CGS: Continual Gaussian Splatting for Evolving 3D Scene Reconstruction

Shuojin Yang [1] Haoxiang Chen [1] and Taijiang Mu[1]

[1]BNRist, Department of Computer Science and Technology, Tsinghua University, China

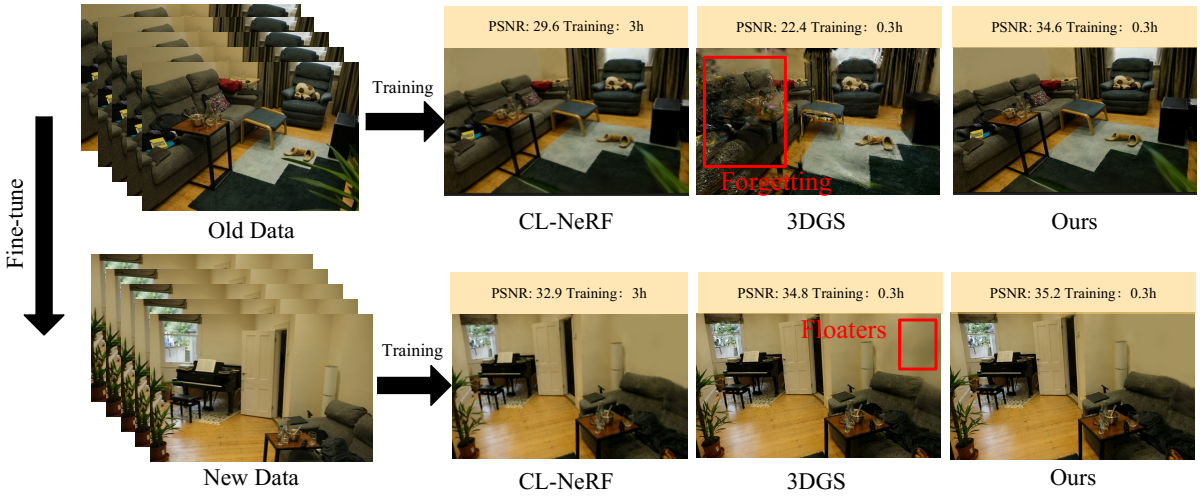

Figure 1: *Comparison with other methods when continuously reconstructing new scenes on top of old scenes in the synthetic dataset. Following the reconstruction of the new scene, the original Gaussian Splatting method fail to address catastrophic forgetting, resulting in damage to parts of old scene. Although recent methods based on NeRF counteract catastrophic forgetting by implicitly representing 3D scenes, they suffer from limitations in training time and reconstruction quality compared to our approach*

Abstract

3D Gaussian Splatting (3DGS) has gained significant attention for its fast optimization and high-quality rendering capabilities. However, in the context of continual scene reconstruction, optimizing newly observed regions often leads to degradation in previously reconstructed areas due to changes in camera viewpoints. To address this issue, we propose Continual Gaussian Splatting (CGS)—an efficient incremental reconstruction method that updates dynamic scenes using only a limited amount of new data while minimizing computational overhead. CGS is composed of three core components. First, we introduce a similarity-based registration algorithm that leverages the strong semantic understanding and translation invariance of pretrained Transformers to identify and align similar regions between new and existing scenes. These regions are then modeled as Gaussian Mixture Models (GMMs) to handle sparsity and outliers in point clouds, ensuring geometric consistency across scenes. Second, we propose Continual Gaussian Optimization (CGO), an importance-aware optimization strategy. By computing the Fisher Information Matrix, we evaluate the significance of each Gaussian point in the old scene and automatically restrict updates to those deemed critical, allowing only non-sensitive components to be adjusted. This ensures the preservation of the original scene while efficiently integrating new content. Finally, to address remaining issues such as geometric inconsistencies, blurring, and ghosting artifacts during optimization, we introduce a series of geometric regularization techniques. These terms guide the optimization toward geometrically coherent 3D structures, ultimately enhancing rendering quality. Extensive experiments demonstrate that CGS effectively mitigates forgetting and significantly improves overall reconstruction fidelity. (see https://www.acm.org/publications/class-2012)

CCS Concepts

• ***Computing methodologies*** → *Reconstruction;*

DOI: 10.2312/pg.20251284

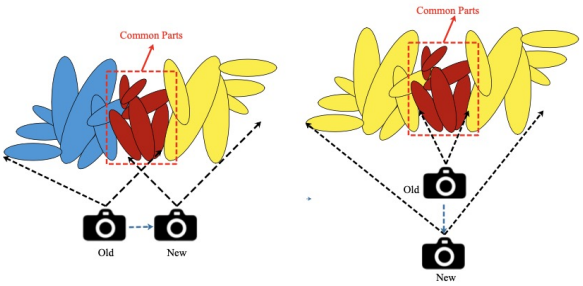

Figure 2: *Explanation of the causes of partial and complete catastrophic forgetting: Due to changes in camera poses, some areas within the common regions experience catastrophic forgetting. The left image is partial forgetting, while the right image is complete forgetting.*

1. Introduction

Gaussian splatting is a powerful model capable of reconstructing 3D scenes from a limited number of multi-view images [KKLD23]. Compared to Neural Radiance Fields (NeRF) [MST*21], it offers higher reconstruction quality and faster performance, nearly achieving real-time capabilities. These attributes make it particularly valuable for applications in augmented reality and autonomous driving. However, most existing methods are designed for static scenes, whereas in practical applications, there are often continuously evolving, such as in simultaneous localization and mapping (SLAM), where the scene expands, and objects may be added or removed. Therefore, investigating how to reconstruct evolving 3D scenes with minimal cost is meaningful.

However, when reconstructing new scenes, existing methods still face the challenge of catastrophic forgetting. Although recent NeRF-based approaches attempt to alleviate this issue through implicit 3D representations, they suffer from notable limitations in training efficiency and reconstruction quality. While Gaussian Splatting provides an explicit 3D representation, the overlapping regions between scenes are still prone to forgetting due to changes in camera poses during the acquisition of new and old scene images. This is because scene updates affect all Gaussians with non-zero opacity along the ray paths from the camera to the image plane in the original scene. Prior studies (e.g., [SMLM24a]) have attempted to mitigate forgetting via replay mechanisms, but these methods typically require storing keyframes from the previous scene, resulting in unnecessary storage overhead. Other methods [CXL*24] employ registration-based stitching to achieve continual reconstruction, but their pipelines are often complex and not well-suited for real-world applications. Fig. 2 provides a visual explanation of the causes behind partial and complete catastrophic forgetting during 3DGS-based reconstruction.

To enable efficient continual reconstruction, we propose a method called Continual Gaussian Splatting (CGS). This approach facilitates incremental reconstruction of evolving scenes based on a limited amount of new 2D data, while updating the existing scene at a low computational cost.

Due to discrepancies in camera poses between the old and new scenes—and the inability to achieve accurate initialization using only new scene data via Structure-from-Motion (SfM) [SF16]—we introduce a registration algorithm. This algorithm leverages a pretrained Transformer (e.g., Swin Transformer [LLC*21]) to extract features from both scenes. Benefiting from the Transformer's strong semantic understanding and translation invariance, the model identifies and matches similar regions across scenes. Gaussians within these regions are modeled as two Gaussian Mixture Models (GMMs), which effectively handle sparse and noisy point clouds, thereby improving robustness to irregular structures and local variations. The transformation matrix between scenes is then estimated to complete the registration.

To prevent catastrophic forgetting in overlapping areas, we propose an importance-aware optimization strategy. Specifically, we compute the Fisher Information Matrix to assess the significance of each Gaussian point in the old scene. Highly important points are tightly constrained to remain unchanged during the optimization of the new scene, while less critical points are allowed greater flexibility for adaptation. This ensures the integration of new content without compromising existing scene integrity.

To address remaining issues such as geometric inconsistency, image blur, and ghosting artifacts during optimization, we further introduce a set of geometric regularization techniques. These regularization terms guide the optimization process toward geometrically consistent 3D structures, thereby mitigating the impact of artifacts and enhancing the final rendering quality.

To validate the effectiveness of our method, we conducted extensive experiments across a series of datasets. The results demonstrate that our model consistently outperforms all compared methods in the task of reconstructing evolving scenes. It even achieves competitive results compared to the latest SLAM methods. Notably, our approach effectively mitigates forgetting, outperforming methods based on replay or attention mechanisms. In summary, our primary contributions are:

- We propose a Similarity-based Registration (SBR) module that aligns old and new scenes by leveraging Transformer-based semantic features and robust Gaussian Mixture Model (GMM) registration.
- We propose a Continual Gaussian Optimization (CGO) strategy that uses the Fisher Information Matrix to selectively constrain important Gaussians, thereby preventing catastrophic forgetting in overlapping regions.
- We propose a set of geometric regularization techniques that guide the optimization toward consistent 3D structures, reducing artifacts such as blur and ghosting.

2. Related Work

2.1. 3D Gaussian Splatting

Recent advancements in 3D Gaussian methods have shown significant promise in various applications, such as Fuzzy Metaballs [KH22, KH23], VoGE [WWS*22], and 3DGS [KKLD23]. 3DGS stands out for its exceptional performance in real-time novel-view synthesis of high quality. This technique employs 3D Gaussians to depict scenes and incorporates a fast NeRF-

inspired rendering algorithm to enable anisotropic splatting, resulting in cutting-edge visual fidelity and swift high-resolution rendering. Beyond its rendering capabilities, Gaussian splatting offers a clear geometric scene structure and visual representation, which aids in accurate scene modeling [YYP*23]. This promising technology has quickly been applied in various fields, including 3D generation [CWL23, TRZ*23, YFW*23], dynamic scene modeling [LKLR23, WYF*23, YGZ*23], and photorealistic drivable avatars [ZBS*23]. Based on the excellent performance of Gaussian splatting, many researchers have applied it to SLAM and proposed numerous improvement methods [SLOD21, ZPL*22, RLC23, JGW*24, SMLM24b, YHS*24]. However, SLAM is only a specific application of continual reconstruction. Currently, there is no more generalized continual learning method based on Gaussian Splatting.

2.2. Continual Learning

Continual learning aims to learn from a sequence of data with distributional shifts without storing historical data [DLAM*21]. Simple training on non-i.i.d. sequential data suffers from catastrophic forgetting [LWTX24] and performs poorly on historical data. A popular approach is to regularize the training objective to prevent forgetting [LH17]. However, since regularization does not rely on historical data, its practical effectiveness is limited. Parameter isolation methods prevent forgetting by freezing neurons used for previous tasks and using new neurons for subsequent tasks [ML18, SSMK18]. Although these methods guarantee memory retention, they have limited capacity or significantly increase network size when faced with many tasks. Replay-based methods use historical data to prevent forgetting, either storing this data in small replay buffers [CRE*19] or synthesizing it using generative models. Generative replay [SLKK17], which synthesizes historical data, performs poorly for image classification because generative models introduce additional parameters and struggle with high-resolution images. Some researchers have proposed continual learning models based on task similarity [YC23], which avoid catastrophic forgetting and learn from old tasks to improve performance on new tasks.

2.3. 3D Reconstruction of Evolving Scenes

Several methods combining continual learning and Neural Radiance Fields have been proposed, which introduce trainable appearance embeddings [MBRS*21, TCY*22, WDD*24] to ensure efficient model updates during the continual learning process, thereby enabling a single model to handle complex scene changes. Chung et al. [CLBL22] also explored NeRF in the context of continual learning; however, their work still focused on static scenes and the original NeRF architecture, lacking the ability to handle scenes with varying appearances or geometries. The field of Simultaneous Localization and Mapping (SLAM) involves reconstructing 3D scenes from continuous image streams, which is similar to the setting of continual learning. Recent studies [SLOD21, ZPL*22, RLC23] have successfully combined neural implicit representations with traditional SLAM-based methods, achieving notable results. These approaches can be classified as memory replay, as they manage continual learning tasks by explicitly storing keyframes from previous image streams. However, explicitly storing data is costly, and designing effective heuristics for keyframe selection remains challenging. Recently, a method using 3D Gaussian models and regularization techniques to prevent forgetting was proposed [SMLM24a], but it is specifically designed for SLAM scenarios and may not be applicable in more general settings. The latest methods [XYG*25, MSL*25], although capable of achieving fast continual reconstruction, still fall short in reconstruction quality.

3. Methods

Our goal is to incrementally update Gaussian point clouds from a few new 2D images with minimal cost while preserving the existing scene. We define the continual 3D reconstruction task and present our framework (Fig. 2). First, new scene data is processed by a pretrained transformer to align features with the old scene and detect overlapping regions, where Gaussians are modeled as GMMs for robust registration against outliers and sparsity. Second, to mitigate catastrophic forgetting, only less critical Gaussians in overlapping areas are updated, guided by the Fisher information matrix. Third, to handle brightness inconsistency and floaters, we introduce geometric regularization model (GRM) to adjust exposure, ensuring cross-scene consistency and high-quality reconstruction.

3.1. Problem Formulation of Continual 3d Reconstruction

Given the temporal evolution of real-world environments, we formulate the task of *continual 3D reconstruction* as follows: Initially, we train a Gaussian-based scene model using m multi-view images captured from the original scene. Then, at each subsequent time step, we acquire a new set of m images that reflect changes in the scene—such as structural additions or modifications to existing content. Based solely on these new images, we aim to reconstruct the updated scene. Importantly, the process is designed to minimize computational overhead and memory usage while avoiding catastrophic forgetting of previously learned content.

3.2. Similarity-Based Registration

Since the new scene is typically reconstructed using Structure-from-Motion (SfM), its resulting camera poses may reside in a different coordinate system from that of the original scene, and the reconstruction is often sparser compared to the optimized scene. To achieve spatial alignment and content fusion between dense and sparse representations, we first propose a similarity-based registration method. This method identifies overlapping regions between the two scenes in both geometric and semantic spaces, and leverages the corresponding point clouds to estimate a rigid transformation matrix through probabilistic modeling, thereby effectively aligning the scenes.

Similarity Region Localization. To accurately identify overlapping regions between the old and new scenes, we propose a hybrid similarity region localization strategy that integrates both geometric and semantic cues. Specifically, we uniformly sample two subsets of images from the original and updated scenes, denoted as $\{Ca_i\}_{i=1}^{m} \subset \{I^A\}$ and $\{Cb_j\}_{j=1}^{n} \subset \{I^B\}$, with $m = n = 20$ in our

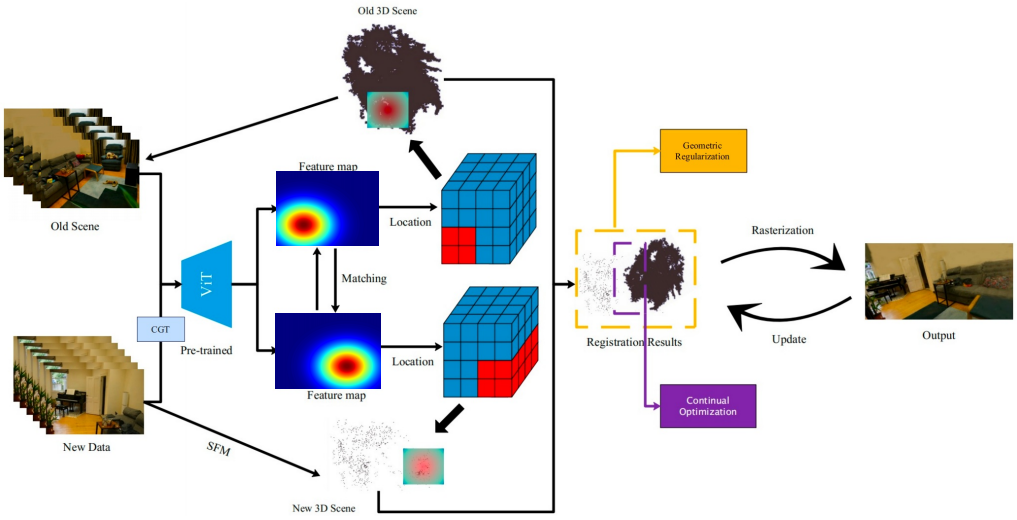

Figure 3: *Explanation of the causes of partial and complete catastrophic forgetting: Due to changes in camera poses, some areas within the common regions experience catastrophic forgetting. The left image is partial forgetting, while the right image is complete forgetting.*

experiments. A coarse global transformation (s_c, R_c, T_c), obtained via SfM, is applied to the new images to produce approximately aligned views $\{\hat{Cb}_j\}$.

Each image in $\{Ca_i\}$ and $\{\hat{Cb}_j\}$ is then passed through a pretrained Vision Transformer [RJL*24] to extract high-level, translation-invariant semantic feature maps. These maps are reshaped into sequences of spatial tokens, and the average pairwise cosine similarity between tokens is used to quantify the semantic similarity of image pairs. In parallel, we compute the cosine similarity between camera orientations to assess geometric alignment.

For each candidate image pair $(Ca_i, \hat{Cb}_j)$, we define a joint similarity score that combines semantic and geometric similarity:

$$\text{Score}_{i,j} = \lambda_1 \cdot \cos(\theta_{i,j}) + \lambda_2 \cdot \text{Sim}_{i,j}^{\text{sem}}, \tag{1}$$

where λ_1 and λ_2 control the relative weighting of geometric and semantic components, and are set to $\lambda_1 = 0.4$ and $\lambda_2 = 0.6$ by default. The top-K image pairs (typically $K = 10$) with the highest similarity scores are selected as the most probable overlapping views for subsequent point cloud registration.

Point Cloud Registration. Based on the selected image pairs, we reconstruct sparse local point clouds that serve as anchors linking the original and updated scenes. Specifically, we use [SP25] to convert the Gaussian representation of the old scene into a point cloud. Both point clouds from the two scenes are then modeled as Gaussian Mixture Models (GMMs), which enhances robustness against noise, sparsity, and structural irregularities. By constructing an optimal transport mapping between the two GMMs and applying a weighted Singular Value Decomposition (SVD), we estimate the rigid transformation that best aligns the new point cloud to the old coordinate frame. This registration result is subsequently used to transform all images and 3D content of the new scene into the global coordinate system, thereby obtaining consistent camera poses and ensuring stability for subsequent reconstruction stages. Compared with SLAM-based methods, our approach integrates both semantic and geometric features, making it more robust to noise, sparse point clouds, and partial overlaps.

The GMM establishes a multimodal generative probabilistic distribution in 3D space, represented as a weighted sum of L Gaussian densities, formulated as Eq. (2):

$$p(x) = \sum_{j=1}^{L} \pi_j N(x|\mu_j, \Sigma_j), \quad x \in \mathbb{R}^3. \tag{2}$$

First, we use the iterative Expectation-Maximization (EM) algorithm to estimate the parameters of the Gaussian Mixture Model (GMM), including the mixture weights p_{ij}, means μ_j, and covariance matrices Σ_j. The EM algorithm consists of two steps: the Expectation step (E-step) and the Maximization step (M-step). Before starting the EM algorithm, the parameters must be initialized. This can be done by randomly selecting points from the point cloud as initial means, and setting the initial covariance matrices to identity matrices.

During the E-step, we calculate the posterior probabilities for each data point belonging to each Gaussian component. The posterior probability matrix γ_{ij} represents the probability that the ith data point belongs to the jth Gaussian component.

$$\gamma_{ij} = \frac{\pi_j \mathcal{N}(x_i|\mu_j, \Sigma_j)}{\sum_{k=1}^{L} \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)} \tag{3}$$

In the Maximization step (M-step), we use the posterior probabilities calculated in the Expectation step:

$$\pi_j = \frac{1}{N}\sum_{i=1}^{N}\gamma_{ij} \tag{4}$$

update the mean μ_j

$$\mu_j = \frac{\sum_{i=1}^{N}\gamma_{ij}x_i}{\sum_{i=1}^{N}\gamma_{ij}} \tag{5}$$

update the covariance matrix Σ_j:

$$\Sigma_j = \frac{\sum_{i=1}^{N}\gamma_{ij}(x_i-\mu_j)(x_i-\mu_j)^2}{\sum_{i=1}^{N}\gamma_{ij}} \tag{6}$$

The Expectation step (E-step) and Maximization step (M-step) are iteratively repeated until the parameters converge, that is, the change in parameters is less than a predetermined threshold.

Given the GMM parameters estimated through the aforementioned formula, along with feature centroids ν_{p_j} for j=1 to L and ν_{q_j} for j=1 to L), we first calculate the cluster-level matching matrix by solving the following Optimal Transport (OT) problem [PC*19] as Eq. (7):

$$\begin{aligned}&\min_{\Gamma}\sum_{i=1}^{L}\sum_{j=1}^{L}\Gamma_{ij}\|\nu_i^p-\nu_j^q\|_2^2,\\ &\text{s.t.}\Gamma 1_M=\pi^p,\Gamma^T 1_N=\pi^q,\Gamma_{ij}\in[0,1]\end{aligned} \tag{7}$$

where $\pi_t = (\pi_{t_1}, \pi_{t_2}, \ldots, \pi_{t_L})$ and $t \in \{p, q\}$. The minimization of this problem can be solved in polynomial time via linear programming. We utilize an efficient version of the Sinkhorn-Knopp algorithm [Cut13] to solve this problem.

After obtaining the matrix Γ, we compute the transformation using the following formula:

$$\min_{T} L\sum_{i=1}^{L}\sum_{j=1}^{L}\Gamma_{ij}\left\|T(\mu_i^p)-\mu_j^q\right\|. \tag{8}$$

Ultimately, we can employ a weighted version of the Singular Value Decomposition (SVD) approach to solve the transformation matrix T in closed form, and we can adjust the camera poses and point cloud positions of the new scene, completing the incremental reconstruction process.

3.3. Continual Gaussian Optimization

Our goal is to ensure that when scenes continuously grow or change, the reconstruction can be completed using only the data from the current scene without damaging the old scene, and with minimal cost. Due to the same Gaussian potentially being 'seen' from multiple camera poses, Gaussian parameters are often over-optimized to suit the most recent frames, leading to degraded performance in older scenes. To overcome these challenges, we introduce regularization terms into the optimization process to suppress updates to specific Gaussian parameters. This involves selectively updating the Gaussian kernels in the similarity regions based on their importance to the scene. Specifically, if a Gaussian is crucial to the old scene, it receives a stronger constraint to prevent alterations by the new scene, and vice versa, thereby achieving the goal of preventing forgetting. Our approach starts by unlocking the gradient features of Gaussians in the old scene's similarity regions, then updates are made based on the gradient's importance, measured using the Fisher information matrix.

Formally, the overall loss function is defined as:

$$L(\theta) = L_{SSIM}(\theta) + \sum_i \frac{\lambda}{2}F_i(\theta_i-\theta_i^*)^2, \tag{9}$$

where the term $L_{SSIM}(\theta)$ represents the final loss function for Gaussian volumetric rendering, θ are the Gaussian parameters to be optimized for the new scene, θ_i are the Gaussian parameters from the old scene, and θ_i^* represent the optimal values of these parameters.

The Fisher information F_i for each parameter is approximated with the empirical Fisher on mini-batches of rendered views and pixels:

$$F_i = \mathbb{E}\left[\left(\frac{\partial \log p(\mathcal{D}|\theta)}{\partial\theta_i}\right)^2\right] \approx \frac{1}{N}\sum_{n=1}^{N}\left(\frac{\partial \ell(x_n;\theta)}{\partial\theta_i}\right)^2, \tag{10}$$

where $\ell(x_n;\theta)$ denotes the per-pixel residual loss, which can be defined as an ℓ_2 photometric error or SSIM-based error. Back-propagating through the differentiable Gaussian Splatting renderer allows us to obtain gradients with respect to each Gaussian parameter. These gradients are accumulated across pixels sampled from conditioning views to form the Fisher estimate.

To maintain efficiency, only diagonal or block-diagonal Fisher entries are stored, grouping parameters of each Gaussian (position, scale, opacity, SH coefficients). Fisher accumulation is restricted to Gaussians in similarity regions; non-intersecting pixels are excluded. Values are updated with exponential moving averaging, stabilized with damping ε, clipped at a high percentile, and weighted by a cosine-decay λ.

Gradients are computed at the old snapshot θ^* using held-out views, so F_i measures old-parameter importance. In practice, tens of thousands of pixels are sampled per update, Fisher statistics are accumulated periodically, and only diagonals are kept, limiting overhead to <10% of training. This yields a low-variance estimator that penalizes updates to critical Gaussians, suppressing forgetting while enabling flexible adaptation.

3.4. Geometric Regularization Methods

Although the proposed optimization methods can reconstruct high-quality, geometrically consistent images to some extent, in practice, relying solely on image reconstruction loss often leads the model to overfit imperfect supervisory signals. As a result, issues such as geometric inconsistencies, blurriness, and ghosting artifacts may still persist during optimization. Therefore, we further propose a series of geometric regularization methods. These regularization terms guide the optimization toward geometrically consistent 3D structures, mitigating the effects of artifacts and blurriness, and ultimately improving rendering quality.

Normal Alignment Loss. To ensure geometric consistency between the rendered appearance and the underlying 3D structure, we encourage alignment between the normals obtained from the differentiable renderer and the analytical surface normals derived from geometry:

$$\mathcal{L}_{\text{normal}} = \lambda_{\text{normal}} \cdot \sum_{\mathbf{x}} (1 - \langle \mathbf{n}_{\text{rend}}(\mathbf{x}), \mathbf{n}_{\text{surf}}(\mathbf{x}) \rangle) \tag{11}$$

where $\mathbf{x}$ denotes a pixel in the image, $\mathbf{n}_{\text{rend}}(\mathbf{x})$ is the rendered normal at $\mathbf{x}$, $\mathbf{n}_{\text{surf}}(\mathbf{x})$ is the surface normal estimated from the depth geometry, and λ_{normal} is the weighing factor (e.g., 0.15).

Depth Compactness Loss. To encourage localized and confident surface representation, we introduce a depth compactness loss that penalizes overly large splatting regions in image space. During training, some 3D Gaussians may grow excessively in size to compensate for uncertainty or sparsity, leading to oversmoothing, depth leakage, or ghosting effects in rendered views. To mitigate this, we constrain the projected 2D radii of Gaussians, encouraging them to remain spatially compact:

$$\mathcal{L}_{\text{dist}} = \lambda_{\text{dist}} \cdot \sum_{\mathbf{g} \in \mathcal{G}} \rho(\mathbf{g}) \tag{12}$$

where $\mathcal{G}$ is the set of all Gaussians in the scene, $\rho(\mathbf{g})$ denotes the projected radius of Gaussian $\mathbf{g}$ in image space, and λ_{dist} is the weighing factor (e.g., 0.08).

Edge-Aware Normal Loss. To better preserve geometric discontinuities and enhance surface sharpness near object boundaries, we propose an edge-aware normal loss that leverages image gradients to guide local normal field regularization. Unlike global smoothness constraints, this loss specifically operates on pixels with strong image gradients, where surface normal changes are expected and even desirable. Given a pixel p in the rendered image that lies on a strong intensity edge (i.e., $\nabla I(p) > \lambda_e$), we define a four-neighbor set consisting of its immediate adjacent pixels: up, down, left, and right. From these neighbors, we retrieve the corresponding rendered surface normals $\mathbf{N}_1$ (up), $\mathbf{N}_2$ (left), $\mathbf{N}_3$ (down), and $\mathbf{N}_4$ (right). To encourage angular diversity at the edge, we compute the average cosine similarity between opposing normal pairs (i.e., top–bottom and left–right), and minimize it:

$$\mathcal{L}_{\text{edge}} = \sum_{p \in w} \frac{1}{2} \left(\mathbf{N}_1^\top \mathbf{N}_3 + \mathbf{N}_2^\top \mathbf{N}_4 \right) \tag{13}$$

Here, $w = \{p \in \mathcal{I} \mid \|\nabla I(p)\| > \lambda_e\}$ denotes the set of edge pixels selected by an image gradient threshold λ_e, with $\nabla I(p)$ denoting the normalized gradient magnitude (rescaled to $[0,1]$) of the rendered image at pixel p.

Normal Smoothness Loss. In addition to edge-aware strategies, we also apply a general smoothness constraint on the rendered surface normals to regularize noisy or unstable geometry, particularly in low-texture regions. While the edge-aware loss focuses on preserving sharp transitions at image boundaries, this smoothness term complements it by promoting gradual changes in surface orientation elsewhere. We formulate the normal smoothness loss as the spatial gradient magnitude of the rendered normal field across the entire image domain:

$$\mathcal{L}_{\text{smooth}} = \lambda_{\text{smooth}} \cdot \sum_{\mathbf{x} \in \mathcal{I}} \|\nabla \mathbf{n}_{\text{rend}}(\mathbf{x})\| \tag{14}$$

where $\mathbf{n}_{\text{rend}}(\mathbf{x})$ is the rendered normal at pixel $\mathbf{x}$, ∇ denotes the spatial (finite-difference) gradient operator, and λ_{smooth} controls the strength of this regularization (e.g., 0.05).

Progressive Densification and SH Degree Growth. To progressively enhance scene fidelity and surface completeness, we adopt a dynamic densification strategy during the early optimization phase. At each iteration, the system monitors the per-Gaussian 2D radius (projected splat size) and gradient-based importance. Gaussians that are highly visible, contribute to image gradients, or cover insufficient areas are duplicated to densify underrepresented regions. Conversely, small or redundant Gaussians are pruned to reduce overfitting and maintain efficiency. Simultaneously, we increase the complexity of the radiance representation via Spherical Harmonics (SH). Starting from a low-degree SH basis, we progressively grow the degree as training advances to better capture view-dependent appearance:

$$l_{\text{SH}}^{(t)} = \min \left(l_{\max}, \left\lfloor \frac{t}{1000} \right\rfloor \right) \tag{15}$$

where t is the current training iteration, and $l_{\max}$ is the maximum SH degree. This strategy enables a smooth transition from coarse color modeling to fine angular detail, avoiding early overfitting to noisy supervision.

Opacity Reset and Adaptive Regularization Schedules. To prevent degenerate solutions where opacity collapses to extreme values (either fully transparent or oversaturated), we periodically reset the opacity of all Gaussians to a fixed initial value:

$$\alpha_g \leftarrow \alpha_0, \quad \forall g \in \mathcal{G} \quad \text{if } t \bmod T_{\text{reset}} = 0 \tag{16}$$

where $\mathcal{G}$ is the set of all Gaussians, α_0 is a predefined reset opacity value (e.g., 0.1), and T_{reset} is the reset interval (e.g., 3000). This helps maintain diversity in volumetric contribution during training and facilitates reconfiguration of Gaussians in early and mid stages.

Additionally, all regularization terms are activated gradually using linear ramp-up schedules. This allows the network to focus on coarse appearance matching in early iterations, and shift to finer geometric regularization later:

$$\eta(t) = \eta_{\max} \cdot \texttt{weight\_schedule}(t; t_{\text{start}}, t_{\text{end}}) \tag{17}$$

where $\eta(t)$ is the current weight of a specific loss term, and `weight- schedule` defines a linear interpolation from 0 to $\eta_{\max}$ over the interval start (e.g., 1000), t_{end} (e.g., 6000). This schedule prevents early domination of regularization and improves convergence stability.

After several thousand iterations, the optimized Gaussians represent a high-fidelity 3D scene with sharp details, smooth surfaces, and faithful lighting effects.

4. Experiment

4.1. Evaluation Metrics

To evaluate the reconstruction performance and the quality of color image rendering, we utilize several metrics: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and

Learned Perceptual Image Patch Similarity (LPIPS). PSNR measures the pixel-level RGB discrepancies, SSIM gauges the structural likeness, and LPIPS evaluates the perceptual similarity of image patches using neural network-based methods. For geometric reconstruction evaluation, we are also guided by principles from continual learning systems and apply backward transfer PSNR to gauge the level of model forgetting. This measure calculates the mean decrease in PSNR each time a new task is introduced, from the initial to the present task, as defined in Eq. (18).

$$\mathrm{BTP} = -\frac{1}{T-1}\sum_{i=1}^{T-1}\left(\mathrm{PSNR}_{T,i} - \mathrm{PSNR}_{i,i}\right) \tag{18}$$

where T denotes the current scene.

4.2. Dataset and Details

We conducted experiments on three datasets: the Replica dataset [SWM*19], the synthetic dataset [RLC23], and the TUM-RGBD dataset [SEE*12]. These datasets were divided into multiple scenes to evaluate the model's capability for continual reconstruction. The image resolution was kept consistent with the original data resolution, and we used the original 3DGS implementation with 30,000 training iterations. The experimental results demonstrate that our method consistently outperforms all compared methods in the task of reconstructing dynamically evolving scenes, and even achieves competitive results compared to the latest SLAM-based and on-the-fly reconstruction methods.

4.3. Main Results

To demonstrate the performance of our method, we conducted extensive experiments on the Replica dataset. The experimental results, as shown in Table 1, indicate that our method achieves state-of-the-art performance. Compared to NeRF-based methods, the Gaussian-based approach inherently excels in various metrics due to its inherent advantages. Point-SLAM, which applies NeRF to SLAM, benefits from using the same point cloud as primitives, with each point accompanied by individual color feature descriptors and geometric feature descriptors. This combination of explicit and implicit features leads to significantly better performance metrics compared to NeRF-based methods. GS-SLAM, which directly applies 3D Gaussian to SLAM tasks, lacks measures to prevent scene forgetting, resulting in inferior performance in most scenarios compared to our method.

To validate the anti-forgetting capability of our model, we divided the apartment scene into four parts to meet the continual learning task requirements. We used BTP and average PSNR as metrics to evaluate the capabilities of each model, as shown in Table 2. Our method outperforms baseline methods, exhibiting the highest performance in preventing forgetting. Although the best NeRF-based method, CL-NeRF, achieves comparable BTP due to its well-designed anti-forgetting strategies, its PSNR is significantly lower than our method due to the inherent limitations of NeRF. On-the-Fly NVS [MSL*25] benefits from optimization strategies specifically designed for consumer-grade devices, resulting in very short reconstruction times; however, its reconstruction quality still lags behind the proposed method.

Fig. 4 and Fig.5 shows the compared results in the Replica and TUM dataset. Due to the inherent shortcomings of NeRF-based methods, they can avoid forgetting to some extent but often suffer from poor reconstruction quality, resulting in either blurry or noisy images. [MMKD24]and [MSL*25], since they are not optimized for 3DGS, produce many artifacts. In contrast, our method leverages GRM to ensure higher rendering quality. Moreover, thanks to the anti-forgetting techniques we proposed, our method protects old scenes from degradation when reconstructing new ones, outperforming basic 3D Gaussian methods. Fig. 6 shows the visualization results of continuously reconstructing three new scenes.

4.4. Ablation Study

Our model comprises three main components: similarity-based registration, continual optimization, and geometric regularization. We conducted an ablation study on these components, with the results presented in Table 4. The result indicate that our method performs almost identically on real datasets as it does on synthetic datasets and all three components contribute to improvement. When similarity-based registration is omitted, the model's performance metrics drop most significantly. This is because, without proper alignment, the Gaussian kernels cannot correctly optimize the scene toward the accurate camera poses using only new scene data, severely compromising the reconstruction of the old scene. Thus, registration is crucial.

Without continual optimization, some common parts of the new and old scenes experience catastrophic forgetting, adversely affecting the reconstruction of the old scene. While the absence of geometric regularization does not significantly impact the model's anti-forgetting performance, it noticeably enhances visual quality by effectively eliminating artifacts. Therefore, it is evident that these three designs work synergistically, enabling our method to achieve exceptional results.

5. Conclusions

We propose Continual Gaussian Splatting (CGS), an efficient framework for incremental 3D scene reconstruction in evolving environments. CGS effectively mitigates catastrophic forgetting by combining similarity-based registration, importance-aware optimization using the Fisher Information Matrix, and geometric regularization to ensure consistency and rendering quality. Experimental results demonstrate that CGS achieves high-fidelity reconstruction with limited new data and low computational cost.

In future work, we plan to extend CGS to handle real-time streaming data, enable automatic detection of scene changes, and explore integration with SLAM systems for broader applications in robotics and AR.

References

[CLBL22] CHUNG J., LEE K., BAIK S., LEE K. M.: Meil-nerf: Memory-efficient incremental learning of neural radiance fields. *arXiv preprint arXiv:2212.08328* (2022). 3

[CRE*19] CHAUDHRY A., ROHRBACH M., ELHOSEINY M., AJANTHAN T., DOKANIA P. K., TORR P. H., RANZATO M.: On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486* (2019). 3

Table 1: *The performance on the Replica dataset compared with other methods.*

Method	Metric	Room0	Room1	Room2	Office0	Office1	Office2	Office3	Office4
ICL-NeRF [PDBW23]	PSNR	29.36	30.15	30.41	33.14	34.12	29.94	28.87	29.74
	SSIM	0.94	0.94	0.94	0.95	0.96	0.94	0.93	0.94
	LPIPS	0.18	0.17	0.17	0.16	0.15	0.17	0.18	0.18
CL-NeRF [WDD*24]	PSNR	30.21	31.78	32.24	34.89	36.29	30.19	29.86	30.39
	SSIM	0.95	0.95	0.95	0.96	0.97	0.97	0.95	0.96
	LPIPS	0.15	0.14	0.14	0.15	0.14	0.14	0.16	0.16
Point-SLAM [SLVGO23]	PSNR	32.40	34.08	35.50	38.26	39.16	33.99	33.48	33.49
	SSIM	0.97	0.98	0.98	0.98	0.99	0.96	0.96	0.98
	LPIPS	0.11	0.12	0.11	0.1	0.12	0.16	0.13	0.14
GS-SLAM [YQW*23]	PSNR	31.56	32.86	32.59	38.70	**41.17**	32.36	32.03	32.92
	SSIM	0.96	0.97	0.97	0.98	0.99	0.97	0.97	0.96
	LPIPS	0.09	0.07	0.09	0.05	0.03	0.09	0.11	0.11
SplaTAM [KKJ*24]	PSNR	32.86	33.89	35.25	38.26	39.17	31.97	29.70	31.81
	SSIM	0.98	0.97	0.98	0.98	0.98	0.97	0.95	0.95
	LPIPS	0.07	0.10	0.08	0.09	0.09	0.10	0.12	0.15
MonoGS [MMKD24]	PSNR	28.94	26.12	31.82	32.73	34.47	27.01	30.76	27.29
	SSIM	0.88	0.80	0.92	0.92	0.93	0.88	0.91	0.90
	LPIPS	0.18	0.32	0.16	0.21	0.19	0.26	0.16	0.25
On-the-Fly GS [XYG*25]	PSNR	31.79	33.67	34.65	39.44	38.77	**34.03**	**35.38**	**36.65**
	SSIM	0.94	0.95	0.96	0.97	0.96	0.95	0.96	0.96
	LPIPS	0.12	0.11	0.11	0.09	0.14	0.13	0.09	0.10
VBGS [VdMCT*24]	PSNR	26.14	27.66	27.75	33.65	36.24	26.33	26.13	26.95
	SSIM	0.98	0.98	0.98	0.98	0.99	0.97	0.97	0.97
Ours	PSNR	**33.12**	**35.62**	**36.65**	**39.94**	40.58	33.42	33.51	34.55
	SSIM	**0.99**	**0.98**	**0.99**	**0.99**	**0.99**	**0.98**	**0.98**	**0.98**
	LPIPS	**0.08**	**0.06**	**0.05**	**0.04**	**0.03**	**0.08**	**0.10**	**0.07**

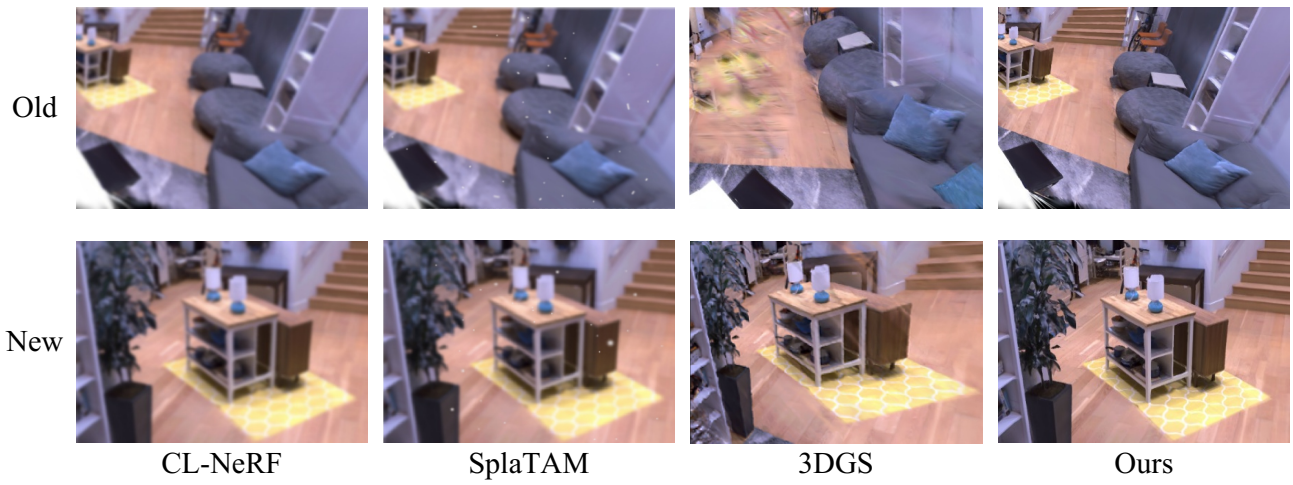

Figure 4: *Visualization results compared our method with others when learning new scenes in Replica dataset.*

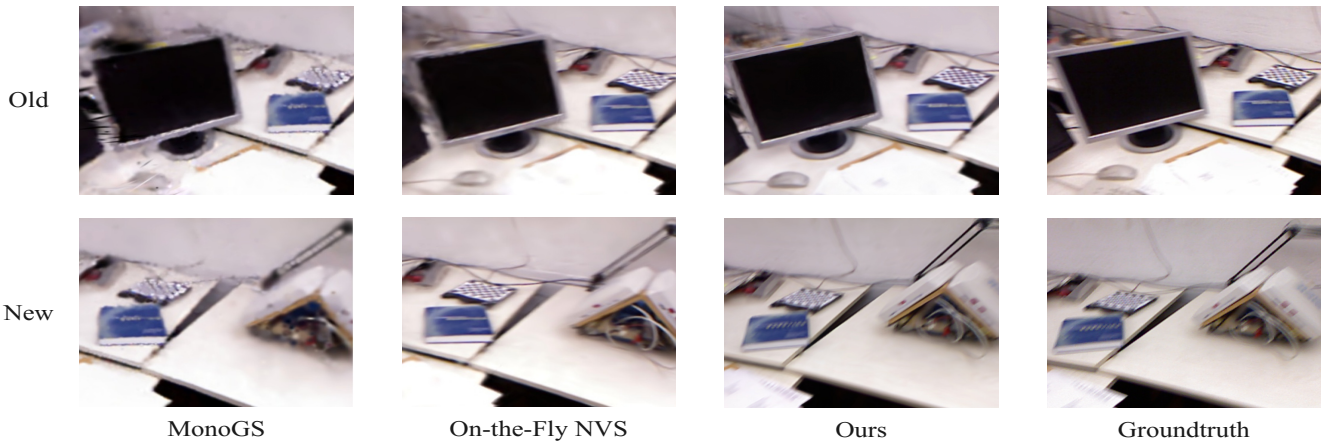

Figure 5: *Visualization results compared our method with others when learning new scenes in TUM dataset.*

Table 2: *Forgetting metrics in continual 3D reconstruction*

Models	PSNR	BTP
CL-NeRF [WDD*24]	31.25	1.89
Point-SLAM [SLVGO23]	33.69	3.35
3DGS [KKLD23]	31.92	6.43
MonoGS [MMKD24]	34.68	1.93
On-the-Fly NVS [MSL*25]	23.02	1.67
Ours	**35.24**	**1.54**

Table 3: *Results of ablation study in TUM [SEE* 12] dataset*

Operations	PSNR	BTP
w/o SBR	20.15	11.31
w/o CGO	30.18	6.24
w/o GRM	33.38	1.64
Ours	**35.24**	**1.54**

[Cut13] CUTURI M.: Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems 26* (2013). 5

[CWL23] CHEN Z., WANG F., LIU H.: Text-to-3d using gaussian splatting. *arXiv preprint arXiv:2309.16585* (2023). 3

[CXL*24] CHANG J., XU Y., LI Y., CHEN Y., FENG W., HAN X.: Gaussreg: Fast 3d registration with gaussian splatting. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XV* (2024), Leonardis A., Ricci E., Roth S., Russakovsky O., Sattler T., Varol G., (Eds.), vol. 15073 of *Lecture Notes in Computer Science*, Springer, pp. 407–423. URL: https://doi.org/10.1007/978-3-031-72633-0_23, doi:10.1007/978-3-031-72633-0_23. 2

[DLAM*21] DE LANGE M., ALJUNDI R., MASANA M., PARISOT S., JIA X., LEONARDIS A., SLABAUGH G., TUYTELAARS T.: A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence 44*, 7 (2021), 3366–3385. 3

[JGW*24] JIN R., GAO Y., WANG Y., WU Y., LU H., XU C., GAO F.: Gs-planner: A gaussian-splatting-based planning framework for active high-fidelity reconstruction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2024, Abu Dhabi, United Arab Emirates, October 14-18, 2024* (2024), IEEE, pp. 11202–11209. URL: https://doi.org/10.1109/IROS58592.2024.10801715, doi:10.1109/IROS58592.2024.10801715. 3

[KH22] KESELMAN L., HEBERT M.: Approximate differentiable rendering with algebraic surfaces. In *European Conference on Computer Vision* (2022), Springer, pp. 596–614. 2

[KH23] KESELMAN L., HEBERT M.: Flexible techniques for differentiable rendering with 3d gaussians. *arXiv preprint arXiv:2308.14737* (2023). 2

[KKJ*24] KEETHA N., KARHADE J., JATAVALLABHULA K. M., YANG G., SCHERER S., RAMANAN D., LUITEN J.: Splatam: Splat track & map 3d gaussians for dense rgb-d slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), pp. 21357–21366. 8

[KKLD23] KERBL B., KOPANAS G., LEIMKÜHLER T., DRETTAKIS G.: 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics 42*, 4 (2023), 1–14. 2, 9

[LH17] LI Z., HOIEM D.: Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence 40*, 12 (2017), 2935–2947. 3

[LKLR23] LUITEN J., KOPANAS G., LEIBE B., RAMANAN D.: Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. *arXiv preprint arXiv:2308.09713* (2023). 3

[LLC*21] LIU Z., LIN Y., CAO Y., HU H., WEI Y., ZHANG Z., LIN S., GUO B.: Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision* (2021), pp. 10012–10022. 2

[LWTX24] LIU X., WANG H., TIAN Y., XIE L.: Overcoming catastrophic forgetting with classifier expander. In *Asian Conference on Machine Learning* (2024), PMLR, pp. 803–817. 3

[MBRS*21] MARTIN-BRUALLA R., RADWAN N., SAJJADI M. S., BARRON J. T., DOSOVITSKIY A., DUCKWORTH D.: Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 7210–7219. 3

[ML18] MALLYA A., LAZEBNIK S.: Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (2018), pp. 7765–7773. 3

[MMKD24] MATSUKI H., MURAI R., KELLY P. H., DAVISON A. J.:

Gaussian splatting slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), pp. 18039–18048. 7, 8, 9

[MSL*25] MEULEMAN A., SHAH I., LANVIN A., KERBL B., DRETTAKIS G.: On-the-fly reconstruction for large-scale novel view synthesis from unposed images. *ACM Transactions on Graphics (TOG) 44*, 4 (2025), 1–14. 3, 7, 9

[MST*21] MILDENHALL B., SRINIVASAN P. P., TANCIK M., BARRON J. T., RAMAMOORTHI R., NG R.: Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM 65*, 1 (2021), 99–106. 2

[PC*19] PEYRÉ G., CUTURI M., ET AL.: Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning 11*, 5-6 (2019), 355–607. 5

[PDBW23] PO R., DONG Z., BERGMAN A. W., WETZSTEIN G.: Instant continual learning of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), pp. 3334–3344. 8

[RJL*24] REN T., JIANG Q., LIU S., ZENG Z., LIU W., GAO H., HUANG H., MA Z., JIANG X., CHEN Y., ET AL.: Grounding dino 1.5: Advance the" edge" of open-set object detection. *arXiv preprint arXiv:2405.10300* (2024). 4

[RLC23] ROSINOL A., LEONARD J. J., CARLONE L.: Nerf-slam: Real-time dense monocular slam with neural radiance fields. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2023), IEEE, pp. 3437–3444. 3, 7

[SEE*12] STURM J., ENGELHARD N., ENDRES F., BURGARD W., CREMERS D.: A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems* (2012), IEEE, pp. 573–580. 7, 9

[SF16] SCHONBERGER J. L., FRAHM J.-M.: Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 4104–4113. 2

[SLKK17] SHIN H., LEE J. K., KIM J., KIM J.: Continual learning with deep generative replay. *Advances in neural information processing systems 30* (2017). 3

[SLOD21] SUCAR E., LIU S., ORTIZ J., DAVISON A. J.: imap: Implicit mapping and positioning in real-time. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 6229–6238. 3

[SLVGO23] SANDSTRÖM E., LI Y., VAN GOOL L., OSWALD M. R.: Point-slam: Dense neural point cloud-based slam. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), pp. 18433–18444. 8, 9

[SMLM24a] SUN S., MIELLE M., LILIENTHAL A. J., MAGNUSSON M.: High-fidelity slam using gaussian splatting with rendering-guided densification and regularized optimization. *arXiv preprint arXiv:2403.12535* (2024). 2, 3

[SMLM24b] SUN S., MIELLE M., LILIENTHAL A. J., MAGNUSSON M.: High-fidelity SLAM using gaussian splatting with rendering-guided densification and regularized optimization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2024, Abu Dhabi, United Arab Emirates, October 14-18, 2024* (2024), IEEE, pp. 10476–10482. URL: https://doi.org/10.1109/IROS58592.2024.10802373, doi:10.1109/IROS58592.2024.10802373. 3

[SP25] STUART L. A., POUND M. P.: 3dgs-to-pc: Convert a 3d gaussian splatting scene into a dense point cloud or mesh. *arXiv preprint arXiv:2501.07478* (2025). 4

[SSMK18] SERRA J., SURIS D., MIRON M., KARATZOGLOU A.: Overcoming catastrophic forgetting with hard attention to the task. In *International conference on machine learning* (2018), PMLR, pp. 4548–4557. 3

[SWM*19] STRAUB J., WHELAN T., MA L., CHEN Y., WIJMANS E., GREEN S., ENGEL J. J., MUR-ARTAL R., REN C., VERMA S., ET AL.: The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797* (2019). 7

[TCY*22] TANCIK M., CASSER V., YAN X., PRADHAN S., MILDENHALL B., SRINIVASAN P. P., BARRON J. T., KRETZSCHMAR H.: Block-nerf: Scalable large scene neural view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 8248–8258. 3

[TRZ*23] TANG J., REN J., ZHOU H., LIU Z., ZENG G.: Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653* (2023). 3

[VdMCT*24] VAN DE MAELE T., CATAL O., TSCHANTZ A., BUCKLEY C. L., VERBELEN T.: Variational bayes gaussian splatting. *arXiv preprint arXiv:2410.03592* (2024). 8

[WDD*24] WU X., DAI P., DENG W., CHEN H., WU Y., CAO Y.-P., SHAN Y., QI X.: Cl-nerf: Continual learning of neural radiance fields for evolving scene representation. *Advances in Neural Information Processing Systems 36* (2024). 3, 8, 9

[WWS*22] WANG A., WANG P., SUN J., KORTYLEWSKI A., YUILLE A.: Voge: a differentiable volume renderer using gaussian ellipsoids for analysis-by-synthesis. *arXiv preprint arXiv:2205.15401* (2022). 2

[WYF*23] WU G., YI T., FANG J., XIE L., ZHANG X., WEI W., LIU W., TIAN Q., WANG X.: 4d gaussian splatting for real-time dynamic scene rendering. *arXiv preprint arXiv:2310.08528* (2023). 3

[XYG*25] XU Y., YU Y., GAN W., WANG T., ZHAN Z., CHENG H., WANG X.: Gaussian on-the-fly splatting: A progressive framework for robust near real-time 3dgs optimization. *arXiv preprint arXiv:2503.13086* (2025). 3, 8

[YC23] YANG S., CAI Z.: Cross domain lifelong learning based on task similarity. *IEEE transactions on pattern analysis and machine intelligence* (2023). 3

[YFW*23] YI T., FANG J., WU G., XIE L., ZHANG X., LIU W., TIAN Q., WANG X.: Gaussiandreamer: Fast generation from text to 3d gaussian splatting with point cloud priors. *arXiv preprint arXiv:2310.08529* (2023). 3

[YGZ*23] YANG Z., GAO X., ZHOU W., JIAO S., ZHANG Y., JIN X.: Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. *arXiv preprint arXiv:2309.13101* (2023). 3

[YHS*24] YU J., HARI K., SRINIVAS K., EL-REFAI K., RASHID A., KIM C. M., KERR J., CHENG R., IRSHAD M. Z., BALAKRISHNA A., KOLLAR T., GOLDBERG K.: Language-embedded gaussian splats (LEGS): incrementally building room-scale representations with a mobile robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2024, Abu Dhabi, United Arab Emirates, October 14-18, 2024* (2024), IEEE, pp. 13326–13332. URL: https://doi.org/10.1109/IROS58592.2024.10802196, doi:10.1109/IROS58592.2024.10802196. 3

[YQW*23] YAN C., QU D., WANG D., XU D., WANG Z., ZHAO B., LI X.: Gs-slam: Dense visual slam with 3d gaussian splatting. *arXiv preprint arXiv:2311.11700* (2023). 8

[YYP*23] YANG Z., YANG H., PAN Z., ZHU X., ZHANG L.: Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. *arXiv preprint arXiv:2310.10642* (2023). 3

[ZBS*23] ZIELONKA W., BAGAUTDINOV T., SAITO S., ZOLLHÖFER M., THIES J., ROMERO J.: Drivable 3d gaussian avatars. *arXiv preprint arXiv:2311.08581* (2023). 3

[ZPL*22] ZHU Z., PENG S., LARSSON V., XU W., BAO H., CUI Z., OSWALD M. R., POLLEFEYS M.: Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 12786–12796. 3

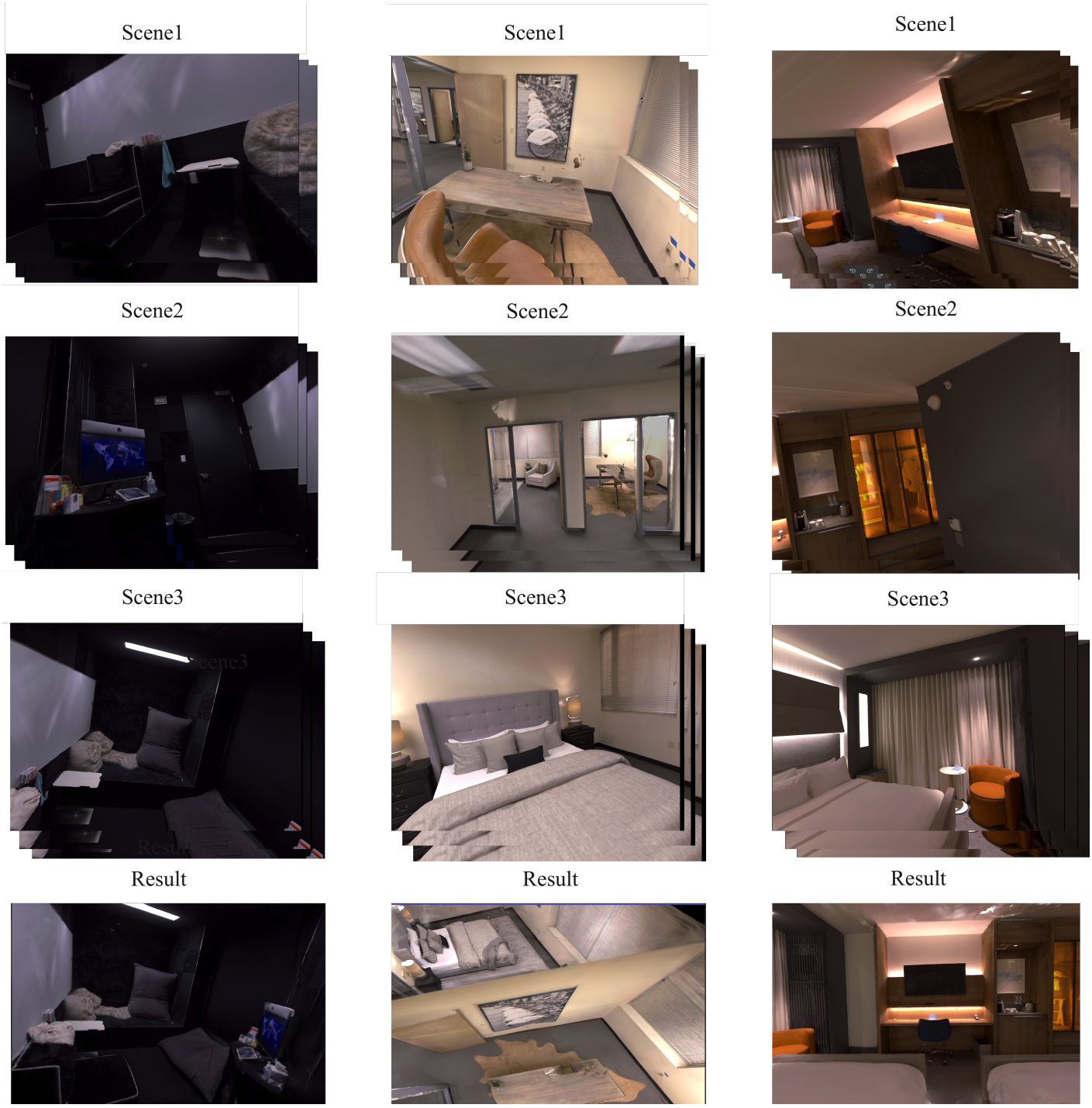

Figure 6: *visualization results of continuously reconstructing three new scenes in Replica dataset.*