LongSplat: Online Generalizable 3D Gaussian Splatting from Long Sequence Images

Guichen Huang[1,2], **Ruoyu Wang**[3], **Xiangjun Gao**[4], **Che Sun**[2*], **Yuwei Wu**[1,2*], **Shenghua Gao**[3,5], **Yunde Jia**[2,1]

[1] Beijing Key Laboratory of Intelligent Information Technology, Beijing Institute of Technology, Beijing, China
[2] Guangdong Laboratory of Machine Perception and Intelligent Computing, Shenzhen MSU-BIT University, China
[3] Transcengram
[4] The Hong Kong University of Science and Technology
[5] The University of Hong Kong
huangguichen@bit.edu.cn,dwawayu@gmail.com,xgaobq@connect.ust.hk,sunche@smbu.edu.cn,
wuyuwei@bit.edu.cn,gaosh@hku.hk,jiayunde@smbu.edu.cn

Abstract

3D Gaussian Splatting (3DGS) achieves high-fidelity novel view synthesis, but its application in online long-sequence scenarios is still restricted. Existing methods either rely on slow per-scene optimization or lack efficient frame-wise 3DGS updates, making them unsuitable for online long-sequence videos. In this paper, we propose LongSplat, an online real-time 3D Gaussian reconstruction framework designed for long-sequence image input. The core idea of LongSplat is to maintain a global 3DGS set and design a streaming 3DGS update mechanism that selectively compressing redundant historical Gaussians and introducing new Gaussians by comparing the current observations with the historical Gaussian. To achieve this goal, we design a Gaussian-Image Representation (GIR), which encodes 3D Gaussian parameters into a structured, image-like 2D format. GIR simultaneously enables identity-aware redundancy compression as well as the fusion of current view and historical Gaussians, which are used for online reconstruction and adapt the model to long sequences without overwhelming memory or computational costs. Extensive experiments demonstrate that LongSplat achieves state-of-the-art efficiency-quality trade-offs in real-time novel view synthesis, delivering real-time reconstruction while reducing Gaussian counts by 44% compared to per-pixel prediction paradigms.

Introduction

Growing interest in 3D scene reconstruction and novel view synthesis has led to rapid advancements in the field, among which 3D Gaussian splatting (3DGS) (Kerbl et al. 2023; Yu et al. 2024; Huang et al. 2024; Lu et al. 2024) has gained particular attention for its effectiveness. Despite its impressive rendering speed at inference time, most existing methods still rely on slow per-scene optimization for reconstruction, which can take minutes to hours even for scenes of moderate size. This slow optimization is a significant barrier for applications requiring real-time perception and response, such as embodied AI and robotics, where online scene updates are essential. To address these challenges, there is an increasing need for systems that can process long sequences of visual data in real-time, incrementally updating with each new frame input while maintaining high-quality reconstruction.

Recent efforts have aimed to improve reconstruction efficiency by developing generalizable Gaussian splatting models that directly predict 3D Gaussian parameters from images in a feed-forward manner. These methods (Charatan et al. 2024; Xu et al. 2024; Li et al. 2025) significantly reduce processing time and perform well under sparse-view settings. However, their performance often degrades when applied to long sequences or dense multi-view scenarios: the reconstructed Gaussians become increasingly redundant and noisy, resulting in artifacts such as floating points and blurred regions. Moreover, memory and computational costs grow rapidly as more views are processed, making these approaches difficult to scale to real-world applications involving hundreds of frames. These limitations arise primarily from two factors: a lack of global historical Gaussians modeling and the absence of an efficient incremental update mechanism, both of which are essential for robust long-term reconstruction. Although some recent works (Wang et al. 2025a, 2024, 2025b; Ziwen et al. 2024; Li et al. 2025) extending generalizable 3D GS to sequential inputs sets, they still struggle with incremental updates or rely on fixed-length reconstruction pipelines, limiting their flexibility and scalability in online long-sequence scenarios.

In this paper, we propose LongSplat, an online 3DGS framework designed for real-time, incremental reconstruction from long-sequence images. Its core innovation lies in an incremental update mechanism that integrates current-view observations while selectively compressing redundant historical Gaussian. This mechanism efficiently performs two key operations per frame: (1) Adaptive Compression: selectively compressing accumulated Gaussians from past views to eliminate redundancy and minimize storage/rendering costs, and (2) Online Integration: fusing current-view Gaussians with the historical state. These strategies aim to mitigate a core limitation of generalizable 3DGS: per-pixel prediction inherently produces dense but redundant Gaussians. By progressively refining the Gaussian field over

*Corresponding author.

time, our method seeks to improve scalability and memory usage while enhancing consistency across views. In addition, the compression mechanism reduces redundancy and offers a potential path toward dynamic scene modeling, where outdated or redundant elements can be removed in a lightweight, incremental manner without reprocessing the entire sequence.

Specifically, we propose Gaussian-Image Representation (GIR) that projects 3D Gaussian parameters into a structured 2D image-like format. This representation enables online reconstruction by facilitating the propagation of information across views and supporting localized compression. To enhance cross-view interaction, GIR projects historical Gaussians into the current frame, enabling feature-level fusion. This fusion not only improves the spatial consistency of the reconstructed 3D Gaussian field, but also provides a structured basis for subsequent compression of redundant historical information. In addition, GIR plays a central role in localized compression by maintaining the mapping between 2D projections and their corresponding historical 3D Gaussians. This identity-aware structure makes 3DGS more tractable and removes redundant splats accumulated over time. Such compression not only reduces memory and rendering cost, but also improves visual quality by eliminating overlapping or outdated Gaussians. Furthermore, we leverage GIR's image-like structure to apply supervision from reference 3DGS, using an optimized per-scene Gaussian dataset constructed with existing image compression techniques (Fan et al. 2024). This strategy improves both compactness and fidelity of the learned 3D Gaussians without requiring full 3D loss computation.

Through extensive evaluations, we demonstrate that LongSplat achieves state-of-the-art efficiency-quality tradeoffs in real-time novel view synthesis. Our method achieves real-time rendering and reduces Gaussian counts by 44% on DL3DV(Ling et al. 2024). Moreover, LongSplat outperforms the baseline by 3.6 dB(Xu et al. 2024) in PSNR on the DL3DV benchmark and exhibits superior scalability for long-sequence scene reconstruction. Through effective online compression of 3DGS, LongSplat enables scalable, efficient, and high-quality real-time reconstruction. Our contributions can be summarized as follows:

- We propose LongSplat, a real-time 3D Gaussian reconstruction framework tailored for arbitrary-view, long-sequence image inputs. By introducing a mechanism that compresses historical redundancy while integrating new views, LongSplat enables scalable, memory-efficient reconstruction and real-time synthesis.
- We introduce Gaussian-Image Representation (GIR), a structured 2D representation of 3D Gaussians that enables efficient historical feature fusion, redundancy compression, lightweight 2D operations, and GIR-space supervision.
- Extensive experiments show that LongSplat achieves state-of-the-art real-time novel view synthesis, improving PSNR by 3.6dB and reducing Gaussian counts by 44% compared to the baseline method(Xu et al. 2024).

Related Work

Traditional 3D Gaussian Splatting. Traditional 3D Gaussian Splatting (3DGS) methods (Kerbl et al. 2023; Yu et al. 2024; Huang et al. 2024; Lu et al. 2024; Wang, Ma, and Gao 2025; Gao et al. 2025) have emerged as a powerful paradigm for high-fidelity novel view synthesis, leveraging explicit 3D Gaussian primitives to represent scenes. Unlike Neural Radiance Fields (NeRFs) (Mildenhall et al. 2021; Barron et al. 2021, 2023; Fridovich-Keil et al. 2023; Cao and Johnson 2023; Liu et al. 2022; Wang et al. 2021), which rely on computationally intensive ray-marching-based volume rendering, 3DGS achieves real-time rendering speeds through tile-based rasterization of differentiable Gaussian primitives. These methods optimize Gaussian parameters (e.g., position, scale, rotation, and opacity) through per-scene optimization, resulting in high-quality novel view synthesis and fast rendering. However, the per-scene optimization process is inherently time-consuming, which significantly limits its applicability in real-time perception tasks.

Generalizable 3D Gaussian Splatting. Inspired by prior progress in generalizable NeRFs (Yu et al. 2021; Lin et al. 2022; Gao et al. 2022; Chen et al. 2021; Ye, Wang, and Wang 2023), recent efforts have focused on generalizable 3D Gaussian Splatting methods that enable feed-forward prediction of 3D Gaussians from input images(Roh et al. 2024; Sheng et al. 2025; Bao et al. 2024; Yan et al. 2025; Kang et al. 2025; Chen et al. 2025). Pioneering works such as PixelSplat (Charatan et al. 2024) and GPS-Gaussian (Zheng et al. 2024) explore feed-forward 3D Gaussian reconstruction using epipolar geometry from just two input views, achieving fast and high-quality novel view synthesis. MVS-based methods (Chen et al. 2024a; Xu et al. 2024; Li et al. 2025) extends this direction by leveraging multi-view geometry through cost volumes for enhanced accuracy and generalization. Adaptive Gaussian (Fei et al. 2024) departs from fixed pixel-wise Gaussian representations by dynamically adapting the distribution and number of 3D Gaussians based on local geometric complexity.

Other approaches extend these approaches to sequential inputs: For example, FreeSplat (Wang et al. 2024, 2025b) proposes a cross-view aggregation scheme and a pixel-wise triplet fusion strategy that jointly optimizes overlapping view regions, enabling free-view synthesis with geometrically consistent scene reconstruction. Yet, due to its latent GS representation, the heavy computational overhead limits scalability to long-sequence inputs, making it less suitable for real-time processing of long-sequence inputs. Long-LRM (Ziwen et al. 2024) leverages a hybrid architecture merging and Gaussian pruning—to process up to 32 views in a single feed-forward pass, reconstructing entire scenes with performance comparable to optimization-based methods. Despite these advances, its reliance on fixed-length reconstruction limits flexibility, making it unsuitable for dynamic, open-ended sequences. Zpressor (Wang et al. 2025a) significantly reduces memory requirements via anchor-frame propagation while achieving high reconstruction quality. Compared to such anchor-frame methods that still rely on per-frame prediction and fixed-feature transfer,

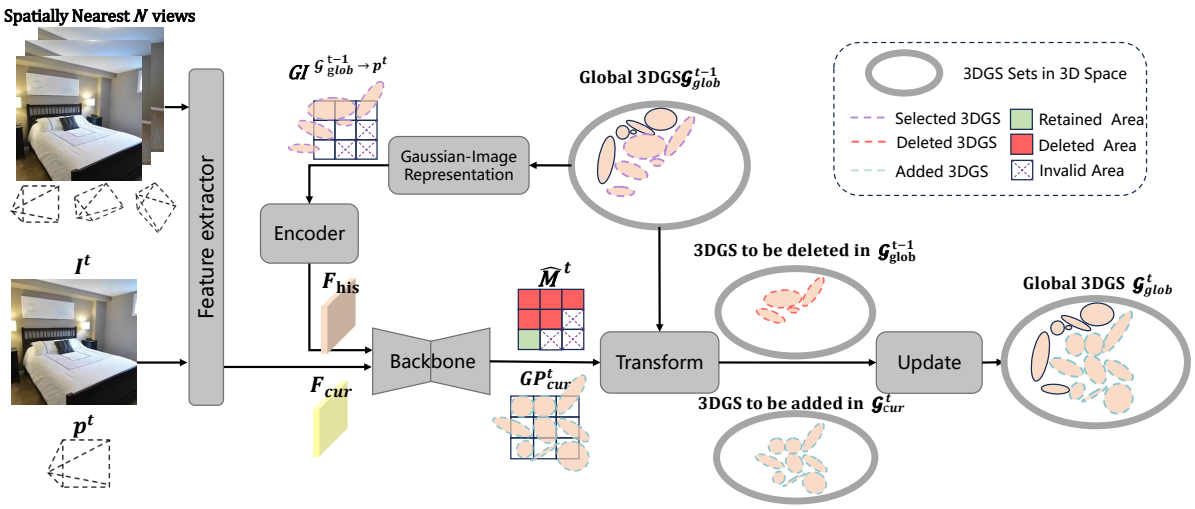

Figure 1: Overview of the framework. Given a posed images sequence $\{I^t, p^t\}_{t=1}^T$, the model progressively builds a global Gaussian representation $\mathcal{G}_{\text{glob}}$. At each timestep t, we extract spatial features $\mathbf{F}_{\text{cur}}$ from the current frame and spatially nearby N views, and historical features $\mathbf{F}_{\text{his}}$ by rendering the previous global Gaussians $\mathcal{G}_{\text{glob}}^{t-1}$ into view p^t as a 2D Gaussian-Image Representation (GIR) $\mathbf{GI}^{\mathcal{G}_{\text{glob}}^{t-1} \to p^t}$. These features are fused by a transformer-based backbone to predict a binary update mask $\hat{\mathcal{M}}^t$ and per-pixel Gaussian parameters $\mathbf{GP}_{\text{cur}}^t$. A transform module then (1) uses the mask to identify redundant Gaussians in $\mathcal{G}_{\text{glob}}^{t-1}$ via lookup, and (2) lifts $\mathbf{GP}_{\text{cur}}^t$ into a new 3D set $\mathcal{G}_{\text{cur}}^t$. The updated global representation $\mathcal{G}_{\text{glob}}^t$ is obtained by removing outdated Gaussians and inserting the new ones, enabling consistent and compact scene modeling over time.

our approach supports dynamic updates to Gaussians across frames, enabling better use of historical information. As a per-pixel predictor, it is also compatible with our framework and can serve as a feature encoder within our fusion pipeline. We propose LongSplat, an online 3D Gaussian reconstruction framework specifically designed for long-sequence inputs, supporting scalable temporal modeling under streaming and interactive conditions. Our approach enables real-time editing and streaming integration without compromising reconstruction fidelity through 3DGS updating and history view fusion techniques.

Method

Vanilla 3D Gaussian Splatting

The vanilla 3D Gaussian Splatting (3DGS) represents scenes as a collection of anisotropic Gaussians $\mathcal{G} = \{\mu, \Sigma, c, \alpha\}$, where μ denotes position, Σ the covariance matrix, c the color, and α the opacity. The rendering process follows alpha compositing along each ray:

$$C(u,v) = \sum_{i \in \mathcal{S}(u,v)} c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j), \tag{1}$$

where $\mathcal{S}(u,v)$ represents the set of Gaussians sorted by depth. The Gaussian parameters are optimized by a photometric loss to minimize the difference between renderings and image observations.

Longsplat Pipeline

We propose Longsplat, a method that incrementally constructs a global 3D Gaussian scene representation $\mathcal{G}_{\text{g}}$ from an input posed image sequence $\{I^t, p^t\}_{t=1}^T$. At each timestep t, the model processes the current frame I^t by jointly leveraging (1) multi-view spatial features $\mathbf{F}_{\text{cur}}$ extracted from the current image and its spatial N nearest views, and (2) global context features $\mathbf{F}_{\text{his}}$ derived from $\mathbf{GI}^{\mathcal{G}_{\text{glob}}^{t-1} \to p^t}$ — the Gaussian-Image Representation (GIR) rendered by projecting the previous global Gaussians $\mathcal{G}_{\text{glob}}^{t-1}$ into the current view. Based on these features, our backbone module predicts a set of per-pixel Gaussian predictions $\mathbf{GP}_{\text{cur}}^t$ and a binary mask $\hat{\mathbf{M}}^t$ that indicates Gaussians to be removed. Using this mask, we remove redundant or outdated Gaussians in $\mathcal{G}_{\text{glob}}^{t-1}$ and incorporate the new predictions $\mathbf{GP}_{\text{cur}}^t$ into the global set $\mathcal{G}_{\text{cur}}^t$, producing the updated scene representation $\mathcal{G}_{\text{glob}}^t$. To support removal, our GIR encodes a unique ID for Seach projected 3D Gaussian, enabling direct and reliable deletion from the global representation via its image-space projection.

Feature extractors. We extract two feature streams $\mathbf{F}_{\text{cur}}$ and $\mathbf{F}_{\text{his}}$ to support per-frame updates: (1) Multi-view Spatial Features $\mathbf{F}_{\text{cur}}$. We directly utilize the outputs from DepthSplat(Xu et al. 2024) as inputs, including the feature representations extracted from the current frame and its N nearest frames and the raw Gaussian proposals. This feature encodes local 3D geometry and appearance cues, providing strong priors for the current-view reconstruction. (2) Historical Feature Map $\mathbf{F}_{\text{his}}$. To integrate historical context, we employ our Gaussian-Image Representation (GIR) to project the accumulated global Gaussians $\mathcal{G}_{\text{glob}}^{t-1}$ into the current camera view. The resulting 2D projection $\mathbf{GI}^{\mathcal{G}_{\text{glob}}^{t-1} \to p^t}$ is then encoded by a shallow network to produce the historical

feature map $\mathbf{F}_{\text{his}}$, which captures global scene structure from prior observations.

Backbone

Our backbone module jointly performs three core functions: History Fusion, Compressed Fusion, and Gaussian Parameter Prediction. Specifically, the two feature streams, $\mathbf{F}_{\text{cur}}$ and $\mathbf{F}_{\text{his}}$, are first fused via the History Fusion module to produce a context-aware representation $\mathbf{F}_{\text{fuse}}$, along with a soft update mask that is later thresholded into the binary history mask $\hat{\mathbf{M}}^t$. Then, $\mathbf{F}_{\text{fuse}}$ and $\hat{\mathbf{M}}^t$ are jointly processed by the Compressed Fusion module to produce the final per-pixel feature embedding $\mathbf{F}_{\text{final}}$, which implicitly encodes both the fused spatio-temporal context and the Gaussian removal decisions. Finally, $\mathbf{F}_{\text{final}}$ is used to predict a structured tensor of per-pixel Gaussian parameters predictions $\mathcal{GP}^t_{\text{cur}}$.

History Fusion. To ensure consistent reconstruction and enable redundancy detection, we introduce a History Fusion module that integrates current-view features $\mathbf{F}_{\text{cur}}$ with historical context $\mathbf{F}_{\text{his}}$.

$$\begin{aligned} \mathbf{F}_{\text{fusion}} &= \text{Transformer}(\mathbf{F}_{\text{cur}}, \mathbf{F}_{\text{his}}), \\ \mathbf{M}^t_{\text{pred}} &= \text{Sigmoid}(\text{Head}(\mathbf{F}_{\text{fusion}})). \end{aligned} \tag{2}$$

These features are fused via a transformer-based attention mechanism, producing a unified representation $\mathbf{F}_{\text{fusion}}$ that encodes both current-view appearance and temporal priors. To explicitly indicate redundant Gaussians for removal while retaining essential Gaussians, we predict a soft update mask $\hat{\mathbf{M}}^t_{\text{pred}} \in [0,1]^{H\times W}$ from $\mathbf{F}_{\text{fusion}}$. A masking threshold τ is then applied to $\hat{\mathbf{M}}^t_{\text{pred}}$ to determine the final binary mask $\hat{\mathbf{M}}^t$ for Gaussian removal.

Compressed Fusion. With the binary history mask $\hat{\mathbf{M}}^t$ indicating which historical Gaussians are retained or removed, we guide the generation of a task-aware embedding via another transformer:

$$\mathbf{F}_{\text{final}} = \text{Transformer}_{\text{final}}(\mathbf{F}_{\text{fusion}}, \hat{\mathbf{M}}^t). \tag{3}$$

This mask provides explicit supervision to the feature generator, prompting it to (1) refine the retained Gaussians for improved quality, and (2) synthesize new Gaussians to fill the gaps left by removed ones. By conditioning on this spatially structured prior, $\mathbf{F}_{\text{final}}$ enables accurate and compact per-pixel Gaussian prediction, leading to efficient and consistent scene updates.

Gaussian Parameter Prediction. We follow DepthSplat (Xu et al. 2024) to initialize Gaussian centers by unprojecting the predicted depth maps into 3D. Instead of computing depth independently, we directly use DepthSplat's predicted depth as initialization and add a learnable offset Δd. The remaining Gaussian parameters — opacity, covariance, and color — are predicted by an additional DPT head that takes the encoded feature map $\mathbf{F}_{\text{final}}$ as input.

Gaussian-Image Representation

We propose a *Gaussian-Image Representation* (GIR) that encodes per-pixel Gaussian attributes into a structured 2D format. We note prior image-based and projection-aware designs for 3D data (Szymanowicz, Rupprecht, and Vedaldi 2024; Wang et al. 2022). This compact view-aligned representation enables efficient memory usage, supports localized updates, and bridges the gap between 3D scene modeling and 2D image-space supervision.

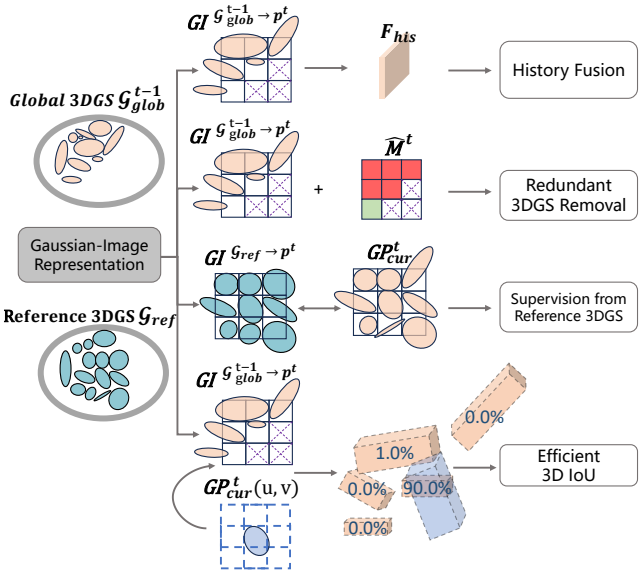

Figure 2: Overview of the proposed **Gaussian-Image Representation (GIR)** and its four core capabilities. GIR encodes per-pixel Gaussian parameters into a structured 2D image space, enabling efficient and flexible 3D reasoning.

Formally, for each pixel (u, v) in a rendered view, the Gaussian-Image $\mathbf{GI} \in \mathbb{R}^{H\times W\times C}$ stores the 3D position $\boldsymbol{\mu}_{\text{uv}}$, the upper-triangular components of the covariance matrix $\text{vec}(\boldsymbol{\Sigma}_{\text{uv}})$, opacity α_{uv}, color c_{uv}, and a unique Gaussian identifier ID_{uv}:

$$\mathbf{GI}(u,v) = [\,\boldsymbol{\mu}_{\text{uv}},\ \text{vec}(\boldsymbol{\Sigma}_{\text{uv}}),\ \alpha_{\text{uv}},\ c_{\text{uv}},\ ID_{\text{uv}}\,]\,. \tag{4}$$

Unlike standard 3D Gaussian Splatting (3DGS), which blends all overlapping Gaussians along a ray, our GIR adopts a sparse rendering strategy in which each pixel is associated with only a single dominant Gaussian. We consider two selection methods:

(1) Nearest Rendering. The first visible Gaussian (with opacity above threshold θ) is selected:

$$k = \arg\min_i \{d_i \mid \alpha_i > \theta\}\,. \tag{5}$$

(2) Most-Contributive Rendering. The Gaussian that contributes most to the final color along the ray is chosen, based on transmittance-weighted opacity:

$$k = \arg\max_i \left(\alpha_i \prod_{j=1}^{i-1}(1-\alpha_j)\right). \tag{6}$$

By projecting 3D Gaussians into a structured 2D image space, our GIR representation enables better spatial

alignment, more efficient operations, and per-pixel supervision—all of which are difficult to achieve directly in 3D. It also allows deterministic Gaussian lookup via unique IDs, supporting updates across time and views. Specifically, GIR provides four key functional advantages: **(1) History Fusion.** Enables 2D feature-level fusion between current and historical views for temporal consistency. **(2) Redundant Gaussian Removal.** Encoded Gaussian IDs allow direct lookup and removal of outdated Gaussians via 2D projections. **(3) 3D Gaussian Dataset Supervision.** Transforms optimized 3D Gaussians into 2D supervision targets, supporting per-pixel training (detailed in §Training). **(4) Efficient 3D IoU.** GIR enables sliding-window-style heuristics in 2D to accelerate 3D IoU-guided redundancy detection (detailed in §Training). Together, these functionalities make GIR a unified and efficient bridge between 3D Gaussian representations and scalable 2D learning and processing.

Training

We design a training framework with three loss functions: a RGB reconstruction loss for supervision over the appearance and spatial attributes of 3DGS, a mask-guided loss to remove redundant 3DGS, and a geometry alignment loss to improve compactness and fidelity of 3DGS.

RGB Reconstruction Loss. To supervise novel view synthesis, we render the predicted Gaussians $\mathcal{G}_{\text{glob}}$ into a set of randomly sampled target views $\{p^t\}_{t=1}^K$ as $\{\hat{I}^t_{\text{render}}\}_{t=1}^K$ and compare them with the corresponding ground truth images $\{I^t\}_{t=1}^K$ using a combination of MSE and LPIPS losses, as is typically done in prior work:

$$\mathcal{L}_{\text{RGB}} = \sum_{t=1}^{M} (\|\hat{I}^t_{\text{render}} - I^t\|_{\text{MSE}} + \lambda_{\text{L}} \cdot \text{LPIPS}(\hat{I}^t_{\text{render}}, I^t)). \quad (7)$$

This loss encourages both photometric accuracy and perceptual quality in the synthesized views, guiding Gaussian attributes like color, position, and opacity.

Mask-Guided Loss. Long-sequence reconstruction often results in redundant Gaussians across frames. To suppress them, we predict a visibility mask $\hat{\mathbf{M}}^t_{\text{pred}} \in [0,1]^{H\times W}$ that removes outdated or unnecessary Gaussians during rendering. We observe that such redundant Gaussians typically exhibit high spatial overlap with those in previous frames. To detect this, we compute a pairwise 3D overlap score based on Oriented Bounding Boxes (OBBs)—the minimal rectangles enclosing each 3D Gaussian ellipsoid. Unlike standard symmetric IoU, we adopt a view-asymmetric metric:

$$\text{IoU}_{\text{uv}} = \max_{i\in\mathcal{N}} \frac{|\text{OBB}_{\text{uv}} \cap \text{OBB}_i|}{|\text{OBB}_{\text{uv}}|}. \quad (8)$$

This formulation identifies outdated Gaussians whose OBBs redundantly cover current view details, allowing them to be masked out in favor of finer, view-specific ones. Thanks to GIR, these 3D overlap checks are implemented as local, grid-aligned operations, enabling efficient, parallel GPU execution without global scene traversal.

To supervise the visibility mask $\hat{\mathbf{M}}^t_{\text{pred}}$, we label Gaussians with high 3D overlap as redundant and assign corresponding reference $\mathbf{M}^t_{\text{ref}} = 0$, while others are assigned $\mathbf{M}^t_{\text{ref}} = 1$. Specifically, for each Gaussian proposal $\mathbf{GP}^t_{\text{cur}}(u,v)$, we evaluate its 3D IoU with Gaussians from the $\mathbf{GI}^{\mathcal{G}^{t-1}_{\text{glob}}\to p^t}$ within a 3×3 kernel centered at pixel (u,v)—i.e., those indexed in $[u-1,u+1]\times[v-1,v+1]$. A weighted binary cross-entropy loss is applied over all pixels:

$$\mathcal{L}_{\text{mask}} = \frac{1}{|\Omega|} \sum_{\mathbf{p}\in\Omega} \lambda_{\text{ref}} \cdot \text{BCE}\left(\hat{\mathbf{M}}^t_{\text{pred}},\ \mathbf{M}^t_{\text{ref}}\right), \quad (9)$$

where Ω is the set of all pixel locations, and the weight function is defined as:

$$\lambda_{\text{ref}}(\mathbf{M}^t_{\text{ref}}) = \begin{cases} \lambda_{\text{pos}}, & \text{if } \mathbf{M}^t_{\text{ref}}(u,v) = 1 \\ \lambda_{\text{neg}}, & \text{if } \mathbf{M}^t_{\text{ref}}(u,v) = 0. \end{cases} \quad (10)$$

While high 3D overlap often indicates redundancy, some Gaussians with large overlaps are essential for capturing fine-grained details. To avoid mistakenly discarding such informative components, we couple the visibility mask $\hat{\mathbf{M}}^t_{\text{pred}} \in [0,1]^{H\times W}$ with the rendered opacity through a modulation rule:

$$\hat{\alpha}_{uv} = \hat{\mathbf{M}}^t_{\text{pred}}(u,v) \cdot \alpha_{uv}, \quad \text{if } \hat{\mathbf{M}}^t_{\text{pred}}(u,v) < \tau. \quad (11)$$

This design enables soft pruning: Gaussians deemed redundant by geometric overlap will be assigned lower mask values, reducing its opacity. However, if removing such Gaussians causes a noticeable degradation in reconstruction quality—as reflected in an increased RGB loss—the model is incentivized to raise their opacity, which in turn increases their mask value. Thus, the model learns to retain visually important Gaussians while suppressing truly redundant ones.

Geometry Alignment Loss. Per-scene optimized Gaussians generally exhibit higher quality and compactness compared to generalizable prediction-based methods. To leverage this advantage, we construct a reference dataset $\mathcal{G}_{\text{ref}}$ using LightGaussian (Fan et al. 2024), an existing 3D Gaussian compression method that performs per-scene optimization. These optimized Gaussians serve as strong structural references to guide our model during training.

We transform reference Gaussians $\mathcal{G}_{\text{ref}}$ into view p^t via GIR, yielding the 2D image-like structure $\mathbf{GI}^{\mathcal{G}_{\text{ref}}\to p^t}$. We supervise alignment between the predicted Gaussians $\mathbf{GP}^t_{\text{cur}}$ and the $\mathbf{GI}^{\mathcal{G}_{\text{ref}}\to p^t}$ using two lightweight losses. A position alignment loss ensures spatial consistency:

$$\mathcal{L}_{\mu} = \frac{1}{|T|} \sum_{t\in T} \|\hat{\boldsymbol{\mu}}^t_{\text{pred}} - \boldsymbol{\mu}^t_{\text{ref}}\|_1; \quad (12)$$

A covariance loss encourages shape consistency:

$$\mathcal{L}_{\Sigma} = \frac{1}{|T|} \sum_{t\in T} \|\hat{\boldsymbol{\Sigma}}^t_{\text{pred}} - \boldsymbol{\Sigma}^t_{\text{ref}}\|_1. \quad (13)$$

Here, $\boldsymbol{\mu}^t_{\text{pred}}, \boldsymbol{\Sigma}^t_{\text{pred}}$ are extracted from $\mathbf{GP}^t_{\text{cur}}$, and $\boldsymbol{\mu}^t_{\text{ref}}, \boldsymbol{\Sigma}^t_{\text{ref}}$ from $\mathbf{GI}^{\mathcal{G}_{\text{ref}}\to p^t}$. The total geometry alignment loss is:

$$\mathcal{L}_{\text{geo}} = \mathcal{L}_{\mu} + \lambda_{\Sigma}\mathcal{L}_{\Sigma}. \quad (14)$$

Furthermore, we discard supervision from Gaussians with excessively large scales or significant center offsets, which are unsuitable for per-pixel prediction.

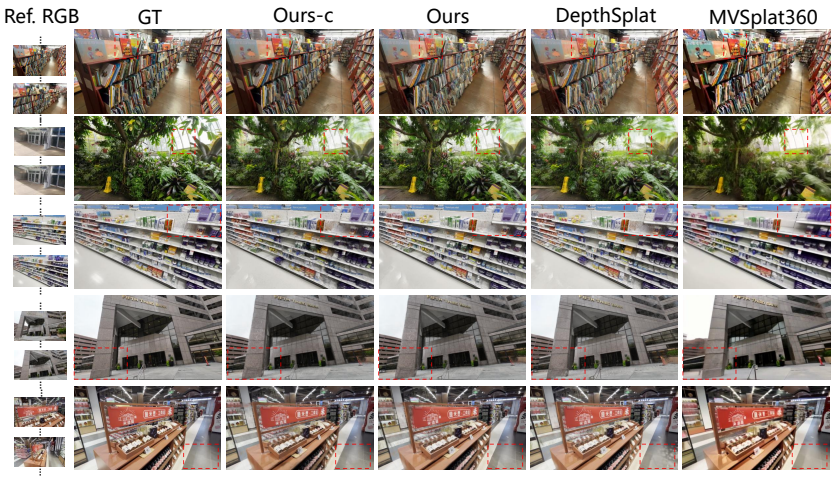

Figure 3: Novel view synthesis on 12 context views. Ref. RGB, GT are the input image and the ground truth. Ours-c delete the mask regions. Ours, MVSplat360, and DepthSplat are full-scene results. Our method better removes floaters and preserves fine details, producing more accurate and consistent renderings.)

Experiments

Implementation Details. We adopt pretrained DepthSplat(Xu et al. 2024) as our baseline framework while keeping all its parameters fixed during training. The feature representations and Gaussian splatting outputs from DepthSplat are directly utilized as our model inputs. During rendering, we consistently use 10 target views for loss computation, ensuring multi-view consistency in the optimization process. For optimization, we employ the AdamW optimizer with a base learning rate of 1×10^{-4}.

Training Datasets We conduct training on the DL3DV-10K(Ling et al. 2024) dataset, which consists of 9,896 training scenes and 140 official test scenes. Additionally, we provide optional auxiliary data reconstructed through DepthSplat (24-view inputs) with LightGaussian compression, which contributes to conditional auxiliary losses when available during training. We selected 6,845 scenes meeting our quality criteria: compression rate >30% and PSNR >28.0.

DL3DV Benchmark Evaluation

Quantitative Results. As shown in Table 1, our method outperforms DepthSplat across 12, 50, and 120 view settings in both quality and compactness. At 12 views, we achieve 22.68 PSNR vs. 22.02 for DepthSplat. Our compressed variant (Ours-c) retains 21.69 PSNR with 25.52% fewer Gaussians. As view count increases, DepthSplat degrades (17.77 PSNR at 120 views) due to uncontrolled Gaussian growth. In contrast, our method remains stable: at 50 views, Ours-c reaches 23.54 PSNR with 43.77% compression; at 120 views, it maintains 21.34 PSNR with 44.37% compression. These results confirm the scalability and efficiency of our approach for long-sequence reconstruction.

Qualitative Results. Figure 3 shows qualitative results on several DL3DV scenes with 12 input views. We compare our method (Ours), its compressed variant (Ours-c), MVsplat360(Chen et al. 2024b), and our baselines, DepthSplat(Xu et al. 2024). DepthSplat tends to produce floaters and blurred surfaces in complex regions, while MVsplat360 occasionally exhibits structural inconsistencies when applied to long sequences. Our method generally produces sharper and more consistent reconstructions, with clearer geometry and fewer visual artifacts. Notably, the compressed variant (Ours-c) maintains comparable quality, and in some cases—such as the first row—shows improved clarity in fine details like book spines. This suggests that removing low-confidence Gaussians may help reduce visual clutter and enhance overall fidelity.

Views	Method	PSNR↑	SSIM↑	LPIPS↓	c-ratio↑
12	MVSplat-360	17.05	0.4954	0.3575	0.00%
	DepthSplat	22.02	0.7609	0.2060	0.00%
	Ours	22.68	0.7824	0.1923	0.00%
	Ours-c	21.69	0.7482	0.2213	25.52%
50	MVSplat-360	OOM	/	/	/
	DepthSplat	21.39	0.7341	0.2212	0.00%
	Ours	23.71	0.8159	0.1683	0.00%
	Ours-c	23.54	0.8056	0.1742	43.77%
120	DepthSplat	17.77	0.5899	0.3622	0.00%
	Ours	21.02	0.7176	0.2608	0.00%
	Ours-c	21.34	0.7345	0.2449	44.37%

Table 1: Performance comparison with SOTA methods

Ablation and Analysis

Ablation Study on Component-wise Contributions. Table 2 reports the incremental impact of each core module. Adding Unet refinement (**U**) to the baseline improves PSNR from 21.39dB to 21.71dB. History fusion (**F**) brings a larger boost to 22.78dB PSNR and significantly lowers LPIPS, highlighting the value of temporal context. Introducing 3D dataset supervision (**G**) further enhances alignment, reaching 23.12 PSNR. We then examine compression and mask-

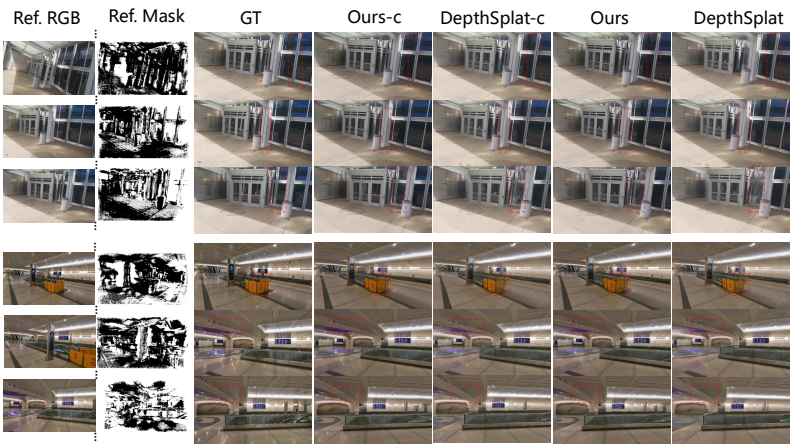

Figure 4: Novel view synthesis on 50 context views. Ref. RGB and Ref. mask are the input image and compression mask. GT is the ground truth. Ours-c and DepthSplat-c delete the mask regions. Ours and DepthSplat are full-scene results. Our method better removes floaters and preserves fine details, producing more accurate and consistent renderings.)

Config	Components					Metrics			
	U	F	G	C	M	PSNR↑	SSIM↑	LPIPS↓	c-ratio↑
Baseline	–	–	–	–	–	21.39	0.7341	0.2212	0.00
+ Unet	✓	–	–	–	–	21.71	0.7463	0.2172	0.00
+ Fusion	✓	✓	–	–	–	22.78	0.7934	0.1777	0.00
+ 3D Dataset	✓	✓	✓	–	–	23.12	0.8087	0.1679	0.00
+ 3D Dataset(M)	✓	✓	✓	–	✓	22.47	0.7793	0.1851	43.77
+ Compression	✓	✓	✓	✓	–	23.71	0.8159	0.1683	0.00
+ Compression(M)	✓	✓	✓	✓	✓	23.54	0.8056	0.1742	43.77

Table 2: Ablation study on core modules

τ	Method	PSNR↑	SSIM↑	LPIPS↓	c-ratio↑
No Mask	DepthSplat	21.39	0.7341	0.2212	0.0%
	Ours	**23.71**	**0.8159**	**0.1683**	0.0%
0.1	DepthSplat	21.82	0.7497	0.2081	23.32%
	Ours	**23.66**	**0.8141**	**0.1680**	23.32%
0.3	DepthSplat	22.17	0.7621	0.1996	38.11%
	Ours	**23.61**	**0.8092**	**0.1716**	38.11%
0.5	DepthSplat	22.32	0.7668	0.1964	43.77%
	Ours	**23.54**	**0.8056**	**0.1742**	43.77%

Table 3: Performance under varying masking thresholds τ.

ing. Applying a fixed compression mask on the 3D dataset (G&M) reduces Gaussians by 43.77% but lowers PSNR to 22.47dB, indicating that static pruning can discard relevant points. Introducing the learned compression fusion (**C**) without masking restores high fidelity (23.71dB, SSIM 0.8159, LPIPS 0.1683) while keeping c-ratio at 0%. Finally, combining compression with adaptive masking (C&M) retains most of this gain (PSNR 23.54dB, SSIM 0.8056, LPIPS 0.1742) at a 43.77% compression ratio. These results validate that each component contributes complementary gains, with joint compression and masking offering the best balance of quality and compactness.

Evaluation of Masking Strategies Under Confidence Thresholds. We evaluate how the masking threshold τ impacts reconstruction quality and compression (Table 3). For DepthSplat, increasing τ slightly improves PSNR and LPIPS, but this mainly results from removing artifacts like floaters, rather than actual improvements in reconstruction. In contrast, our method learns a confidence-based mask that selectively removes redundant or noisy Gaussians. Even at a high compression rate (e.g., $\tau = 0.5$, 43.77%), our model maintains high fidelity (23.54 PSNR, 0.1742 LPIPS), closely matching the uncompressed setting. This shows our approach enables precise, structure-aware compression that preserves both geometry and appearance. Visual results (Fig. 4) confirm that our method eliminates floaters while retaining fine details, whereas DepthSplat often oversmooths or distorts compressed regions.

Conclusion

We present LongSplat, a real-time 3D Gaussian Splatting framework tailored for long-sequence reconstruction. To address scalability and redundancy issues in existing feed-forward pipelines, LongSplat introduces an incremental update mechanism that compresses redundant Gaussians and incrementally integrates current-view observations into a consistent global scene. Central to our design is the Gaussian-Image Representation (GIR), which projects 3D Gaussians into structured 2D maps for efficient fusion, identity-aware compression, and 2D-based supervision. By enabling lightweight per-frame updates and effective historical modeling, LongSplat mitigates memory overhead and quality degradation in dense-view settings. Extensive experiments show that it achieves real-time rendering, improves visual quality, and reduces Gaussian redundancy by over 44%, offering a scalable solution for high-quality online 3D reconstruction.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant No. 62172041, 62176021, and 62172279; the General Research Fund of the Research Grants Council of Hong Kong under Grant No. 17200725; the Shenzhen Science and Technology Program under Grant No. JCYJ20241202130548062; the Natural Science Foundation of Shenzhen under Grant No. JCYJ20230807142703006; the Key Research Platforms and Projects of the Guangdong Provincial Department of Education under Grant No. 2023ZDZX1034; and the JC STEM Lab funded by The Hong Kong Jockey Club Charities Trust.

References

Bao, Y.; Liao, J.; Huo, J.; and Gao, Y. 2024. Distractor-free Generalizable 3D Gaussian Splatting. *arXiv preprint arXiv:2411.17605*.

Barron, J. T.; Mildenhall, B.; Tancik, M.; Hedman, P.; Martin-Brualla, R.; and Srinivasan, P. P. 2021. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, 5855–5864.

Barron, J. T.; Mildenhall, B.; Verbin, D.; Srinivasan, P. P.; and Hedman, P. 2023. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 19697–19705.

Cao, A.; and Johnson, J. 2023. Hexplane: A fast representation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 130–141.

Charatan, D.; Li, S. L.; Tagliasacchi, A.; and Sitzmann, V. 2024. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 19457–19467.

Chen, A.; Xu, Z.; Zhao, F.; Zhang, X.; Xiang, F.; Yu, J.; and Su, H. 2021. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF international conference on computer vision*, 14124–14133.

Chen, Y.; Xu, H.; Zheng, C.; Zhuang, B.; Pollefeys, M.; Geiger, A.; Cham, T.-J.; and Cai, J. 2024a. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *European Conference on Computer Vision*, 370–386. Springer.

Chen, Y.; Zheng, C.; Xu, H.; Zhuang, B.; Vedaldi, A.; Cham, T.-J.; and Cai, J. 2024b. Mvsplat360: Feed-forward 360 scene synthesis from sparse views. *arXiv preprint arXiv:2411.04924*.

Chen, Z.; Wu, C.; Shen, Z.; Zhao, C.; Ye, W.; Feng, H.; Ding, E.; and Zhang, S.-H. 2025. Splatter-360: Generalizable 360 Gaussian Splatting for Wide-baseline Panoramic Images. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 21590–21599.

Fan, Z.; Wang, K.; Wen, K.; Zhu, Z.; Xu, D.; Wang, Z.; et al. 2024. Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps. *Advances in neural information processing systems*, 37: 140138–140158.

Fei, X.; Zheng, W.; Duan, Y.; Zhan, W.; Tomizuka, M.; Keutzer, K.; and Lu, J. 2024. Pixelgaussian: Generalizable 3d gaussian reconstruction from arbitrary views. *arXiv preprint arXiv:2410.18979*.

Fridovich-Keil, S.; Meanti, G.; Warburg, F. R.; Recht, B.; and Kanazawa, A. 2023. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12479–12488.

Gao, X.; Li, X.; Zhuang, Y.; Zhang, Q.; Hu, W.; Zhang, C.; Yao, Y.; Shan, Y.; and Quan, L. 2025. Mani-gs: Gaussian splatting manipulation with triangular mesh. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 21392–21402.

Gao, X.; Yang, J.; Kim, J.; Peng, S.; Liu, Z.; and Tong, X. 2022. Mps-nerf: Generalizable 3d human rendering from multiview images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Huang, B.; Yu, Z.; Chen, A.; Geiger, A.; and Gao, S. 2024. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 conference papers*, 1–11.

Kang, G.; Yoo, J.; Park, J.; Nam, S.; Im, H.; Shin, S.; Kim, S.; and Park, E. 2025. SelfSplat: Pose-free and 3D prior-free generalizable 3D Gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 22012–22022.

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

Li, Y.; Wang, J.; Chu, L.; Li, X.; Kao, S.-h.; Chen, Y.-C.; and Lu, Y. 2025. StreamGS: Online Generalizable Gaussian Splatting Reconstruction for Unposed Image Streams. *arXiv preprint arXiv:2503.06235*.

Lin, H.; Peng, S.; Xu, Z.; Yan, Y.; Shuai, Q.; Bao, H.; and Zhou, X. 2022. Efficient neural radiance fields for interactive free-viewpoint video. In *SIGGRAPH Asia 2022 Conference Papers*, 1–9.

Ling, L.; Sheng, Y.; Tu, Z.; Zhao, W.; Xin, C.; Wan, K.; Yu, L.; Guo, Q.; Yu, Z.; Lu, Y.; et al. 2024. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 22160–22169.

Liu, Y.; Peng, S.; Liu, L.; Wang, Q.; Wang, P.; Theobalt, C.; Zhou, X.; and Wang, W. 2022. Neural rays for occlusion-aware image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7824–7833.

Lu, T.; Yu, M.; Xu, L.; Xiangli, Y.; Wang, L.; Lin, D.; and Dai, B. 2024. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20654–20664.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. Nerf: Representing

scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106.

Roh, W.; Jung, H.; Kim, J. W.; Lee, S.; Yoo, I.; Lugmayr, A.; Chi, S.; Ramani, K.; and Kim, S. 2024. CATSplat: Context-Aware Transformer with Spatial Guidance for Generalizable 3D Gaussian Splatting from A Single-View Image. *arXiv preprint arXiv:2412.12906*.

Sheng, Y.; Deng, J.; Zhang, X.; Zhang, Y.; Hua, B.; Zhang, Y.; and Ji, J. 2025. SpatialSplat: Efficient Semantic 3D from Sparse Unposed Images. *arXiv preprint arXiv:2505.23044*.

Szymanowicz, S.; Rupprecht, C.; and Vedaldi, A. 2024. Splatter image: Ultra-fast single-view 3d reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10208–10217.

Wang, G.; Wu, X.; Jiang, S.; Liu, Z.; and Wang, H. 2022. Efficient 3d deep lidar odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5): 5749–5765.

Wang, Q.; Wang, Z.; Genova, K.; Srinivasan, P. P.; Zhou, H.; Barron, J. T.; Martin-Brualla, R.; Snavely, N.; and Funkhouser, T. 2021. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4690–4699.

Wang, R.; Ma, Y.; and Gao, S. 2025. Recollection from Pensieve: Novel View Synthesis via Learning from Uncalibrated Videos. *arXiv preprint arXiv:2505.13440*.

Wang, W.; Chen, D. Y.; Zhang, Z.; Shi, D.; Liu, A.; and Zhuang, B. 2025a. ZPressor: Bottleneck-Aware Compression for Scalable Feed-Forward 3DGS. *arXiv preprint arXiv:2505.23734*.

Wang, Y.; Huang, T.; Chen, H.; and Lee, G. H. 2024. Freesplat: Generalizable 3d gaussian splatting towards free view synthesis of indoor scenes. *Advances in Neural Information Processing Systems*, 37: 107326–107349.

Wang, Y.; Huang, T.; Chen, H.; and Lee, G. H. 2025b. FreeSplat++: Generalizable 3D Gaussian Splatting for Efficient Indoor Scene Reconstruction. *arXiv preprint arXiv:2503.22986*.

Xu, H.; Peng, S.; Wang, F.; Blum, H.; Barath, D.; Geiger, A.; and Pollefeys, M. 2024. Depthsplat: Connecting gaussian splatting and depth. *arXiv preprint arXiv:2410.13862*.

Yan, J.; Peng, R.; Wang, Z.; Tang, L.; Yang, J.; Liang, J.; Wu, J.; and Wang, R. 2025. Instant gaussian stream: Fast and generalizable streaming of dynamic scene reconstruction via gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 16520–16531.

Ye, J.; Wang, N.; and Wang, X. 2023. Featurenerf: Learning generalizable nerfs by distilling foundation models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8962–8973.

Yu, A.; Ye, V.; Tancik, M.; and Kanazawa, A. 2021. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4578–4587.

Yu, Z.; Chen, A.; Huang, B.; Sattler, T.; and Geiger, A. 2024. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 19447–19456.

Zheng, S.; Zhou, B.; Shao, R.; Liu, B.; Zhang, S.; Nie, L.; and Liu, Y. 2024. Gps-gaussian: Generalizable pixel-wise 3d gaussian splatting for real-time human novel view synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 19680–19690.

Ziwen, C.; Tan, H.; Zhang, K.; Bi, S.; Luan, F.; Hong, Y.; Fuxin, L.; and Xu, Z. 2024. Long-lrm: Long-sequence large reconstruction model for wide-coverage gaussian splats. *arXiv preprint arXiv:2410.12781*.