined as:

$$w_i = \begin{cases} w_{\text{high}}, & \text{if } |\theta_i| > \theta_{\text{threshold}} \\ w_{\text{base}}, & \text{otherwise} \end{cases}, \tag{26}$$

where the weight w_{high} is substantially increased for Gaussians at high latitudes. In our experiments, we set $\theta_{\text{threshold}}$ to $75°$, w_{high} to 1, w_{base} to 0.5, and the regularization loss weight λ to 0.005.

Adding this regularization loss term to the total training objective, as shown in Figure 5, significantly suppresses the

Figure 5. The addition of a geometry-aware isotropic regularization loss to our baseline method effectively suppresses needle-like Gaussian artifacts in the Roming scene. The images demonstrate the superiority of our method (bottom) over ODGS (top) in clearly reconstructing object contours and fine textural details.

prevalent needle-like Gaussian artifacts in the scene. It effectively guides the optimization process to maintain the geometric reasonableness of Gaussian primitives while fitting the image, resulting in a more accurate geometric representation.

3.4. Panoramic Scene Compression Strategy

Panoramic image reconstruction requires a dense Gaussian primitive representation due to its 360° spherical field of view, particularly to preserve details in distortion-prone high-latitude regions. Existing pruning strategies like Mini-Splatting[7] score Gaussians by cumulative weight, which, for panoramic images, favors large low-frequency components (e.g., unbounded scenes,High-latitude large Gaussian) over high-frequency texture details, thereby degrading rendering quality. In contrast, RadSplat's[19] importance score is based on a Gaussian's maximum visual contribution, written as Eq. (27), where $\alpha_i^r \cdot T_i^r$ is the contribution of Gaussian G_i along ray r, it can effectively preserving sparse yet structurally key primitives.

$$\mathbf{S}_i = \max_{\mathbf{I_f} \in \mathcal{I}_\{, r \in \mathbf{I_f}} \alpha_i^r \cdot T_i^r, \tag{27}$$

Building on this, we adapt the LP-3DGS[28] framework, which is based on RadSplat importance scores, to the spherical camera model. LP-3DGS assigns each Gaussian a learnable pruning gate m_i controlled by the Gumbel-Sigmoid function, enabling a continuous and differentiable approximation of the discrete pruning decision for seamless end-to-end training. During training, we use $\mathbf{S}_i$ to guide the gate: when $\mathbf{S}_i$ is high, m_i tends to approach 1 (keep); when $\mathbf{S}_i$ is low, under the mask regularization, m_i tends to approach 0 (prune), as shown in Figure 1(f). To encourage sparsity, LP-3DGS introduces a mask loss R_{mask} to regularize all gates:

$$R_{mask} = \frac{1}{N} \sum_{i=1}^{N} |m_i| \tag{28}$$

Combining the Geometrically-Aware Isotropic Regularization loss $\mathcal{L}_{\text{iso-reg}}$, the final loss function is defined as follows:

$$L = (1-\lambda_{ssim}) * L_{L1} + \lambda_{ssim} * L_{ssim} + \lambda_m * R_{mask} + \mathcal{L}_{\text{iso-reg}}, \tag{29}$$

where the weight λ_m is set to 0.0005. This mechanism allows the model to collaboratively optimize reconstruction quality and model sparsity during training, achieving smooth pruning with minimal impact on rendering quality.

4. Experiments

To comprehensively evaluate the performance of our proposed method across diverse scenarios, we conduct experiments on three representative public panoramic datasets. The first is the *OmniBlender* and *Ricoh360* datasets released by EgoNeRF [6]. Specifically, *OmniBlender* is a high-quality synthetic dataset rendered with the Blender [3] engine, covering both indoor and outdoor environments under idealized conditions. The camera follows a spiral-upward trajectory, providing a noise-free benchmark for algorithmic performance evaluation. In contrast, the *Ricoh360* dataset consists of real-world outdoor scenes captured by rotating the panoramic camera along a cross-shaped trajectory at fixed positions. The complex real-world lighting and textures in this dataset pose higher challenges for the robustness of reconstruction algorithms. In addition, we evaluate our method on the *360Roam* [8] dataset, which contains indoor roaming scenes. We follow the official train/test splits for all datasets. The resolutions are set to 2000×1000 for OmniBlender, 1920×960 for Ricoh360 and OmniPhotos, and 3040×1520 for 360Roam. For all datasets, camera poses and sparse initial point clouds are re-estimated using the Structure-from-Motion (SfM) algorithm implemented in OpenMVG [18].

We compare our approach against **widely recognized benchmarks that represent the distinct paradigms** in panoramic scene reconstruction, spanning both Neural Radiance Field (NeRF) and 3D Gaussian Splatting (3DGS) families. Specifically, we select *EgoNeRF* as the representative NeRF-based method, which introduces spherical feature grids for ego-centric panoramic reconstruction. For 3DGS-based methods, we adopt *OmniGS* [15] and

Table 1. Quantitative comparison across panoramic datasets. Best and second-best per metric are highlighted in **pink** and yellow, respectively.

Dataset	Method	PSNR [dB]↑	SSIM↑	LPIPS↓	FPS↑	Splats↓
Ricoh360	*EgoNeRF* [6]	24.52	0.747	0.337	3	N/a
	OmniGS [15]	25.27	0.817	0.206	**96**	**0.66M**
	ODGS [13]	23.57	0.789	0.237	74	1.29M
	Ours	**25.82**	**0.837**	**0.104**	92	0.82M
OmniBlender	*EgoNeRF* [6]	33.02	0.906	0.207	4	N/a
	OmniGS [15]	**34.75**	0.933	0.185	114	**0.44M**
	ODGS [13]	32.01	0.892	0.176	71	1.28M
	Ours	34.68	**0.940**	**0.053**	**116**	0.53M
360Roam	*EgoNeRF* [6]	20.81	0.694	0.380	2	N/a
	OmniGS [15]	**24.75**	0.806	**0.155**	**84**	**0.83M**
	ODGS [13]	20.83	0.746	0.295	49	2.14M
	Ours	22.49	**0.812**	0.193	77	1.17M

ODGS [13] as the **foundational frameworks**. These methods constitute the most established and widely acknowledged baselines for direct projection and tangent-plane approximation approaches respectively, providing a rigorous standard for evaluating real-time rendering quality and efficiency.

All experiments are conducted on a single NVIDIA A40 GPU, while inference speed (FPS) is evaluated on an NVIDIA RTX 4070 SUPER GPU to simulate consumer-level performance. To ensure fair comparison, all 3DGS-based frameworks, including OmniGS, ODGS, and our proposed method, are trained for 30,000 iterations using the Adam optimizer. EgoNeRF is trained for 200,000 iterations following its original setup.

Experiment Details. Our implementation builds upon ODGS with several hyperparameter refinements. For the dynamic densification strategy, we retain the threshold configuration described in their supplementary material. However, since the geometric scale of camera trajectories in panoramic datasets is typically much smaller than the effective visible depth range, directly adopting the original scale setting tends to over-densify the scene. Therefore, For the scene scale, we adjust the parameter from the ODGS setting of $e + 100$ to an empirically optimized value of $e \times 40$. Regarding the geometry-aware isotropic regularization loss, $\mathcal{L}_{\text{iso-reg}}$, we introduce it at the start of optimization and remove it at the $15,000$-iteration mark. Afterward, we introduce the LP-3DGS importance score mask loss, R_{mask}, which is applied until iteration $25,000$.

Table 2. Ablation study of our proposed components on the Bricks scene. Best and second-best values are highlighted in **pink** and yellow, respectively. BL: Baseline (ODGS); GO: Gradient Optimization; LP: Learning to Prune; Iso.: Isotropic Regularization.

Components				Metrics			
BL	GO	LP	Iso.	PSNR↑	SSIM↑	LPIPS↓	Splats↓
✓				22.75	0.760	0.192	1.34M
✓	✓			24.35	0.851	0.103	1.47M
✓	✓	Mini		24.76	0.826	0.112	**0.74M**
✓	✓	Rad		25.17	0.852	0.106	0.83M
✓	✓	**Rad**	✓	**25.52**	**0.867**	**0.091**	0.85M

4.1. Experiment Results

Quantitative Results

To validate the effectiveness of our proposed method, we evaluate the performance of all methods after training is complete. We use three widely recognized metrics to quantitatively assess the image quality of novel view synthesis: **Peak Signal-to-Noise Ratio (PSNR)** to measure pixel-level reconstruction accuracy, **Structural Similarity (SSIM)** [21] to evaluate the similarity between rendered and ground-truth images from a structural perspective, and **Learned Perceptual Image Patch Similarity (LPIPS)** [27] which utilizes deep features to measure differences at the perceptual level. Additionally, for methods based on 3DGS, we compare the number of Gaussians (**Splats**) and rendering speed (**FPS**) to evaluate the balance between storage efficiency, model complexity, and real-time rendering capabilities.

As shown in Table 1, the NeRF-based method, EgoNeRF, suffers from low inference speeds, failing to meet real-time requirements. In contrast, all 3DGS-based approaches achieve real-time rendering. Specifically, on the real-world Ricoh360 dataset, our method outperforms the others on all quality metrics. In the synthetic OmniBlender scenes, our PSNR is competitive with the state-of-the-art method, OmniGS.

It is worth noting that while our method maintains a higher splat count than OmniGS to preserve high-frequency details, it achieves comparable real-time frame rates. We attribute this efficiency to our geometry-aware regularization, which suppresses the needle-like ellipsoids prevalent in baseline methods. By enforcing geometric compactness, we effectively reduce the screen-space footprint and overdraw of individual Gaussians. Consequently, the computational overhead of sorting more points is offset by the significantly reduced burden on the rasterizer, enabling our method to deliver superior visual fidelity without compromising real-time performance. On both datasets, our method maintains a substantial lead in LPIPS, demonstrating that our approach achieves a better trade-off: a significant improvement in scene reconstruction quality with only a moderate increase in splats compared to OmniGS. This remarkable enhancement in perceptual quality is primarily attributed to our optimized gradient representation. However, on the 360Roam indoor roaming dataset, our method's reliance on the tangent plane approximation leads to less accurate representations for roaming scenarios, resulting in only a slight lead in SSIM.

Qualitative Results

Figure 6 showcases the novel view renderings from various methods on scenes from the Ricoh 360, OmniBlender, and 360Roam datasets. Our method achieves superior rendering fidelity compared to other baselines, particularly in capturing fine details within complex textured areas and along object edges. It more clearly reconstructs object contours and high-frequency textures while effectively reducing artifacts such as ghosting and blurry distortions.

In egocentric scenes such as "bricks" and "lonemonk," both EgoNeRF and OmniGS can reconstruct the overall architecture of the buildings and preserve the uniform tone of the sky, although EgoNeRF's results are slightly blurry. ODGS struggles with the reconstruction of edge structures, often producing fine, needle-like Gaussians that degrade sharpness. In contrast, our method demonstrates high rendering fidelity for both complex high-frequency details (including light and shadow transitions in dark areas) and distant regions like the "white tower."

In the roaming scene "lab," EgoNeRF and OmniGS excel at restoring scene illumination and high-frequency texture details. ODGS, however, suffers from cumulative errors arising from camera motion and tangent plane approximations, which leads to ghosting artifacts along object edges and missing content (e.g., ceiling lines), ultimately degrading its rendering fidelity. By applying isotropic regularization, our method effectively suppresses the extreme Gaussians that emerge with camera movement. While its performance does not fully match that of OmniGS in this case, it still preserves key scene details with high fidelity.

4.2. Ablation study

To validate the effectiveness of our proposed components, we conduct a series of ablation studies on the Bricks scene from the real-world Ricoh360 dataset. The quantitative results are presented in Table 2. Our analysis begins with the ODGS baseline, progressively integrating each of our contributions: Gradient Optimization (GO,see Section 3.2), the learning to prune module (LP), and the geometry-aware isotropic regularization loss ($\mathcal{L}_{\text{iso-reg}}$). As shown in the table 2, integrating our Gradient Optimization (+GO) module into the ODGS baseline yields a significant improvement across all quality metrics. This demonstrates the crucial role of stabilizing gradient propagation in panoramic scenes. We then compare two pruning strategies: one based on Mini-Splatting's[7] importance score (+GO+LP_{Mini}) and another on RadSplat's[19] score (+GO+LP_{Rad}). While both effectively reduce the number of Gaussians, the RadSplat-based approach better preserves high-fidelity details, which is reflected in superior quantitative scores. Finally, our full model, which incorporates the isotropic regularization loss (+GO+$\mathcal{L}_{\text{iso-reg}}$+$LP_{Rad}$), achieves the best overall performance, further enhancing all metrics to their peak values. This confirms that our regularization strategy effectively suppresses extreme needle-like Gaussians and enhances fine details.

Figure 7 shows the qualitative results of our ablation study on the lonemonk scene. Since LP is a compression strategy and does not visibly improve scene quality, it is excluded from this comparison. (a) The baseline ODGS renders blurry high-frequency texture details on surfaces like the suspended ceiling and columns. (b) After incorporating Gradient Optimization (GO), the rendering is improved; however, abnormal light spots are visible on the ceiling, and artifacts caused by extreme needle-like Gaussians appear along some edges. (c) With the further addition of Iso regularization, the high-frequency texture details become much clearer, resulting in a high-quality rendering.

5. Conclusion

This paper presents an efficient, high-fidelity 3D reconstruction framework for panoramic images, addressing the unstable gradients and geometric degradation that standard 3D Gaussian Splatting encounters with spherical projections.

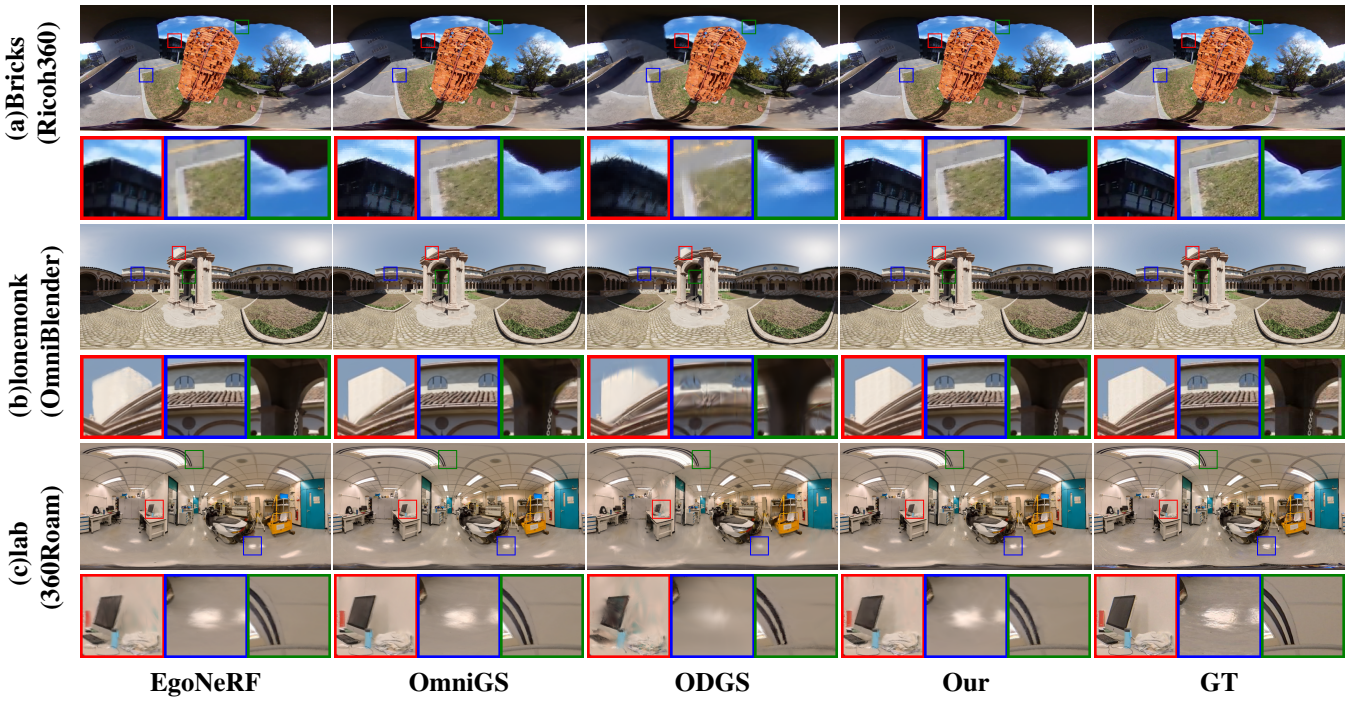

Figure 6. Qualitative comparison across datasets: (a) Ricoh360, (b) OmniBlender, and (c) 360Roam. Each column shows a different method, and each row corresponds to one scene. Our approach achieves sharper details and fewer artifacts, producing outputs closer to the ground truth.

Figure 7. Qualitative results of the ablation study on our proposed modules in the lonemonk scene from the OmniBlender dataset.(a) The original ODGS baseline.(b) The model with Gradient Optimization (GO) added.(c) The full model with the final addition of Iso, achieving a high-quality rendering.

To tackle these challenges, we first discuss the properties of gradient propagation in panoramic images and introduce a Gradient Optimization method for Continuous Panoramic Projection. This method mitigates gradient collision and improves optimization stability. We also design a geometry-aware isotropic regularization to suppress distortion artifacts and enhance detail, alongside an integrated learnable compression strategy to reduce model size while preserving quality. Experiments on public datasets show our method achieves state-of-the-art performance on metrics like PSNR, SSIM, and LPIPS, validating our framework's effectiveness.

Acknowledgement

This work was supported by the Key Research and Development Project in Xinjiang Uygur Autonomous Region (No.2022B01006).

References

[1] J. Bai, L. Huang, J. Guo, W. Gong, Y. Li, and Y. Guo. 360-gs: Layout-guided panoramic gaussian splatting for indoor roaming. In *2025 International Conference on 3D Vision (3DV)*, pages 1042–1053. IEEE, 2025. 3

[2] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. 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. 3

[3] Blender Online Community. Blender – a 3d modelling and rendering package, 2018. 9

[4] Z. Chen, Y.-P. Cao, Y.-C. Guo, C. Wang, Y. Shan, and S.-H. Zhang. Panogrf: Generalizable spherical radiance fields for wide-baseline panoramas. *Advances in Neural Information Processing Systems*, 36:6961–6985, 2023. 1

[5] X. Cheng, L. Wu, Z. Wang, J. Hou, J. Wen, and Y. Xu. Pvnet: Point-voxel interaction lidar scene upsampling via diffusion models. *IEEE Transactions on Image Processing*, 2025. 3

[6] C. Choi, S. M. Kim, and Y. M. Kim. Balanced spherical grid for egocentric view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16590–16599, 2023. 1, 3, 9, 10

[7] G. Fang and B. Wang. Mini-splatting: Representing scenes with a constrained number of gaussians. In *European Conference on Computer Vision*, pages 165–181. Springer, 2024. 8, 11

[8] H. Huang, Y. Chen, T. Zhang, and S.-K. Yeung. 360roam: Real-time indoor roaming using geometry-aware 360° radiance fields. *arXiv preprint arXiv:2208.02705*, 2022. 1, 3, 9

[9] H. Jang, A. Meuleman, D. Kang, D. Kim, C. Richardt, and M. H. Kim. Egocentric scene reconstruction from an omnidirectional video. *ACM Transactions on Graphics (TOG)*, 41(4):1–12, 2022. 3

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

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 2

[12] S. Kulkarni, P. Yin, and S. Scherer. 360fusionnerf: Panoramic neural radiance fields with joint guidance. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7202–7209. IEEE, 2023. 1

[13] S. Lee, J. Chung, J. Huh, and K. M. Lee. Odgs: 3d scene reconstruction from omnidirectional images with 3d gaussian splattings. *Advances in Neural Information Processing Systems*, 37:57050–57075, 2024. 1, 4, 9, 10

[14] J. Li, F. Hahlbohm, T. Scholz, M. Eisemann, J. Tauscher, and M. Magnor. Spags: Fast and accurate 3d gaussian splatting for spherical panoramas. In *Computer Graphics Forum*, volume 44, page e70171. Wiley Online Library, 2025. 3

[15] L. Li, H. Huang, S.-K. Yeung, and H. Cheng. Omnigs: Fast radiance field reconstruction using omnidirectional gaussian splatting. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2260–2268. IEEE, 2025. 1, 3, 9, 10

[16] Z. Liao, S. Chen, R. Fu, Y. Wang, Z. Su, H. Luo, L. Ma, L. Xu, B. Dai, H. Li, et al. Fisheye-gs: Lightweight and extensible gaussian splatting module for fisheye cameras. *arXiv preprint arXiv:2409.04751*, 2024. 3

[17] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1, 3

[18] P. Moulon, P. Monasse, R. Perrot, and R. Marlet. Openmvg: Open multiple view geometry. In *International Workshop on Reproducible Research in Pattern Recognition*, pages 60–74. Springer, 2016. 9

[19] M. Niemeyer, F. Manhardt, M.-J. Rakotosaona, M. Oechsle, D. Duckworth, R. Gosula, K. Tateno, J. Bates, D. Kaeser, and F. Tombari. Radsplat: Radiance field-informed gaussian splatting for robust real-time rendering with 900+ fps. In *2025 International Conference on 3D Vision (3DV)*, pages 134–144. IEEE, 2025. 8, 11

[20] C. Sun, C.-W. Hsiao, N.-H. Wang, M. Sun, and H.-T. Chen. Indoor panorama planar 3d reconstruction via divide and conquer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11338–11347, 2021. 2

[21] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 9

[22] L. Wu, X. Cheng, Y. Xu, H. Zeng, and J. Hou. Unsupervised 3d point cloud completion via multi-view adversarial learning. *IEEE Transactions on Visualization and Computer Graphics*, 2025. 3

[23] L. Wu, Q. Zhang, J. Hou, and Y. Xu. Leveraging single-view images for unsupervised 3d point cloud completion. *IEEE Transactions on Multimedia*, 2023. 3

[24] Q. Wu, J. M. Esturo, A. Mirzaei, N. Moenne-Loccoz, and Z. Gojcic. 3dgut: Enabling distorted cameras and secondary rays in gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26036–26046, 2025. 3

[25] Z. Ye, W. Li, S. Liu, P. Qiao, and Y. Dou. Absgs: Recovering fine details in 3d gaussian splatting. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 1053–1061, 2024. 4

[26] Z. Yu, T. Sattler, and A. Geiger. Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. *ACM Transactions on Graphics (ToG)*, 43(6):1–13, 2024. 4

[27] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 9

[28] Z. Zhang, T. Song, Y. Lee, L. Yang, C. Peng, R. Chellappa, and D. Fan. Lp-3dgs: Learning to prune 3d gaussian splatting. *Advances in Neural Information Processing Systems*, 37:122434–122457, 2024. 8