

Graph-GSReg: Leveraging 3D Scene Graphs for Gaussian Splatting Registration

Jaewon Lee¹, Mangyu Kong¹, and Euntai Kim^{1,2*}

¹ Yonsei University, Seoul, Republic of Korea

² Korea Institute of Science and Technology, Seoul, Republic of Korea
{leejaewon,mangyu0929,etkim}@yonsei.ac.kr
<https://lee-jaewon.github.io/Graph-GSReg/>

Abstract. Merging multiple 3D Gaussian Splatting (3DGS) scenes into a single unified Gaussian representation is essential for large-scale 3D mapping and long-term map management. Despite its importance, this area remains underexplored, and existing solutions exhibit several limitations. Learning-based methods attempt direct correspondence between Gaussian primitives and require training on large 3DGS datasets. Image-based optimization methods depend heavily on coarse initialization from generic foundation models and often incur expensive refinement. We present Graph-GSReg. Our method constructs a 3D scene graph from a 3DGS and its rendered images, *reformulating 3DGS registration as a graph registration problem*. The proposed 3D scene graph represents each 3DGS at a higher-level representation, enabling a globally consistent understanding of semantic information and structural context for accurate registration. To further construct a seamless unified scene, we introduce a Self-Supervised Test-Time Optimization. Naively merging two 3D Gaussian scenes often suffers from occlusion artifacts such as hollows and floaters. To alleviate this issue, we refine the merged Gaussians to preserve visual consistency between the original scenes and the merged scene. We evaluate our method on real and synthetic benchmarks, demonstrating competitive registration accuracy and merged scene rendering quality.

Keywords: Gaussian Splatting · Scene Graph Registration · Scene Merging

1 Introduction

Large-scale 3D mapping is a fundamental challenge [3] in applications such as robot navigation and AR/VR. Recently, 3D Gaussian Splatting (3DGS) [11] has emerged as a new form of map representation, offering fast rendering speed and high visual quality, and is now being explored for large-scale scene representation [13, 17]. However, due to its optimization-based nature that requires refining

* Corresponding author

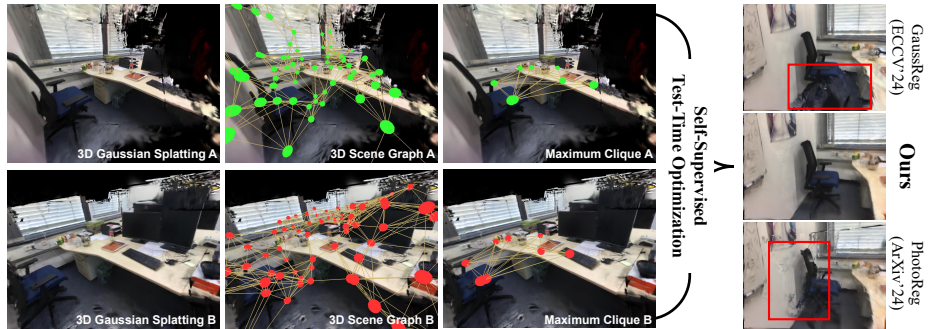


Fig. 1: Graph-GSReg performs 3D Gaussian Splatting (3DGS) registration using 3D scene graphs. Given two partially overlapping 3DGSs, the second column shows the 3D scene graphs constructed from each 3DGS. In the third column, the two scene graphs are matched, and consistent correspondences are obtained through maximum clique search, without erroneous matches.

a massive number of Gaussian primitives, training a single model on large-scale environments leads to excessive memory consumption. This often results in unstable optimization, making it practically infeasible [20, 36]. Furthermore, in real-world scenarios, it is often difficult to capture an entire scene at once [8].

Consequently, in 3DGS, aligning and merging multiple independently generated scenes is essential for large-scale reconstruction, yet this problem has not been sufficiently explored. Several approaches have been proposed for 3DGS registration. GaussReg [4] is a learning-based framework that performs registration using a point cloud network derived from Gaussian primitives and further refines the alignment through a rendered image-based network. However, it requires training on large 3DGS datasets. GaussReg attempts to fuse them based on the distance thresholds between Gaussian points. However, this thresholding-based approach excludes some Gaussians, often resulting in severe hollow regions. PhotoReg [33] performs coarse alignment of rendered image pairs using DUST3R [29] and refines the alignment via image-based optimization. However, since such off-the-shelf models are not specifically tailored for 3DGS registration, they can introduce potential ambiguities in the initial estimation. This reliance on image-based alignment means that if coarse alignment fails, fine optimization also requires a significant amount of time. Moreover, PhotoReg does not employ a dedicated merging strategy and instead directly combines the scenes, resulting in noticeable artifacts and floaters.

We propose Graph-GSReg, a framework that enables robust alignment and seamless merging of multiple scenes through a 3D scene graph representation [9, 10, 15]. Although 3DGS offers high-fidelity rendering, its primitives mainly encode low-level attributes such as color, position, and rotation, making them difficult to directly exploit for higher-level reasoning. To address this limitation, we construct a 3D scene graph from a 3DGS and reformulate 3DGS registration as a graph registration problem, enabling consistent alignment across multiple scenes.

To seamlessly merge multiple 3DGS scenes into a unified representation, we further introduce a Self-Supervised Test-Time Optimization (TTO) strategy. Naively merging 3DGS scenes often suffers from occlusion artifacts such as hollows and floaters. To mitigate this, we use the fact that each original scene remains renderable as a self-supervised reference. Specifically, we render both the merged scene and the corresponding original scene from the same viewpoint, and update the merged Gaussians by minimizing the difference between the two renderings. The renderings from the original scenes serve as reliable references that guide the refinement of the unified representation. This process alleviates artifacts caused by overlapping structures and occlusions while maintaining photometric consistency across the scene. Fig. 1 showcases 3D scene graphs from real-world 3DGS and provides a brief qualitative comparison of our merging results.

The contributions of this paper are summarized as follows:

- We propose Graph-GSReg, a registration framework that constructs a 3D scene graph enriched with geometric and semantic information from 3DGS scenes.
- We propose Self-Supervised Test-Time Optimization, a merging strategy that refines aligned 3D Gaussians and yields a seamless and visually consistent scene.
- We evaluate our method on both real-world and synthetic benchmarks, achieving competitive registration accuracy and high-quality merged-scene renderings.

2 Related Work

2.1 Scene Merging Strategies in Radiance Field

Several studies have explored aligning and seamlessly merging two scenes in Radiance Field. Based on NeRF [19], Block-NeRF [27] partitions large-scale scenes into multiple blocks and synthesizes them using inverse distance weighting (IDW) and appearance alignment. NeRFuser [6] proposes a re-rendering-based registration and a sample-level IDW-Sample blending method to fuse two NeRFs. In methods based on 3D Gaussian Splatting (3DGS) [11], LoopSplat [37] registered between loop keyframes detected via NetVLAD [1] within the SLAM framework, and the additional merging process was performed in the SLAM backend optimization. GaussReg [4] introduces an end-to-end network for 3DGS scene registration, where the coarse stage aligns scenes using point cloud registration networks [22], and the fine stage refines them through a rendered image-based fine registration network that extracts volumetric features from overlapping views. For scene merging in GaussReg, a distance-based thresholding strategy is applied, preserving Gaussians closer to their own scene center. However, this heuristic thresholding approach relies solely on the scene centers, and may erroneously discard important regions when the two scenes differ in size or distribution. Another 3DGS merging method, PhotoReg [33], renders

all images from each scene, selects the image pair with the highest similarity, and then employs a 3D foundation model such as DUS3R [29] to estimate the initial registration pose. PhotoReg then refines the alignment via photometric-based optimization for a fixed number of iterations. Although DUS3R enables image-level registration, failures in the initial stage often affect the subsequent fine optimization, resulting in overall performance degradation. Moreover, when merging two scenes, PhotoReg simply combines them without any additional processing.

2.2 Graph-based Representations for Registration

Graph-based representations are an effective way to transform complex scenes into interpretable forms. In 3D point cloud registration, initial correspondences are modeled as nodes and their geometric compatibility as edges, structurally capturing relations between features. This approach better captures inter-feature dependencies than Euclidean distance-based methods, while the graph connectivity ensures robustness under noise and low overlap by facilitating reliable matching and outlier rejection. Graph-based registration methods can be divided into learning-based and non-learning-based approaches. Recently, SGAligner [26] and SG-PGM [31] have been proposed as learning-based methods. Such approaches leverage graph representations for registration, using GNNs [2] or Transformer-based models [7] to encode structural and semantic relationships that support robust correspondence matching and transformation estimation. Non-learning-based methods rely on the graph structure directly. Previous studies [14, 21, 35] extract geometrically consistent inlier correspondences using maximum clique or maximal clique search. Recent methods such as CLIP-Clique [18] enrich nodes with multi-modal features like CLIP [23] embeddings. They select the highest-scoring clique among candidates to perform global localization. TEASER++ [32] represents correspondences as a graph and performs robust inlier selection, which enables accurate 3D point cloud registration under heavy outlier conditions. Graph-based approaches are not limited to geometric reasoning. When combined with semantic and visual features, they show significantly improved performance. In this work, we render images from 3D Gaussian Splatting (3DGS) scenes and construct them into a consistent 3D scene graph for registration. The proposed method integrates both semantic and structural information to achieve accurate and robust scene registration.

3 Method

Graph-GSReg registers two 3D Gaussian Splatting (3DGS) models, G^A and G^B , and seamlessly merges them into a unified 3DGS scene. Unlike existing 3DGS registration methods that typically rely solely on either geometric or photometric cues, Graph-GSReg is designed to convert 3DGS into a 3D scene graph as a higher-level representation. Through this approach, the 3DGS registration

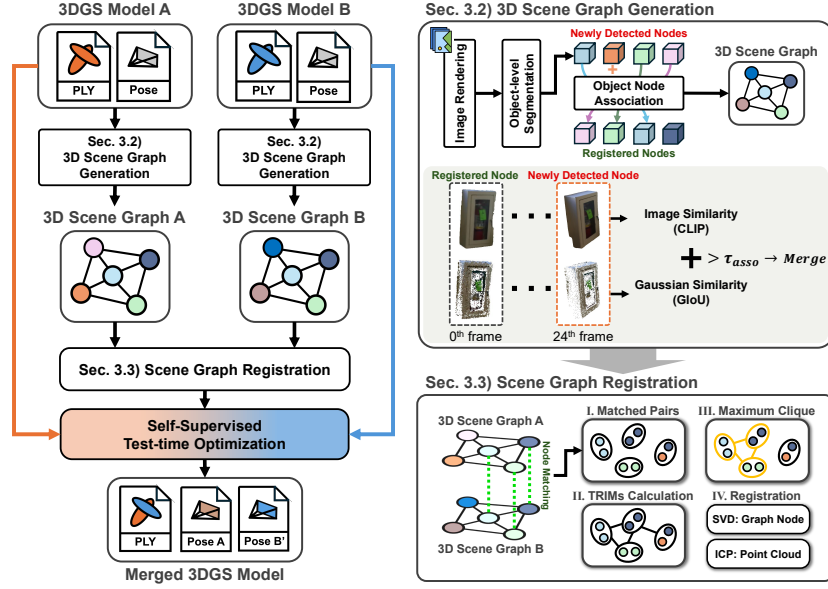


Fig. 2: Overall architecture of Graph-GSReg. Starting from independently constructed 3DGS models, Graph-GSReg constructs an object-level 3D scene graph from the 3DGS scenes by associating the same objects across frames to form globally consistent nodes (Sec. 3.2). It then aligns two scenes by matching their scene graphs to build a compatibility graph, where TRIMs-based filtering and clique selection produce reliable correspondences and the final registration (Sec. 3.3).

problem is reformulated as a graph registration problem. This graph representation jointly encodes rich semantic cues and structural context, enabling robust registration. Finally, to construct a seamless unified scene, we introduce a Self-Supervised Test-Time Optimization that uses the original scenes as references to preserve the visual consistency of the merged scene. While we focus on the registration of two models for notational simplicity and clarity, extending this framework to multiple scenes is a natural progression once pairwise registration is established. Fig. 2 illustrates the overall architecture of our proposed method.

3.1 Problem Setup

The goal of Graph-GSReg is to merge two unregistered 3D Gaussian Splatting (3DGS) models. Each scene is defined as

$$S^X = (G^X, C^X), \quad X \in \{A, B\}, \quad (1)$$

where G^X is the set of Gaussian primitives, and C^X is the set of camera poses. Each scene $X \in \{A, B\}$ consists of a set of Gaussian primitives

$$G^X = \{(\mu_i, \Sigma_i, \sigma_i, \mathbf{c}_i)\}_{i=1}^{N_G^X}, \quad (2)$$

where $\boldsymbol{\mu}_i \in \mathbb{R}^3$ is the mean position, $\boldsymbol{\Sigma}_i \in \mathbb{R}^{3 \times 3}$ is the covariance matrix, σ_i is the opacity, and $\mathbf{c}_i$ is the color attribute of the i -th primitive. Here N_G^X denotes the number of Gaussian primitives in scene X . The set of camera poses for each scene is defined as $C^X = \{(R_j^X, T_j^X)\}_{j=1}^{N_C^X}$, where N_C^X denotes the total number of image frames in each scene. The final transformation used to align scene B to scene A is defined as a rigid transformation $\mathbf{T}_B^A = (\mathbf{R}, \mathbf{t})$, $\mathbf{T}_B^A \in SE(3)$, consisting of a rotation matrix $\mathbf{R}$ and a translation vector $\mathbf{t}$. This transformation is applied only to all components of S^B . After applying the transformation, all camera poses of scene B are updated as

$$C^{B'} = \{(\mathbf{R}R_j^B, \mathbf{R}T_j^B + \mathbf{t})\}_{j=1}^{N_C^B}. \quad (3)$$

The transformation of the Gaussian primitive set is denoted, for notational simplicity, as $G^{B'} = \mathbf{T}_B^A(G^B)$. This represents the result of applying the rigid transformation to all Gaussians in scene B . The transformation of each Gaussian’s position, covariance, and color is performed in the same way as GaussReg [4]. For a robust evaluation, the initial states of the two scenes are set with random orientations and large displacements to ensure no spatial overlap.

3.2 3D Scene Graph from Gaussian Splatting

Since independently reconstructed 3DGS scenes often contain different Gaussian primitives and inconsistent structures, establishing correspondences directly between Gaussians is unreliable. To address these inconsistencies, we convert each 3DGS scene into a higher-level 3D scene graph representation that jointly captures rich semantic cues and structural information. This globally consistent graph enables accurate registration.

The proposed method constructs a graph utilizing mask-level visual cues and their corresponding Gaussians, obtained from per-frame rendered images of each pre-built 3DGS scene S^X , where $X \in \{A, B\}$. We construct a globally consistent 3D scene graph $\mathcal{G}^X$ for the entire scene by associating nodes corresponding to the same object across multiple frames. The graph is defined as $\mathcal{G}^X = (\mathcal{V}^X, \mathcal{E}^X)$, consisting of a node set $\mathcal{V}^X$ and an edge set $\mathcal{E}^X$. Since our method constructs the scene graph through cross-frame object association, rendering every frame densely is unnecessary. Accordingly, we render images I_j^X only from a subset of camera poses C_{sampled}^X uniformly sampled from the entire set.

$$I_j^X = \text{Render}(G^X \mid (R_j^X, T_j^X)), \quad j \in C_{\text{sampled}}^X, \quad X \in \{A, B\}, \quad (4)$$

where *Render* denotes the 3DGS rendering procedure.

As shown in Fig. 2, we extract object masks from the rendered images using SAM [12], utilizing mask information above a certain confidence threshold c_{mask} . Denoting the k -th specific mask in the j -th frame as $m_{j,k}^X$, we first extract the image feature of the corresponding mask region via CLIP [23] to exploit the rich information of the high-fidelity rendered image. Then, utilizing the current frame’s camera pose (R_j^X, T_j^X) and camera intrinsics $\mathbf{K}$, we determine whether

the Gaussians in the 3D space belong to the mask. Specifically, the Gaussian center $\boldsymbol{\mu}_i$ is projected onto the 2D image plane at pixel coordinates $\mathbf{p}_{i,j}^X = [u_{i,j}^X, v_{i,j}^X]^T$ according to

$$z_{i,j}^X \begin{bmatrix} \mathbf{p}_{i,j}^X \\ 1 \end{bmatrix} = \mathbf{K}(R_j^X)^T(\boldsymbol{\mu}_i - T_j^X). \quad (5)$$

Here, $z_{i,j}^X$ denotes the depth value along the Z-axis in the camera coordinate system. To identify the set of 3D Gaussians corresponding to the object mask $m_{j,k}^X$, we define the subset $\mathcal{A}_{j,k}^X$ consisting of Gaussians whose projected pixel coordinates fall within the mask region and are located in front of the camera ($z_{i,j}^X > 0$)

$$\mathcal{A}_{j,k}^X = \{\boldsymbol{\mu}_i \in G^X \mid \mathbf{p}_{i,j}^X \in m_{j,k}^X, z_{i,j}^X > 0\}. \quad (6)$$

By disregarding attributes such as rotation and scale, the extracted set $\mathcal{A}_{j,k}^X$ can be treated as a 3D point cloud. The mean of the 3D coordinates $\mathbf{x}_{j,k}^X \in \mathbb{R}^3$ in this set is defined as the 3D position of the corresponding object node.

$$\mathbf{x}_{j,k}^X = \frac{1}{|\mathcal{A}_{j,k}^X|} \sum_{\boldsymbol{\mu}_i \in \mathcal{A}_{j,k}^X} \boldsymbol{\mu}_i \quad (7)$$

As illustrated in the concrete example in Fig. 2, the image features of the masks and their corresponding Gaussians are used for node association. When compared with nodes appearing in subsequent frames, if the sum of the image feature similarity of mask and the 3D GIoU score between the two Gaussian coordinate sets exceeds the threshold τ_{asso} , the two nodes are considered to represent the same object and are merged. Otherwise, the object is regarded as unseen and added as a new node. Specifically, during the merge process, the two point clouds are combined and then downsampled to maintain uniform density. Then, we average the two CLIP features and normalize the result. Finally, to generate as many candidates as possible and identify inliers among them, the nodes are initially constructed as a fully connected graph.

Based on this graph structure, we further enrich each node representation by incorporating structural context. Appearance-only matching between nodes is often ambiguous when objects share similar visual patterns. To incorporate structural context, we augment each node’s CLIP feature with a histogram of CLIP features in its 3-hop neighborhood. While the fully connected graph is used to consider broad candidate relations, the 3-hop neighborhood histogram is computed on a local graph that connects only spatially nearby nodes within a distance threshold. Therefore, this histogram reflects the local context around each object rather than global graph statistics. The final node embedding is defined by concatenating the normalized CLIP feature with the 3-hop neighborhood CLIP histogram,

$$f(v) = [f_{\text{CLIP}}(v), f_{\text{hist}}(v)], \quad \forall v \in \mathcal{V}^X, \quad (8)$$

which captures both local appearance and the distribution of appearance cues in its neighborhood.

3.3 Clique-driven Scene Graph Registration

Building on this rich information, rather than matching nodes independently, we seek correspondences that can be jointly explained across all nodes. Therefore, we formulate scene graph registration as a global consensus problem rather than a node-wise matching task. Candidate correspondences are generated using context-enriched node similarity and refined via maximum clique search [14, 32] to identify the largest set of mutually consistent correspondences.

Compatibility Graph with TRIMs. From the constructed two scene graphs $\mathcal{G}^A = (\mathcal{V}^A, \mathcal{E}^A)$ and $\mathcal{G}^B = (\mathcal{V}^B, \mathcal{E}^B)$, we estimate a globally consistent transformation $\mathbf{T}_B^A$ between the scenes. Rather than treating candidate correspondences independently, we construct a compatibility graph $\mathcal{G}_{\text{comp}} = (\mathcal{V}_{\text{comp}}, \mathcal{E}_{\text{comp}})$ that captures pairwise geometric consistency among candidate matches.

Candidate correspondences are generated using node embedding similarity, defined as

$$s(q, p) = \frac{f(q) \cdot f(p)}{\|f(q)\| \|f(p)\|}, \quad \mathcal{V}_{\text{comp}} = \{(q', p') \mid s(q, p) \geq \tau_{\text{node}}, q \in \mathcal{V}^A, p \in \mathcal{V}^B\}. \quad (9)$$

But objects that are visually highly similar, yet located at different spatial positions, can still produce incorrect correspondences. To eliminate such mismatches, we note that for correct correspondences, the distance between any two matched objects in one scene should be consistent with the distance between the corresponding objects in the other scene. Accordingly, we employ TRIMs [32] to enforce this pairwise distance consistency, and define edges in the compatibility graph as

$$((q'_1, p'_1), (q'_2, p'_2)) \in \mathcal{E}_{\text{comp}} \iff 1 - \tau_{\text{TRIMs}} < \frac{\|q'_1 - q'_2\|}{\|p'_1 - p'_2\|} < 1 + \tau_{\text{TRIMs}}. \quad (10)$$

This constraint suppresses geometrically inconsistent matches while preserving correspondences that are mutually compatible under a shared transformation. Consequently, geometrically consistent correspondences become densely connected in the compatibility graph, providing a strong foundation for maximum clique search to identify the largest globally consistent correspondence set.

Maximum Clique & Registration. Under the TRIMs-based distance invariance constraint, correct correspondences preserve relative distances across scenes and therefore remain mutually connected in the compatibility graph. In contrast, incorrect correspondences may satisfy the distance consistency with some matches but violate it with others. As a result, they fail to form a globally consistent set of correspondences and cannot produce a fully connected structure in the compatibility graph.

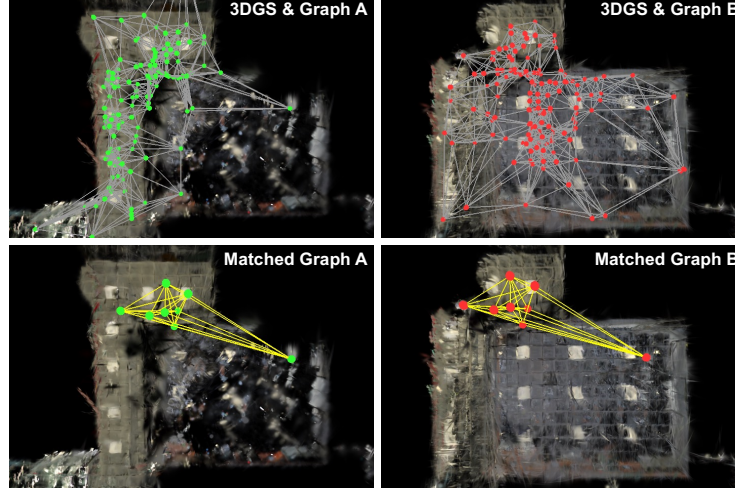


Fig. 3: Visualization of the 3D scene graphs from two 3DGS scenes constructed from uHumans2 [24].

Consequently, geometrically consistent correspondences form fully connected subgraphs in the compatibility graph. Based on this observation, we perform a maximum clique search in the compatibility graph.

Fig. 3 shows the node matching process in large-scale indoor environments with partial overlap, where the first row shows the initial nodes obtained from node association and the second row shows the correspondences after TRIMs-based filtering and maximum clique selection. The structural agreement between the two final graphs shows that incorrect matches are effectively removed while preserving geometrically consistent correspondences.

Using the extracted correspondence set, we estimate a rigid transformation $\mathbf{T}_{\text{graph}} \in SE(3)$ using an SVD-based method. When scale estimation is required, the Umeyama algorithm [28] is applied. The alignment is further refined by performing ICP on the means of the Gaussian primitives initialized with $\mathbf{T}_{\text{graph}}$, yielding the final transformation $\mathbf{T}_B^A \in SE(3)$.

3.4 Self-Supervised Test-Time Optimization for Seamless Merging

The primary purpose of Graph-GSReg is to seamlessly fuse independently reconstructed scenes without information loss or visual degradation. However, inherent imperfections in rigid registration often lead to overlapping artifacts, occlusions, and redundant densities when naively aligning two Gaussian sets.

To address artifacts, occlusions, and redundant densities caused by inherent imperfections in 3D registration, we introduce a Self-Supervised Test-Time Optimization (TTO) strategy. Instead of relying on heuristic pruning [4], our core insight is to leverage the original scenes as reliable rendering references. Although the ground-truth for the unified scene is absent, each individual scene,

S^A and the transformed $S^{B'}$, is still renderable from its original viewpoints. We utilize these source renderings as references to guide the refinement of the unified representation. Specifically, after alignment, we first construct a naive union of the two Gaussian sets, $G^A \cup G^{B'}$. To suppress redundant Gaussians in the partially overlapping regions and establish a memory-efficient, stable initialization, we apply a spatial voxelization with a resolution of 0.01 m to this union. Let G^{merged} denote the Gaussian parameters of this initial merged scene, and let $C^{\text{merged}} = C^A \cup C^{B'}$ denote the combined set of all camera poses.

During the TTO phase, we refine the merged scene using renderings from the original scenes as references. From the same viewpoint, we first render the original scene to obtain a reference image $I_j^X = \text{Render}(G^X | R_j^X, T_j^X)$, and then render the merged scene using G^{merged} from the same viewpoint. The merged Gaussians are updated by minimizing the photometric discrepancy between the two renderings. Formally, the merged scene is optimized by minimizing the following L_1 rendering loss

$$\arg \min_{G^{\text{merged}}} \sum_j \| \text{Render}(G^{\text{merged}} | R_j^X, T_j^X) - I_j^X \|_1, \quad X \in \{A, B'\}. \quad (11)$$

Unlike geometric thresholding based solely on distance [4], this self-supervised mechanism is *based on the expectation that the merged scene should preserve the full content of the original scenes*. Under this assumption, it naturally mitigates occlusion artifacts and maintains photometric consistency across the global scene. Moreover, this refinement achieves strong performance with only one minute of optimization and produces a visually seamless and complete scene representation.

4 Experiments

4.1 Experimental Setup

In this work, we evaluate both registration and merging performance on the real-world dataset ScanNet-GSReg [4] and the synthetic dataset uHumans2 [24]. For registration performance (Sec. 4.2), we report Relative Rotation Error (RRE), Relative Translation Error (RTE), Relative Scale Error (RSE), and Absolute Translation Error (ATE). For merging performance (Sec. 4.3), we compare rendered and ground-truth images using PSNR, SSIM [30], and LPIPS [34].

To evaluate performance in larger indoor environments, we conduct additional experiments on a dataset constructed from uHumans2. Specifically, we generate 69 scene pairs from four static RGB-D sequences without dynamic objects. Each 3DGS model is reconstructed from 150 frames obtained by temporally subsampling the RGB-D sequence at 5 fps, and each scene pair contains 51 overlapping frames. All comparisons are performed by perturbing the initial transformation with random noise, and the reported metrics represent the mean error across all pairs. All experiments are conducted on an NVIDIA RTX 4070 and an Intel Core i5-13600.

Table 1: Quantitative comparison of Gaussian Splatting registration results on the ScanNet-GSReg [4] dataset. (* indicates values reported in the original papers.)

Method	RRE ($^{\circ}$) $\downarrow$	RTE $\downarrow$	RSE $\downarrow$	Time (s) $\downarrow$
GaussReg* (Coarse) [4]	3.403	0.061	0.034	3.700
GaussReg* (w/ Fine) [4]	2.827	0.042	0.032	4.800
PhotoReg (Coarse) [33]	7.825	0.082	-	16.231
PhotoReg (w/ Fine) [33]	7.366	0.072	-	585.216
Liu et al.* [16]	2.595	0.045	0.013	5.041
Ours (Coarse)	3.247	0.039	0.013	2.784
Ours (w/ Fine)	1.970	0.025	-	3.658

Table 2: Quantitative comparison of Gaussian Splatting registration results on the uHumans2 [24] dataset.

Method	RRE ($^{\circ}$) $\downarrow$	RTE $\downarrow$	ATE (m) $\downarrow$	Time (s) $\downarrow$
TEASER++ & ICP [32]	9.099	0.071	1.669	4.570
MAC (FPFH) [35]	108.089	0.652	17.118	6.359
MAC (FCGF) [35]	60.021	0.335	8.862	5.462
PhotoReg (Coarse) [33]	15.895	0.154	6.916	17.814
PhotoReg (w/ Fine) [33]	9.959	0.077	2.754	342.754
Ours (Coarse)	1.748	0.010	0.243	1.805
Ours (w/ Fine)	0.711	0.006	0.192	3.782

Our method requires a preprocessing step that re-renders a pre-built 3DGS to generate a scene graph. However, this step is lightweight compared to the training cost of learning-based approaches and is performed only once per scene. The constructed graph can be reused across multiple registration tasks.

Additional implementation details, extended analyses, and qualitative results are provided in the supplementary material.

4.2 3DGS Registration Evaluation

As reported in Tab. 1, we evaluate 82 partially overlapping scene pairs from ScanNet-GSReg [4]. GaussReg [4] performs both coarse and fine registration using supervised training and requires extensive datasets of completed 3DGS models. PhotoReg [33] does not require training, but it relies only on images. Its coarse registration based on a 3D foundation model [29] produces large errors in some scenes, which leads to poor fine alignment. Liu et al. [16] merge 3DGS using skeleton structures and show strong performance among recent methods. Our 3D scene graph-based registration achieves the best final performance, with a clear improvement in RTE. We do not report RSE for PhotoReg, as it estimates the scale of the two scenes separately, or for our ICP-based refinement, which is restricted to SE(3).

We further evaluate registration on the uHumans2 [24] dataset using 69 scene pairs. We focus on methods applicable without large-scale supervised training, including TEASER++ [32], MAC [35], and PhotoReg [33]. TEASER++ extracts FPFH [25] features for global registration and then applies ICP for refinement.

Table 3: Quantitative comparison of Gaussian Splatting merging results on the ScanNet-GSReg [4] and uHumans2 [24] datasets.

Registration	Merging	ScanNet-GSReg [4]			uHumans2 [24]		
		PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
	Oracle	22.2913	0.8566	0.3347	32.0999	0.9218	0.1416
Ground Truth	PhotoReg [33]	20.1588	0.8117	0.3844	20.9777	0.7081	0.3316
	GaussReg [4]	20.9512	0.8306	0.3551	23.3410	0.7368	0.2871
	Graph-GSReg	21.8954	0.8465	0.3484	25.4032	0.7608	0.2801
Graph-GSReg	PhotoReg [33]	19.3298	0.7926	0.4026	20.7897	0.7048	0.3370
	GaussReg [4]	20.1133	0.8089	0.3715	23.1519	0.7321	0.2904
	Graph-GSReg	21.5248	0.8304	0.3698	25.2425	0.7607	0.2868

Meanwhile, MAC constructs compatibility graphs based on features extracted by FPFH or FCGF [5] and solves the registration problem through maximal clique search to identify the most consistent set of correspondences. However, these traditional point cloud-based methods often face challenges in noisy environments where incorrect feature matches lead to coarse alignment failures. In contrast, our 3D scene graph-based registration with ICP refinement achieves stable and high accuracy across all scenes. Detailed runtime analysis is reported in the supplementary material.

4.3 3DGS Merging Quality Evaluation

In this subsection, we compare the rendered image quality of different 3DGS merging methods. The evaluation is based on comparing rendered images of the merged scenes with ground truth images. The Oracle represents an upper bound as it renders each scene independently before merging and compares them with the ground truth. Since no merging is performed, this setting is free from duplication, removal, or artifacts introduced during the merging process.

As shown in Tab. 3, we apply all merging methods to the same fixed registration results produced by Graph-GSReg on ScanNet-GSReg and uHumans2, isolating the effect of the merging strategy. Under this setting, our Self-Supervised Test-Time Optimization-based merging achieves the best image quality. PhotoReg simply merges the aligned scenes without further processing, while GaussReg selects Gaussians based on distance thresholds.

We further evaluate merging under ground-truth alignment and observe the same trend, indicating that the gain comes from the merging process itself rather than relying on registration accuracy.

Fig. 4 and Fig. 5 show qualitative results on ScanNet-GSReg and uHumans2 using the same fixed registration results. GaussReg often produces hollow regions or strong artifacts due to its distance-based selection. PhotoReg accumulates redundant Gaussians, which degrades the scene quality. In contrast, our method effectively reduces incompleteness and floater artifacts.

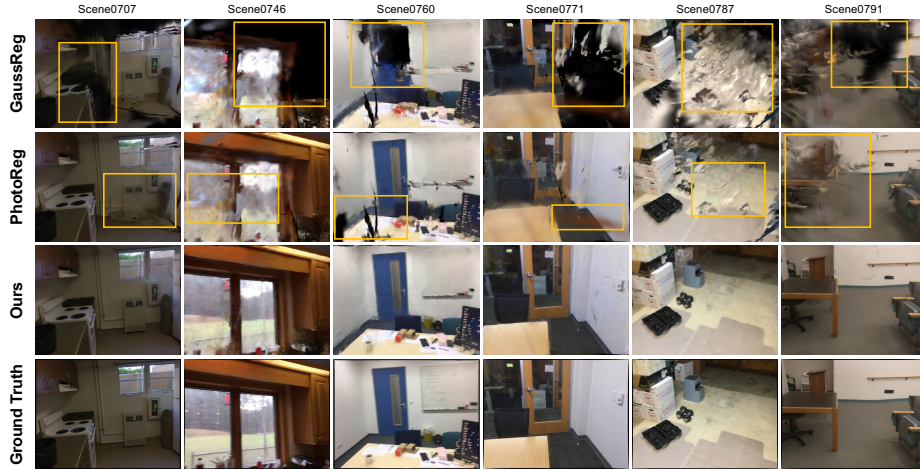


Fig. 4: Qualitative results on the ScanNet-GSReg [4] dataset. Each row shows rendered images from merged Gaussians. The yellow box highlights regions with severe hollows or occlusions.

Table 4: Ablation study of the 3D scene graph registration method on the ScanNet-GSReg [4] dataset.

Method	RRE ↓	RTE ↓	RSE ↓	Time (s) ↓
w/o CLIP Histogram	4.865	0.052	0.019	2.755
w/o TRIMs	6.047	0.140	0.104	2.903
w/o Maximum Clique	6.575	0.151	0.120	2.801
w/ FPFH feature	3.190	0.039	0.013	3.645
CLIP Histogram-5	3.867	0.045	0.015	4.341
Maximal Clique (Best CLIP Score) [18]	3.981	0.045	0.013	2.805
Ours	3.247	0.039	0.013	2.784

Table 5: Ablation study and memory usage comparison of the 3DGS merging method on the ScanNet-GSReg [4] dataset.

Setting	PSNR ↑	SSIM ↑	LPIPS ↓	Size (MB) ↓
Oracle	22.2913	0.8566	0.3347	-
GaussReg [4]	20.1905	0.8106	0.3698	148.18
Ours (w/o Voxelization)	21.4538	0.8304	0.3675	238.10
Ours (w/ Voxelization)	21.5248	0.8304	0.3698	82.84

4.4 Ablation Study

3D Gaussian Splatting Registration. To evaluate the effectiveness of each component, we conducted ablation studies on the ScanNet-GSReg [4] dataset, and the results are summarized in Tab. 4. Incorporating a CLIP histogram improved registration performance compared to not using it. When TRIMs were not applied and all nodes were used, the accuracy decreased, confirming the necessity of geometrically validated node selection. Moreover, skipping clique search after TRIMs in the compatibility graph also led to degraded performance. Adding FPFH [25] features improved accuracy but increased computation time, so we exclude it from the final design, though it may be useful when point clouds are consistent or higher precision is needed. Finally, compared to CLIP-Clique [18], which selects the maximal clique with the highest CLIP similarity among nodes, our maximum clique strategy consistently achieved superior performance by leveraging more nodes. Overall, these results demonstrate that each component contributes to balancing registration accuracy and computational efficiency.

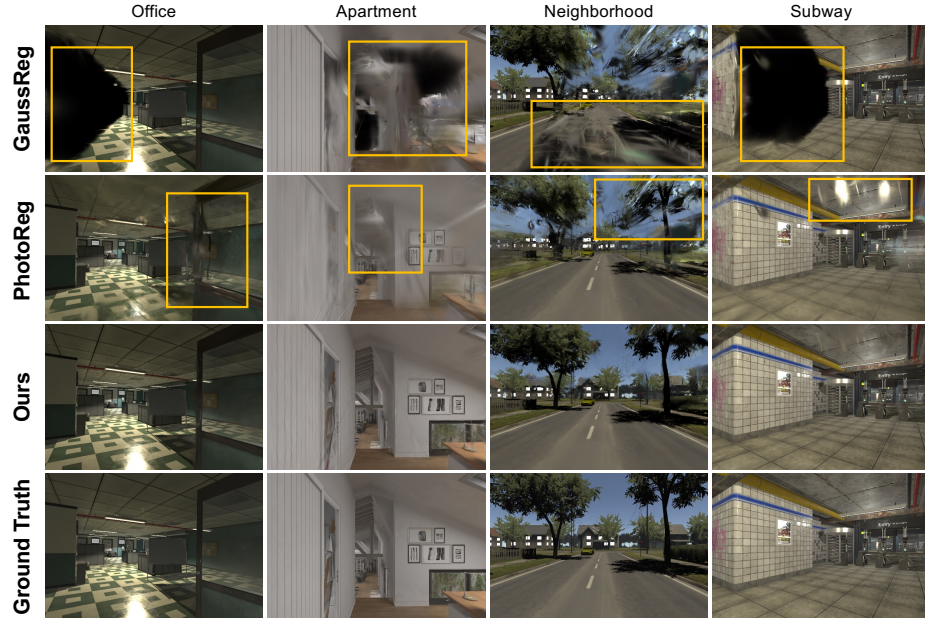


Fig. 5: Qualitative results on the uHumans2 [24] synthetic dataset. Each row shows rendered images from merged Gaussians. The yellow box highlights regions with severe hollows or occlusions.

3D Gaussian Splatting Merging. As discussed in Sec. 3.4, we also perform voxelization after registration before merging the two scenes. If the scenes are simply merged without any additional processing, a large number of duplicated Gaussians remain in the overlapped regions, which leads to a significant increase in storage size. As shown in Tab. 5, such a simple merging setting requires much larger memory compared to GaussReg, which reduces storage by selecting Gaussians based on distance. In contrast, our method applies voxelization before merging, effectively eliminating redundant Gaussians and achieving the smallest storage size, while maintaining or even improving merging performance.

5 Limitations

Our method requires a preprocessing step to construct a scene graph from each 3DGS scene prior to registration. This step involves image rendering and object-level information extraction, which introduces additional computation time. However, the graph is generated only once per scene and can be reused for multiple registration tasks. Moreover, this design removes the need for large-scale training data and can be applied regardless of environment or domain as long as a 3DGS scene is available.

6 Conclusion

We propose Graph-GSReg, a framework for robust registration and seamless merging of 3D Gaussian Splatting (3DGS) scenes. By leveraging 3D scene graphs, we reformulate registration as a graph matching problem and achieve consistent alignment without training. In addition, our Self-Supervised Test-Time Optimization utilizes the existing Gaussian scenes to update merged Gaussians, alleviating occlusions and scene incompleteness and producing seamless merged results. Overall, Graph-GSReg provides an effective solution for large-scale 3DGS-based scene reconstruction and long-term map management in robotics and 3D mapping applications.

References

1. Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: Netvlad: Cnn architecture for weakly supervised place recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 5297–5307 (2016)
2. Brody, S., Alon, U., Yahav, E.: How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491* (2021)
3. Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., Leonard, J.J.: Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics* **32**(6), 1309–1332 (2017)
4. Chang, J., Xu, Y., Li, Y., Chen, Y., Feng, W., Han, X.: Gaussreg: Fast 3d registration with gaussian splatting. In: *European Conference on Computer Vision*. pp. 407–423. Springer (2024)
5. Choy, C., Park, J., Koltun, V.: Fully convolutional geometric features. In: *ICCV* (2019)
6. Fang, J., Lin, S., Vasiljevic, I., Guizilini, V., Ambrus, R., Gaidon, A., Shakhnarovich, G., Walter, M.R.: Nerfuser: Large-scale scene representation by nerf fusion. *arXiv preprint arXiv:2305.13307* (2023)
7. Fu, K., Liu, S., Luo, X., Wang, M.: Robust point cloud registration framework based on deep graph matching. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 8893–8902 (2021)
8. Golodetz, S., Cavallari, T., Lord, N.A., Prisacariu, V.A., Murray, D.W., Torr, P.H.: Collaborative large-scale dense 3d reconstruction with online inter-agent pose optimisation. *IEEE transactions on visualization and computer graphics* **24**(11), 2895–2905 (2018)
9. Gu, Q., Kuwajerwala, A., Morin, S., Jatavallabhula, K.M., Sen, B., Agarwal, A., Rivera, C., Paul, W., Ellis, K., Chellappa, R., et al.: Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. pp. 5021–5028. IEEE (2024)
10. Hughes, N., Chang, Y., Carlone, L.: Hydra: A real-time spatial perception system for 3D scene graph construction and optimization (2022)
11. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics* **42**(4) (July 2023), <https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/>
12. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. In: *Proceedings of the IEEE/CVF international conference on computer vision*. pp. 4015–4026 (2023)
13. Lin, J., Li, Z., Tang, X., Liu, J., Liu, S., Liu, J., Lu, Y., Wu, X., Xu, S., Yan, Y., et al.: Vastgaussian: Vast 3d gaussians for large scene reconstruction. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 5166–5175 (2024)
14. Lin, Y.K., Lin, W.C., Wang, C.C.: K-closest points and maximum clique pruning for efficient and effective 3-d laser scan matching. *IEEE Robotics and Automation Letters* **7**(2), 1471–1477 (2022)
15. Linok, S., Zemskova, T., Ladanova, S., Titkov, R., Yudin, D., Monastyrny, M., Valenkov, A.: Beyond bare queries: Open-vocabulary object grounding with 3d scene graph. In: *2025 IEEE International Conference on Robotics and Automation (ICRA)*. pp. 13582–13589. IEEE (2025)

16. Liu, S., Yang, D., Gao, Y., Ren, B., Yang, Y., Fu, M.: Automated 3d-gs registration and fusion via skeleton alignment and gaussian-adaptive features. *arXiv preprint arXiv:2507.20480* (2025)
17. Liu, Y., Luo, C., Fan, L., Wang, N., Peng, J., Zhang, Z.: Citygaussian: Real-time high-quality large-scale scene rendering with gaussians. In: *European Conference on Computer Vision*. pp. 265–282. Springer (2024)
18. Matsuzaki, S., Tanaka, K., Shintani, K.: Clip-clique: Graph-based correspondence matching augmented by vision language models for object-based global localization. *IEEE Robotics and Automation Letters* (2024)
19. 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), 99–106 (2021)
20. Papantonakis, P., Kopanas, G., Kerbl, B., Lanvin, A., Drettakis, G.: Reducing the memory footprint of 3d gaussian splatting. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* **7**(1), 1–17 (2024)
21. Qiao, Z., Yu, Z., Yin, H., Shen, S.: Pyramid semantic graph-based global point cloud registration with low overlap. In: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. pp. 11202–11209. IEEE (2023)
22. Qin, Z., Yu, H., Wang, C., Guo, Y., Peng, Y., Ilic, S., Hu, D., Xu, K.: Geo-transformer: Fast and robust point cloud registration with geometric transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **45**(8), 9806–9821 (2023)
23. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: *International conference on machine learning*. pp. 8748–8763. PmLR (2021)
24. Rosinol, A., Violette, A., Abate, M., Hughes, N., Chang, Y., Shi, J., Gupta, A., Carlone, L.: Kimera: From slam to spatial perception with 3d dynamic scene graphs. *The International Journal of Robotics Research* **40**(12-14), 1510–1546 (2021)
25. Rusu, R.B., Blodow, N., Beetz, M.: Fast point feature histograms (fpfh) for 3d registration. In: *2009 IEEE international conference on robotics and automation*. pp. 3212–3217. IEEE (2009)
26. Sarkar, S.D., Miksik, O., Pollefeys, M., Barath, D., Armeni, I.: Sgaligner: 3d scene alignment with scene graphs. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 21927–21937 (2023)
27. 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*. pp. 8248–8258 (2022)
28. Umeyama, S.: Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on pattern analysis and machine intelligence* **13**(4), 376–380 (2002)
29. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: Dust3r: Geometric 3d vision made easy. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 20697–20709 (2024)
30. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* **13**(4), 600–612 (2004). <https://doi.org/10.1109/TIP.2003.819861>
31. Xie, Y., Pagani, A., Stricker, D.: Sg-pgm: Partial graph matching network with semantic geometric fusion for 3d scene graph alignment and its downstream tasks.

- In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 28401–28411 (2024)
32. Yang, H., Shi, J., Carlone, L.: Teaser: Fast and certifiable point cloud registration. *IEEE Transactions on Robotics* **37**(2), 314–333 (2020)
 33. Yuan, Z., Zhang, T., Johnson-Roberson, M., Zhi, W.: Photoreg: Photometrically registering 3d gaussian splatting models. *arXiv preprint arXiv:2410.05044* (2024)
 34. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
 35. Zhang, X., Yang, J., Zhang, S., Zhang, Y.: 3d registration with maximal cliques. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 17745–17754 (2023)
 36. Zhao, H., Weng, H., Lu, D., Li, A., Li, J., Panda, A., Xie, S.: On scaling up 3d gaussian splatting training. In: European Conference on Computer Vision. pp. 14–36. Springer (2024)
 37. Zhu, L., Li, Y., Sandström, E., Huang, S., Schindler, K., Armeni, I.: Loopsplat: Loop closure by registering 3d gaussian splats. In: 2025 International Conference on 3D Vision (3DV). pp. 156–167. IEEE (2025)