SG-PGM: Partial Graph Matching Network with Semantic Geometric Fusion for 3D Scene Graph Alignment and Its Downstream Tasks

Yaxu Xie Alain Pagani Didier Stricker
German Research Center for Artificial Intelligence
firstname.lastname@dfki.de

Abstract

Scene graphs have been recently introduced into 3D spatial understanding as a comprehensive representation of the scene. The alignment between 3D scene graphs is the first step of many downstream tasks such as scene graph aided point cloud registration, mosaicking, overlap checking, and robot navigation. In this work, we treat 3D scene graph alignment as a partial graph-matching problem and propose to solve it with a graph neural network. We reuse the geometric features learned by a point cloud registration method and associate the clustered point-level geometric features with the node-level semantic feature via our designed feature fusion module. Partial matching is enabled by using a learnable method to select the top-k similar node pairs. Subsequent downstream tasks such as point cloud registration are achieved by running a pre-trained registration network within the matched regions. We further propose a point-matching rescoring method, that uses the node-wise alignment of the 3D scene graph to reweight the matching candidates from a pre-trained point cloud registration method. It reduces the false point correspondences estimated especially in low-overlapping cases. Experiments show that our method improves the alignment accuracy by 10∼20% in low-overlap and random transformation scenarios and outperforms the existing work in multiple downstream tasks. Our code and models are available here.

1. Introduction

The 3D semantic scene graph [2, 41, 46] is a semantic-rich model for scene representation, which summarizes the scene context in the form of an attributed and directed graph, in which 3D objects and structures as nodes are associated with semantic classes (e.g. sofa, wall), and geometrical semantic relationship between nodes are represented as edges with multiple classes (e.g. *stand_on*, *supported_by*). 3D scene graphs support many applications in spatial understanding, such as global localization for SLAM [14, 17, 23, 32], loop-closure detecting [29], robot navigation [42, 55], visual object grounding [10], graph-to-3D manipulation [9] and augmented reality [37]. One of the main problems of the aforementioned applications is searching for the partial alignment of two or more 3D scene graphs. As illustrated in Figure 1, once the alignment between nodes is found, tasks like localization and navigation can be conducted via point cloud registration within the overlapping area. Alternatively, the determination of whether scene fragments are overlapped or not can be achieved by analyzing the similarity of 3D scene graphs.

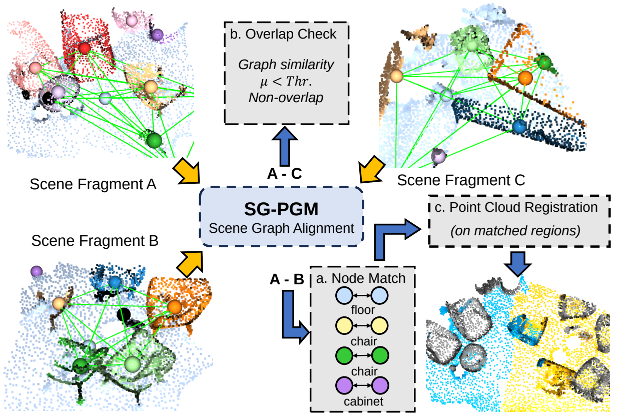


Figure 1. **SG-PGM**: partial graph matching for 3D scene graph alignment. Semantic and geometric features are fused for object-wise matching between fragments (a), and downstream tasks such as (b) overlap-check and (c) point cloud registration.

SGAligner [34] is the first work specifically focusing on this problem. In this work, Sarkar et al. proposed a neural network that learns a joint multi-modal embedding encoded with semantic, geometric, and structural information for each node entity in the graph, which is trained with cross-modal contrastive loss and outputs the similarity between source and reference graph nodes as the alignment result. After the node(object)-level alignment is found, downstream tasks such as point cloud registration are conducted by using a pre-trained registration method, to search point matching within two aligned objects. Later, the point pairs of all aligned objects are fed into a graph-cut RANSAC algorithm [4] to estimate the transformation between the source and reference point clouds. Such decoupled design allows SGAligner to be easily plugged into most of the feature-based registration methods.

However, the simplicity of such a two-stage approach comes with some drawbacks: First, SGAligner employs PointNet [28] to encode object-level geometric embedding. For downstream tasks like point cloud registration or mosaicking, the geometric feature will be extracted twice, once for scene graph alignment and once for registration. We find that reusing the geometric feature extracted from more powerful 3D points encoder like Edge Conv [45], FCGF [8] and KP-Conv [38] is more efficient since they are already integrated into recent registration method [11, 13, 16, 31, 50]. Second, SGAligner achieves a computation complexity less than $O\left(N^2\right)$ (looping through all possible node pairs) by running registration only on the predicted alignment pairs. However, we argue the complexity can be further reduced by introducing explicit mechanisms, that enable one-to-one matching or even partial alignment to surpass false-positive prediction.

Addressing the aforementioned aspects, we first define the 3D scene graph alignment as a **partial graph matching problem**. We build our graph matching neural network following the linear assignment formalism: encoding edge information into node features with the graph convolution and searching node-to-node matching via the Sinkhorn decoder [36]. We reuse the point features learned by the backbone of the point cloud registration method and cluster the point-wise geometric feature into entity nodes via our designed Point to Scene Graph Fusion module (P2SG). We additionally enable explicit partial matching by employing differentiable top-k method [44] to select the k most likely matching pairs. This further increases the alignment accuracy and reduces the false-positive prediction.

Moreover, we design a Superpoint Matching Rescoring method using the predicted scene graph node alignment as the semantic level prior to guiding the point correspondence estimation during registration. We further reduce the search space of point-to-point matching during registration by masking out the non-aligned objects of both scene fragments. We conduct point cloud registration only once between the predicted overlap regions of scene fragments, instead of traversing through node pairs as done in [34]. By employing this strategy, we reduce the inference time, while retaining the long-distance cross-object geometric feature potentially encoded by registration methods [31, 51].

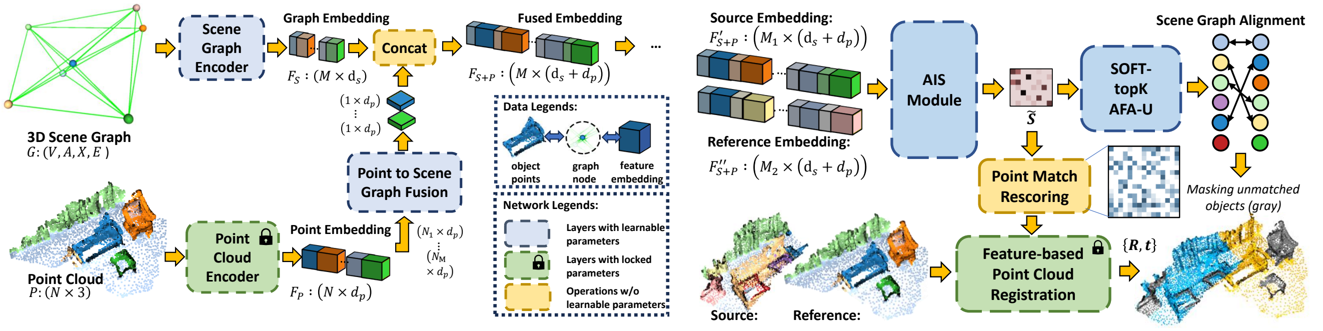
We showcase the effectiveness of our approach, by experimenting with scene graph alignment and its downstream tasks: overlap-checking, point cloud registration, point cloud mosaicking, and alignment with dynamics on the 3RScan [40, 41] dataset. Results show that our approach significantly improves the alignment accuracy by 10∼20% compared to [34], especially when transformation $T \neq I_4$ exists between scene fragments. It reduces the rotation error by 50%, and the translation error by 24% on the point cloud registration task, compared to [34] while keeping the registration RANSAC-free. We also conduct ablation studies to visually demonstrate the effecting mechanism of our proposed Superpoint Matching Rescoring and compare different strategies of using alignment results on registration. We summarize the contributions of this paper as follows:

1. A graph neural network (SG-PGM) for partial graph matching to solve 3D scene graph alignment.
2. The Point to Scene Graph Fusion module and the soft top-k method for increasing alignment accuracy.
3. The Superpoint Matching Rescoring method for guiding the point matching with scene graph alignment results.
4. Revisiting the strategies to stimulate the potential of using 3D scene graph alignment for downstream tasks.

2. Related Work

3D Semantic Scene Graph can be estimated from a video sequence, panoramic image or point could in a bottom-up fashion. Armeni et al. [2] design a semi-automatic framework based on object detector and multi-view consistency and use it to extend the 2D scene graph in [19] into 3D space. Wald et al. [41] present their 3D scene graph dataset extended from 3RScan [40], in which object and structure nodes are annotated with multiple geometric relationships as edges. Their proposed network estimates a 3D semantic scene graph from the point cloud of the scene. Later Wu et al. proposed an incremental method to predict 3D scene graphs from RGB-D [46] and RGB [47] sequence as input. Zhang et al. [54] introduced knowledge learning and knowledge intervention-aided scene graph prediction.

Graph Matching and Subgraph Matching share the same goal of finding the one-to-one alignment between graphs, while the latter is also required to determine the existence of a subgraph isomorphism. NeuroMatch [24] presents the first subgraph matching network that estimates the subgraph relationship with the learned order embedding [26]. Later works [20, 33] estimate the node or edge correspondence between query and target graphs directly, which makes them similar to many general graph matching works [11, 21, 22]. Graph matching is used in many domains of computer vision, such as object key point detection [43, 52] and tracking [15], SfM and SLAM [35]. For 3D scene graph alignment, two graphs are usually partially matched. Wang et al. [44] enable partial graph matching with a differentiable top-k framework to select the most likely matched pairs from the primary one-to-one matching.



(a) Scene Graph and Point Encoding

(b) Scene Graph Alignment and Point Cloud Registration

Figure 2. **The network overview** of the proposed system. (a) shows the feature extraction and our proposed **Point to Scene Graph Feature Fusion** of one single point cloud and its associated 3D scene graph. (b) shows the alignment stage between the source and the reference scene graphs and the registration stage of point clouds with the guidance of our proposed **Superpoint Matching Rescoring** method. We reuse the pretrained point cloud encoder of the point cloud registration method. Its weights are locked during training.

Learning-based Point Cloud Registration can be divided into the end-to-end methods and the feature-based methods. PointNetLK [1] proposes aligning the global descriptors iteratively via Lucas &Kanade algorithm [25]. OMNet [48] introduces overlapping mask prediction into the end-to-end method and enables partial registration. The challenge of low overlap scene-level registration is tackled by Huang et al. [16]. Their proposed Overlap Attention Module extracts co-contextual features between point clouds and predicts overlapping and match-ability scores during early information exchange. Qin et al. propose GeoTransformer [31] that learns transformation-invariant geometric representation on the level of super-point using Transformer [39] with their proposed Geometric Structure Embedding. The correspondence is searched first at the super-point level and then at the point level. A Local-to-Global scheme is designed to solve the transformation with weighted SVD [6].

3D Semantic Graph Alignment and Downstream Tasks are common in the robotics domain, e.g. using semantic information in the scene to improve the localization accuracy and robustness. X-View [14] presents semantic topological graph with nodes assigned with semantic labels and center locations, connected with non-directed edges for global localization. The graph matching is solved by computing the similarity between the random walk descriptors of nodes. Qiao et al. [30] proposed the Object Relation Graph feature that encodes the deep visual and relationship representations of detected objects. After the more unified form [41] of 3D scene graph was defined recently, Sarkar et al. [34] explored in SGAligner 3D scene graph alignment and its downstream applications such as point cloud registration with non-overlap early stopping, point cloud mosaicking, 3D scene alignment with changes. This work proposes the first method for aligning pairs of 3D scene graphs and provides data generation pipelines and benchmarks for each task.

3. Approach

3.1. Scene Graph Matching Network

Problem Definition. A 3D scene graph is a graph model with semantic node and edge attributes: $\mathcal{G} = (\mathcal{V}, \boldsymbol{A}, \boldsymbol{X}, \boldsymbol{E})$. It consists of a finite set of object nodes $\mathcal{V} = \{v_1, v_2, ..., v_M\}$, an adjacency matrix $A \in \{0,1\}^{M \times M}$, a node feature matrix $\boldsymbol{X} \in \mathbb{R}^{M \times \cdot}$ and a edge feature matrix $\boldsymbol{E} \in \mathbb{R}^{M \times M \times \cdot}$. Additionally, each 3D points of the corresponded point cloud $P = \left\{\mathbf{p}_i \in \mathbb{R}^3 \mid i = 1, ..., N\right\}$ is assigned to one specific object node with point-to-object map $O : \{1, 2, \ldots, N\} \rightarrow \{1, 2, \ldots, M\}$.

The 3D scene graph may contain noise due to the imperfect output of graph estimation method [41, 46, 47, 54] and the dynamical scene changes in long-term [40]. Instead of posing the problem as a graph isomorphism search, we formulate the inexact graph matching as optimizing the following objective function:

$$\underset{\mathbf{S}}{\arg\max}\, f(\mathbf{S}; \mathcal{G}_{src}, \mathcal{G}_{ref}), \tag{1}$$

in which $\mathbf{S} \in \{0,1\}^{M_{src} \times M_{ref}}$ is the binary permutation matrix that maps nodes between the source graph $\mathcal{G}_{src}$ and the reference graph $\mathcal{G}_{ref}$. We follow [11, 18, 22] to further relax the constraint from the Quadratic Assignment Problem to the Linear Assignment Problem, and define the objective function $f(\cdot)$ as the negative cross entropy between the ground truth $\mathbf{S}$ and the approximate matching $\tilde{\mathbf{S}}$, which is learned by our neural network $\tilde{\mathbf{S}} = nn\left(\mathcal{G}_{src}, \mathcal{G}_{ref}\right)$.

Partial Graph Matching Network. As illustrated in 2a, our matching network first projects the semantic node features $\boldsymbol{X}$ and semantic edge features $\boldsymbol{E}$ of the source and reference graphs into the graph embedding F_S. We then combine the geometric embedding F_P from the point cloud encoder to form the fused embedding F_{S+P}. In more details, $\boldsymbol{X}$ and $\boldsymbol{E}$ are first encoded into the same dimension d with MLPs, then n-layers GATv2 [7] extract the semantic and topological information of each node in the graph. We built learnable skip connections between layers in the same manner as in [24], which is theoretically proved in [49] to converge more efficiently. Thus, the scene graph encoder outputs multi-layers node embedding $F_S \in \mathbb{R}^{M\times d_s}$ with $d_s = d(n+1)$, as shown in Figure 3.

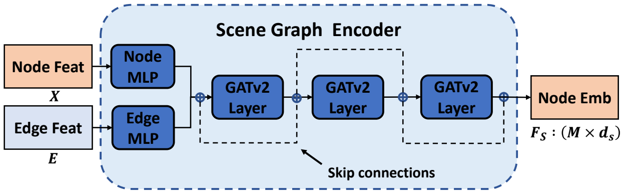


Figure 3. **Scene graph encoder** with GATv2 layers and learnable skip connections.

In the alignment and registration stage (shown in Figure 2b), fused embedding of the source and reference graph is taken by the AIS [13] module to provide a cost matrix that measures the pair-wise similarity. In this module, the joint scene graph and geometric node embedding F^{ref}_{S+P} and F^{src}_{S+P} (see Section 3.2) are used to compute an affinity matrix $\mathbf{A}$ by:

$$\mathbf{A} = F^{ref}_{S+P} \begin{bmatrix} \mathbf{W}_s & 0 \\ 0 & \mathbf{W}_p \end{bmatrix} F^{src}_{S+P}, \tag{2}$$

in which $\mathbf{W}_s$ and $\mathbf{W}_p$ are the learnable weights for computing the affinity of both node embedding. Then $\mathbf{A}$ is normalized via instance normalization and processed by the Sinkhorn [27, 36] operator with an additional row and column of zeros. This enables nodes without correspondence to be matched to the dummy row and column instead. Now we have the soft matching prediction as a doubly-stochastic matrix $\tilde{\mathbf{S}}$, to approximate the one-to-one permutation matrix $\mathbf{S}$.

To explicitly enable partial matching, we employ the pipeline introduced in [44]: the Soft-topK algorithm first flattens $\tilde{\mathbf{S}}$ and selects the K most likely matched candidates, where K is learned by an Attention-fused Aggregation Module [44], more specifically its AFA-U variant. In this module, dummy node features F'_{src} and F'_{ref} (see App. A) are formed into a bipartite graph with $\tilde{\mathbf{S}}$ as the weighted edges, and are brought to a graph attention layer to predict $\tilde{k} \in [0, 1]$ as a graph similarity score, with $K = \tilde{k} \times |M_{ref}|$.

3.2. Point to Scene Graph Feature Fusion

If only considering the semantic information in the 3D scene graph, nodes with the same semantic label and the same edge connection to the other nodes are **symmetric**, e.g. several pillows *lie_on* a sofa. In that case, the subgraph that only consists of these nodes is **automorphism**. Therefore, their graph embedding F_S is identical and it results in unsolvable ambiguity in matching.

Addressing this, we propose to combine the semantic scene graph embedding F_S with the point geometric embedding F_P of each object node, in order to form a more distinguishable joint embedding $F_{S+P} \in \mathbb{R}^{M\times(d_s+d_p)}$. Since our scene graph matching network will cooperate with a feature-based point cloud registration network for solving downstream tasks, it is more efficient to share the point-wise geometric feature encoded by the same backbone network than to introduce another point feature encoder for the same aim.

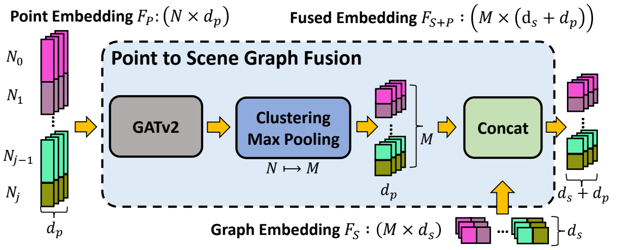

Point Embedding $F_P : (N \times d_p)$
Fused Embedding $F_{S+P} : (M \times (\mathrm{d}_s + d_p))$
Point to Scene Graph Fusion
N_0
N_1
N_{j-1}
N_j
d_p
GATv2
Clustering Max Pooling
$N \longmapsto M$
M
d_p
Concat
$d_s + d_p$
Graph Embedding $F_S : (M \times d_s)$
d_s

Figure 4. **P2SG fusion module** projects point-wise geometric features to node-wise geometric embedding and combines it with the semantic scene graph feature.

As is illustrated in Figure 4, we design this novel Point to Scene Graph Fusion module (P2SG) that projects geometric feature $F_{\mathbf{p}} \in \mathbb{R}^{N\times d_p}$ of N points to object-level feature $F_P \in \mathbb{R}^{M\times d_p}$ of M nodes. The module is defined as:

$$\begin{gathered} F_p^{'} = f_\theta\left(F_p, E_{knn}\right), \\ F_P \in \mathbb{R}^{M\times d_p} \overset{O}{\hookleftarrow} F_p^{'} \in \mathbb{R}^{N\times d_p}, \end{gathered} \tag{3}$$

where E_{knn} is the k-nearest neighbor edges built according to the Euclidean distance between 3d points, and $f_\theta(\cdot)$ is a GATv2 [7] layer for aggregating neighbor features. The clustering max pooling operation $\overset{O}{\hookleftarrow}$ pools the point-wise geometric feature into the node-wise feature with the point-to-object map O.

3.3. Super-point Matching Rescoring

Feature-based point cloud registration methods like GeoTransformer [31] first compare the similarity of points or super-points, to determine the potential point-wise correspondence. Then the transformation can be estimated using weighted SVD [6], RANSAC [12], or its variant [4]. However, only computing the geometric similarity between points will potentially cause incorrect matching, if two points have very similar local geometric features but globally not even belonging to the same object.

We propose the Super-point Matching Rescoring method that uses the semantic similarity learned by our scene graph matching network to reweight the point-wise matching score. Having the scene graph node matching matrix as $\tilde{\mathbf{S}} \in \mathbb{R}^{M\times M}$, the super-point matching matrix[1] $\mathbf{C}$ can be rescored to $\mathbf{C}'$ with:

$$\mathbf{C}' = \mathbf{C} + \gamma \mathbf{R}, \tag{4}$$

where $\mathbf{R} \in \mathbb{R}^{N\times N}$ is the rescoring matrix expanded from $\tilde{\mathbf{S}}$ using the point-to-object maps O_{src} and O_{ref} and $\gamma = 0.2$ is a weighting factor. Because our rescoring method does not introduce any learnable parameters, we do not need to train our method with the point cloud registration method jointly. Therefore, our method can be easily adapted to most feature-based registration methods, bot point-level matching [13] and super-point matching [16, 31, 50].

3.4. Loss Functions

We utilize the Negative Cross-Entropy (NCE) loss in its sparse form to supervise the soft correspondence prediction of scene graph matching. Having $\|\mathbf{S}\|$ as the number of nonzero elements of $\mathbf{S}$, the scene graph matching loss per sample $\mathcal{L}_s$ is defined as:

$$\mathcal{L}_s = \frac{1}{\|\mathbf{S}\|} \sum_{(i,j)}^{\{\mathbf{S}_{(i,j)} \neq 0\}} -\tilde{\mathbf{S}}_{i,j} \log(\mathbf{S}_{i,j}). \tag{5}$$

We compute the ground truth graph similarity k with $k = \|\mathbf{S}\| \,/\, \min(|M_{ref}|\,,|M_{src}|)$ and use Mean Square Error (MSE) loss to supervise the learning of $\tilde{k}$:

$$\mathcal{L}_k = \left(k - \tilde{k}\right)^2 \tag{6}$$

With the weighting factor $\alpha = 10$ and the batch size N, the overall loss per batch is then:

$$\mathcal{L} = \frac{1}{N} \sum_{i}^{N} (\mathcal{L}_s + \alpha \mathcal{L}_k). \tag{7}$$

3.5. Revisiting the Downstream Tasks

Overlap Checking is a direct downstream task of 3D scene graph alignment. Sarkar et al. [34] proposed to compute a scene-level alignment score ξ representing the percentage of aligned nodes against all nodes in the reference graph. It is reported faster and more accurate than first performing point cloud registration on scene fragments and determining overlapping with the matchability score.

However, we find it is an oversimplified solution to only count the number of scene graph alignments, which may fail to distinguish scene fragments with low-overlapping and non-overlapping. Instead, we frame the problem as measuring the graph similarity between scene graphs. Inspired by the two-stages strategy proposed in SimGNN [3], we jointly consider the coarse global graph similarity score of k and the fine-gained node-level similarity $\tilde{\mathbf{S}}$ of all alignment pairs and define the scene-level alignment score μ as:

$$\mu = \tilde{k} \cdot \frac{1}{\|\tilde{\mathbf{S}}\|} \sum_{(i,j)}^{\{\tilde{\mathbf{S}}_{(i,j)} \neq 0\}} \tilde{\mathbf{S}}_{(i,j)} \quad . \tag{8}$$

Point Cloud Registration. The 3D scene graph alignment can be used to reduce the search space of the point-wise matching for the point cloud registration. In SGAligner, the source and reference point clouds are divided into matched object pairs using the estimated graph alignment. Then feature-based point cloud registration is used to search point-wise correspondence traverse through all matched object pairs. Finally, the transformation T is estimated using a robust estimator on all point correspondences.

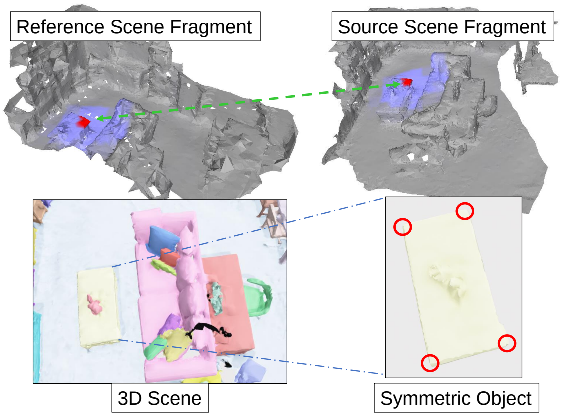


Figure 5. **Long-range cross-object geometric feature** is gathered in registration method [31] with transformer. Points in red circles are difficult to match without taking nearby objects as a reference.

Recent point cloud registration methods [31, 51] successfully encode long-range geometric context with Transformer [39]. As visualized in Figure 5, matching points (colored red) in symmetric objects or planar objects are under-determined if the reference information from other neighbor objects (colored purple) is missing. Dividing the scene fragment into objects will block access to long-range cross-object geometric features and potentially results in less accurate point matching estimation. We simplify this process and use the alignment results to mask out unmatched objects from point clouds and conduct registration on the potential overlapping region only once.

4. Experiments

We evaluate our method for scene graph alignment and overlap-checking (Sec. 4.1), 3D point cloud registration

[1] Please refer Eq.9 in GeoTransfomer [31] for more detail.

and mosaicking (Sec. 4.2) and provide an ablation study (Sec. 4.4). For alignment and registration tasks, we follow the data prepossessing method in [34] and generate 15,277 training samples and 1,882 validation samples from the 3RScan dataset [40, 41]. Sample numbers are different from the original data splits, due to *the uncontrolled random seed* in their implementation. For evaluating the overlap-checking, another 1,882 non-overlap sample pairs are added to the validation subset.

In the following experiments, we ran SGAligner on our generated data splits and marked the results as **SGA*** and listed the results of SGAligner in the original paper as reference. We pick GeoTransformer [31] pretrained on 3DMatch [53] for registration and use its KPConv [38] backbone to extract geometric embedding. For ablation study, we incrementally add our proposed modules to our baseline **B** graph matching network: (1) **B+P** as adding P2SG Fusion, (2) **B+P+K** as adding Soft-topK and AFA-U, (3) **SG-PGM (B+P+K+S)** as adding Super-point Matching Rescoring, (4) **SG-PGM+R** as using Graph-Cut RANSAC [5] for pose estimation. Implementation details and evaluation metrics definitions are in Appendix A and B.

4.1. Scene Graph Alignment and Overlap Checking

We initially evaluate our method for aligning 3D scene graphs using metrics from [34]. However, metrics like Hits@k and Mean Reciprocal Rank (MeanRR) do not account for false-positive matches. Therefore, we also assess our results using the F1-score (the harmonic mean of the precision and recall). We ignore Intra-Graph Alignment Recall metric (IGAR) because ”self-aligned” is by design not allowed in our method.

Methods	Mean RR	F1	Hits @		
			K=1	K=3	K=5
SGA [34]	95.0	-	92.3	97.4	98.7
SGA*	96.3	89.3	94.3	96.9	98.0
B	89.6	62.9	82.2	98.4	99.2
B+P’ (PointNet)	97.5	79.1	95.6	99.5	99.8
B+P (KP-Conv)	**98.7**	79.0	**97.7**	98.7	99.9
B+P+K w/o AIS	94.2	81.8	90.1	97.0	98.4
B+P+K w/ AIS	98.6	**89.4**	97.5	**99.7**	**99.9**

Table 1. **Evaluation on node matching.** We evaluate the scene graph node alignment of our method’s different variants and compare it with SGAligner. All metrics are the-higher-the-better.

As shown in Table 1, adding the proposed P2SG Fusion to the baseline significantly improves the node alignment accuracy and is already higher than SGAligner. With the Soft-topK module, our method can also effectively surpass the false-positive matching pairs and therefore yield the highest F1 score. This is not only important for scene graph alignment but also reduces the inference operations (from an average of 16.6 pairs per sample in the validation set to 12.8 pairs) if we later want to conduct the registration in an object-per-object fashion. Furthermore, we modified our network by using PointNet [28] as the geometric feature extractor (B+P’) for fair comparison with SGAligner, and justify the effectiveness of the AIS Module against simple matrix product for computing node-wise feature similarity (w/o AIS). Since Superpoint Matching Rescoring is only used during registration and has no impact on alignment, we ignore that variant here. Results of alignment on the predicted 3D scene graph are given in Appendix Table 9.

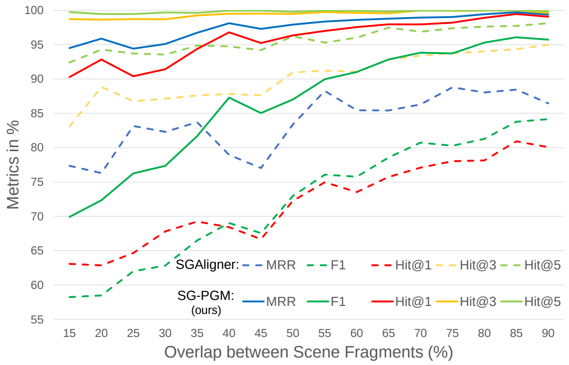


Figure 6. **Evaluation on node matching with transformation** $T \neq I_4$. Results are distributed per overlap range.

We provide a more practical evaluation by augmenting random transformation between two scene fragments, different from the $T = I_4$ benchmark in [34]. We trained SGAligner with random T and Gaussian noise as augmentation (SGA*). The results are divided into different scene overlapping ranges in Figure 6. Even though retrained with augmentation, SGAligner still shows a significant accuracy drop compared to results in Table 1, while the overall performance of our method drops only slightly. This demonstrates that fusing graphs and geometric features with our method is robust against rotation. However, the F1 score of our method in the overlapping range 10-30% is more than 20% lower than in the high-overlap case (see Table 12 in the Appendix). This means that our method provides more false-positive matching in low-overlap cases compared to high-overlap cases, but still much better than [34].

Overlap check of two scene fragments. To check overlaps, we report scene fragment pairs with the scene-level alignment score $\mu < 0.375$ as non-overlapped scenes in Table 2. As mentioned, our method provides more false-positive results in low-overlapping scenarios. Therefore, we suggest using the top 3 of $\tilde{\mathbf{S}}$ scores instead of all $\tilde{\mathbf{S}}$ in Eq. 8 and report non-overlapping with $\mu_3{}' < 0.45$. This variant (SG-PGM@3) suppresses the impact of false-positive node alignments and yields better performance. We also analyze the confusion between low-overlap and non-overlap. While SGAligner predicts about **36%** of low-overlap samples as non-overlap, our method (Ours@3) predicts only **16%** of low-overlap samples incorrectly. An extended experiment on overlap-checking is given in Appendix Table 14.

Methods	Prec.	Recall	F1
SGA [34]	92.03	90.94	91.48
SGA*	93.29	90.34	91.79
SG-PGM (ours)	94.59	92.03	93.29
SG-PGM@3 (ours)	**95.41**	**95.01**	**95.21**

Table 2. **Overlap check for point cloud registration.** $T = I_4$ between fragments. All metrics are the-higher-the-better.

4.2. Point Cloud Registration and Mosaicking

In this section, we use the scene graph alignment result from SGAligner and our method's variants as priors, to support pretrained GeoTransformer [31] for point cloud registration and mosaicking. We evaluate the registration accuracy with Chamfer Distance (CD), Relative Rotation and Translation Error (RRE and RTE), Feature Matching Recall (FMR) and Registration Recall (RR). As shown in Table 3, our method outperforms SGAligner in 4 out of 5 metrics even without a robust estimator (Ours+R). Our registration strategy is also 4 times faster than SGAligner.

Methods	CD	RRE	RTE	FMR	RR
GeoTr [31]	0.0312	2.3726	4.14	98.50	98.37
SGA [34]	0.0111	1.012	1.67	**99.85**	99.40
SGA*	0.0130	1.2929	2.11	99.74	98.87
SG-PGM (ours)	**0.0083**	0.6252	1.32	99.73	**99.57**
SG-PGM+R (ours)	0.0102	**0.5103**	**1.27**	99.73	99.47

Table 3. **3D point cloud registration** with $T = I_4$. Graph-Cut RANSAC [5] is used in "Ours+R" and SGAligner. FMR and RR: higher-the-better. Others: lower-the-better.

We further increase the difficulty of registration and augment the point clouds with random transformation T, as shown in Table 4. The aim of scene graph alignment before registration is to filter non-overlap parts and encourage point matching within object pairs. We design the Semantic Consistency of Point Correspondence (SCC) metric, to measure the consistency between predicted point pairs and the ground truth scene graph node pairs:

$$SCC = \frac{1}{|\mathbf{C}|} \sum_{(i,j)}^{(i,j)\in\mathbf{C}} f(i,j) \quad ,$$
$$f(i,j) = \begin{cases} 1 & \text{if } O_{src}(j) = \mathbf{S}(O_{ref}(i)) \\ 0 & \text{if } O_{src}(j) \neq \mathbf{S}(O_{ref}(i)) \end{cases}, \tag{9}$$

in which $\mathbf{C}$ is the point-level matching between the reference and source point cloud. O is the point-to-object map and $\mathbf{S}$ is the ground truth scene graph alignment.

More accurate scene graph alignment can filter more non-overlapped objects and reduce the search space of the registration method. It explains the accuracy improvement from the B+P variant to the B+P+K variant of our method. Without our Superpoint Rescoring Method, the registration accuracy in low-overlap cases (10-30%) is merely better than [31], though the overall performance is better. After adding the Superpoint Rescoring Method, our complete pipeline shows the best performance in all overlapping ranges, especially improving SCC with a large margin. This shows the effectiveness of guiding point matching with semantic priors in low-overlap scenarios.

Mtds.	Overlap	RRE	RTE	FMR	RR	SCC
GeoTr [31]	10-30	8.2130	19.40	92.47	92.73	76.98
	30-60	0.4584	1.53	99.76	99.76	88.68
	60 -	0.2126	1.02	100.0	99.85	90.34
	overall	1.9398	4.96	98.37	98.37	86.90
B+P	10-30	10.169	22.53	93.33	91.11	78.98
	30-60	0.6513	1.56	99.75	99.63	90.51
	60-	0.1594	0.65	100.0	99.86	91.24
	overall	2.2864	5.23	98.62	98.09	88.58
B+P+K	10-30	8.9309	19.04	94.44	92.22	81.85
	30-60	0.2597	0.90	99.75	99.75	91.15
	60-	0.1598	0.66	100.0	100.0	91.67
	overall	1.8807	4.28	98.83	98.41	89.57
SG-PGM (ours)	10-30	7.3368	15.24	97.22	93.61	87.60
	30-60	0.2419	0.86	100.0	99.88	93.66
	60-	0.1564	0.60	100.0	100.0	93.89
	overall	**1.5668**	**3.51**	**99.47**	**98.72**	**92.59**

Table 4. **3D point cloud registration per overlap.** Random transformation is augmented to the scene fragments. Comparison against GCNet [56] is in Appendix Table 13.

Point cloud mosaicking is the task of registering a set of partial point clouds to reconstruct the completed scene. As proposed in [34], the mosaicking is conducted by running pairwise registration for all pairs. We select 143 scenes for testing point cloud mosaicking and the results are listed in Table 5. We use the same metrics as in [34] to evaluate the results: accuracy and completeness of the resulting reconstruction (the-lower-the-better), precision, recall, and F1-score of registered point clouds (the-higher-the-better). As expected, our method shows higher accuracy than others. Qualitative results of registration and mosaicking are given in Appendix E.

Methods	Acc	Comp	Prec	Recall	F1
GeoTr [31]	0.1213	0.0917	95.84	87.17	90.11
SGA [34]	0.0094	0.0935	90.87	97.44	93.58
SG-PGM (ours)	0.0033	0.0040	99.81	99.79	99.80
SGPGM+R (ours)	**0.0024**	**0.0026**	**99.86**	**99.85**	**99.86**

Table 5. **Point cloud mosaicking from multiple fragments.** Our method outperforms others even without using RANSAC.

4.3. Aligning 3D Scenes with Changes

3RScan dataset provides multiple rescans of one scene with changes such as moved, removed, and deformed objects.

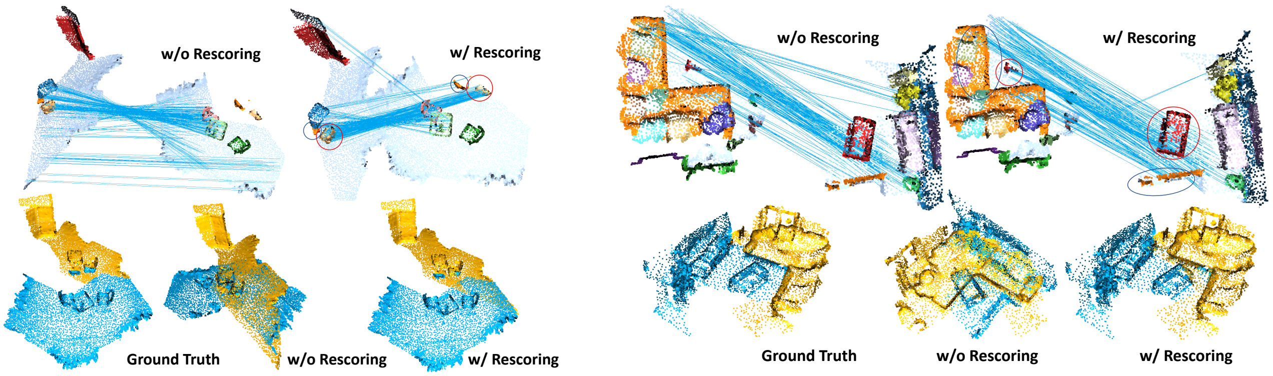


Figure 7. **Registration results with and without Superpoint Matching Rescoring of low overlapping scene fragments.**

Following SGAligner [34], we investigate the alignment in the following scenarios: (i) aligning a sub-scene on the original scan that contains no changes; (ii) aligning a 3D sub-scene on a rescan that contains changes; and (iii) aligning sub-scenes that contains changes.

Methods	Dynamics.	MRR	Hits @		
			K=1	K=3	K=5
SGA* [34]	(i)	97.9	96.6	99.1	99.7
	(ii)	93.6	**90.6**	96.2	97.5
	(iii)	88.8	87.1	94.2	96.2
SG-PGM (ours)	(i)	**99.8**	**99.7**	**99.9**	**100**
	(ii)	**94.2**	90.0	**98.2**	**99.3**
	(iii)	**93.4**	**88.9**	**97.7**	**99.2**

Table 6. **Alignment of a local 3D scene to a prior 3D map with differences in overlap and changes.**

We run SGAligner on our generated data samples and list the results together with ours in Table 6. Our approach outperforms SGAligner in most metrics of all three scenarios, which indicates the strong robustness to scene changes. Details about the data generation of these scenarios and extra experiments against various controlled semantic noises are in Appendix A and C.

4.4. Ablation Study

We focus only on the rescoring method and the registration strategy. Since the effectiveness of the partial graph matching (SOFT-topK) and feature fusion (P2SG) module has been evaluated and verified in Section 4.1 and 4.2.

Super-point Matching Rescoring As shown in Table 4, with the help of the Super-point Matching Rescoring, our method shows obviously better performance in terms of *SCC* compared to other variants. We visualize the point cloud registration results of our method with or without using the Super-point Matching Rescoring method in Figure 7. From that, we obverse that rescoring the point matching with scene graph alignment as prior, can avoid mismatching of points with similar local geometric features but belonging to a different object or semantic class.

Registration Strategy We build up an experiment to evaluate the registration performance on the same validation split used in 4.2 using the ground truth scene graph alignment and run registration with all-to-all (A2A), object-per-object (OPO) and overlap-to-overlap (O2O) fashions. Results in Table 7 indicate that masking the scene fragments with the perfect overlap region (GeoTr+O2O) yields the best result while traversing through object pairs performs the worst. This also supports our analysis in 3.5.

Methods	RRE	RTE	FMR	RR
GeoTr [31]+A2A	1.9398	4.96	98.37	98.37
GeoTr+OPO	5.9528	15.46	99.63	94.69
GeoTr+O2O	**1.4443**	**3.67**	**99.32**	**99.05**

Table 7. **Ablation study on different registration strategies.**

5. Conclusion

We have presented SG-PGM, a graph neural network for scene graph partial matching. We revisited the geometric feature extraction, partial matching mechanism, and strategies for solving downstream tasks of the existing work [34]. We designed our method to use more expressive geometric features with the point to scene graph fusion module. We proposed the Super-point Rescoring method for boosting point cloud registration with semantic priors. Compared to the existing work [31, 34], our method shows significant performance improvements on scene graph alignment, overlap-checking, point cloud registration, and other downstream tasks. Moreover, our scene graph alignment method remains decoupled from registration and robust to scene dynamics and noises. For future work, we would like to explore the approach for using semantic priors from scene graph alignment to design efficient sparse transformers for geometric feature analysis.

Acknowledgement: The research leading to these results has been partially funded by the German Ministry of Education and Research (BMBF) under Grant Agreement 01IW20009 (RACKET) and the EU Horizon Europe Framework Program under Grant Agreement 101058236 (HumanTech).

References

[1] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. Pointnetlk: Robust & efficient point cloud registration using pointnet. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[2] Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 3d scene graph: A structure for unified semantics, 3d space, and camera. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5664–5673, 2019. 1, 2

[3] Yunsheng Bai, Hao Ding, Song Bian, Ting Chen, Yizhou Sun, and Wei Wang. Simgnn: A neural network approach to fast graph similarity computation. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pages 384–392, 2019. 5

[4] Daniel Barath and Jiri Matas. Graph-cut RANSAC. In *Conference on Computer Vision and Pattern Recognition*, 2018. 2, 4

[5] Daniel Barath and Jiří Matas. Graph-cut ransac. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6733–6741, 2018. 6, 7

[6] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, pages 586–606. Spie, 1992. 3, 4

[7] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *International Conference on Learning Representations*, 2022. 4

[8] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8958–8966, 2019. 2

[9] Helisa Dhamo, Fabian Manhardt, Nassir Navab, and Federico Tombari. Graph-to-3d: End-to-end generation and manipulation of 3d scenes using scene graphs. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 1

[10] Mingtao Feng, Zhen Li, Qi Li, Liang Zhang, XiangDong Zhang, Guangming Zhu, Hui Zhang, Yaonan Wang, and Ajmal Mian. Free-form description guided 3d visual graph network for object grounding in point cloud. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3722–3731, 2021. 1

[11] Matthias Fey, Jan E Lenssen, Christopher Morris, Jonathan Masci, and Nils M Kriege. Deep graph matching consensus. In *International Conference on Learning Representations*, 2019. 2, 3

[12] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 4

[13] Kexue Fu, Shaolei Liu, Xiaoyuan Luo, and Manning Wang. Robust point cloud registration framework based on deep graph matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8893–8902, 2021. 2, 4, 5

[14] Abel Gawel, Carlo Del Don, Roland Siegwart, Juan Nieto, and Cesar Cadena. X-view: Graph-based semantic multi-view localization. *IEEE Robotics and Automation Letters*, 3(3):1687–1694, 2018. 1, 3

[15] Jiawei He, Zehao Huang, Naiyan Wang, and Zhaoxiang Zhang. Learnable graph matching: Incorporating graph partitioning with deep feature learning for multiple object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5299–5309, 2021. 2

[16] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, and Konrad Schindler Andreas Wieser. Predator: Registration of 3d point clouds with low overlap. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2021. 2, 3, 5

[17] Nathan Hughes, Yun Chang, and Luca Carlone. Hydra: A real-time spatial perception system for 3d scene graph construction and optimization. *arXiv preprint arXiv:2201.13360*, 2022. 1

[18] Nils M Kriege, Pierre-Louis Giscard, Franka Bause, and Richard C Wilson. Computing optimal assignments in linear time for approximate graph matching. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 349–358. IEEE, 2019. 3

[19] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017. 2

[20] Zixun Lan, Limin Yu, Linglong Yuan, Zili Wu, Qiang Niu, and Fei Ma. Sub-gmn: The subgraph matching network model. *arXiv preprint arXiv:2104.00186*, 2021. 2

[21] He Liu, Tao Wang, Yidong Li, Congyan Lang, Songhe Feng, and Haibin Ling. Deep probabilistic graph matching. *arXiv preprint arXiv:2201.01603*, 2022. 2

[22] Linfeng Liu, Michael C Hughes, Soha Hassoun, and Liping Liu. Stochastic iterative graph matching. In *International Conference on Machine Learning*, pages 6815–6825. PMLR, 2021. 2, 3

[23] Yu Liu, Yvan Petillot, David Lane, and Sen Wang. Global localization with object-level semantics and topology. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4909–4915. IEEE, 2019. 1

[24] Zhaoyu Lou, Jiaxuan You, Chengtao Wen, Arquimedes Canedo, Jure Leskovec, et al. Neural subgraph matching. *arXiv preprint arXiv:2007.03092*, 2020. 2, 4

[25] Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI'81: 7th international joint conference on Artificial intelligence*, pages 674–679, 1981. 3

[26] Brian McFee and Gert Lanckriet. Partial order embedding with multiple kernels. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 721–728, 2009. 2

[27] G Mena, J Snoek, S Linderman, and D Belanger. Learning latent permutations with gumbel-sinkhorn networks. In *ICLR 2018 Conference Track*. OpenReview, 2018. 4

[28] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2, 6

[29] Zhentian Qian, Jie Fu, and Jing Xiao. Towards accurate loop closure detection in semantic slam with 3d semantic covisibility graphs. *IEEE Robotics and Automation Letters*, 7(2):2455–2462, 2022. 1

[30] Chengyu Qiao, Zhiyu Xiang, and Xinglu Wang. Objects matter: Learning object relation graph for robust camera relocalization. *arXiv preprint arXiv:2205.13280*, 2022. 3

[31] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, and Kai Xu. Geometric transformer for fast and robust point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11143–11152, 2022. 2, 3, 4, 5, 6, 7, 8, 1

[32] Antoni Rosinol, Andrew Violette, Marcus Abate, Nathan Hughes, Yun Chang, Jingnan Shi, Arjun Gupta, and Luca Carlone. Kimera: From slam to spatial perception with 3d dynamic scene graphs. *The International Journal of Robotics Research*, 40(12-14):1510–1546, 2021. 1

[33] Indradyumna Roy, Venkata Sai Velugoti, Soumen Chakrabarti, and Abir De. Interpretable neural subgraph matching for graph retrieval. *AAAI 2022*, 2022. 2

[34] Sayan Deb Sarkar, Ondrej Miksik, Marc Pollefeys, Daniel Barath, and Iro Armeni. Sgaligner : 3d scene alignment with scene graphs. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2023. 1, 2, 3, 5, 6, 7, 8

[35] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 2

[36] Richard Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The annals of mathematical statistics*, 35(2):876–879, 1964. 2, 4

[37] Tomu Tahara, Takashi Seno, Gaku Narita, and Tomoya Ishikawa. Retargetable ar: Context-aware augmented reality in indoor scenes based on 3d scene graph. In *2020 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pages 249–255. IEEE, 2020. 1

[38] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6411–6420, 2019. 2, 6, 3

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3, 5

[40] Johanna Wald, Armen Avetisyan, Nassir Navab, Federico Tombari, and Matthias Nießner. Rio: 3d object instance re-localization in changing indoor environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7658–7667, 2019. 2, 3, 6

[41] Johanna Wald, Helisa Dhamo, Nassir Navab, and Federico Tombari. Learning 3d semantic scene graphs from 3d indoor reconstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3961–3970, 2020. 1, 2, 3, 6

[42] Fan Wang, Chaofan Zhang, Wen Zhang, Cuiyun Fang, Yingwei Xia, Yong Liu, and Hao Dong. Object-based reliable visual navigation for mobile robot. *Sensors*, 22(6): 2387, 2022. 1

[43] Runzhong Wang, Junchi Yan, and Xiaokang Yang. Learning combinatorial embedding networks for deep graph matching. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3056–3065, 2019. 2

[44] Runzhong Wang, Ziao Guo, Shaofei Jiang, Xiaokang Yang, and Junchi Yan. Deep learning of partial graph matching via differentiable top-k. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6272–6281, 2023. 2, 4, 1

[45] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019. 2

[46] Shun-Cheng Wu, Johanna Wald, Keisuke Tateno, Nassir Navab, and Federico Tombari. Scenegraphfusion: Incremental 3d scene graph prediction from rgb-d sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7515–7525, 2021. 1, 2, 3

[47] Shun-Cheng Wu, Keisuke Tateno, Nassir Navab, and Federico Tombari. Incremental 3d semantic scene graph prediction from rgb sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5064–5074, 2023. 2, 3

[48] Hao Xu, Shuaicheng Liu, Guangfu Wang, Guanghui Liu, and Bing Zeng. Omnet: Learning overlapping mask for partial-to-partial point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3132–3141, 2021. 3

[49] Keyulu Xu, Mozhi Zhang, Stefanie Jegelka, and Kenji Kawaguchi. Optimization of graph neural networks: Implicit acceleration by skip connections and more depth. In *International Conference on Machine Learning*, pages 11592–11602. PMLR, 2021. 4

[50] Hao Yu, Fu Li, Mahdi Saleh, Benjamin Busam, and Slobodan Ilic. Cofinet: Reliable coarse-to-fine correspondences for robust pointcloud registration. *Advances in Neural Information Processing Systems*, 34:23872–23884, 2021. 2, 5

[51] Junle Yu, Luwei Ren, Yu Zhang, Wenhui Zhou, Lili Lin, and Guojun Dai. Peal: Prior-embedded explicit attention learning for low-overlap point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17702–17711, 2023. 2, 5

[52] Andrei Zanfir and Cristian Sminchisescu. Deep learning of graph matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[53] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser.

3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *CVPR*, 2017. 6

[54] Shoulong Zhang, Aimin Hao, Hong Qin, et al. Knowledge-inspired 3d scene graph prediction in point cloud. *Advances in Neural Information Processing Systems*, 34:18620–18632, 2021. 2, 3

[55] Fengda Zhu, Xiwen Liang, Yi Zhu, Qizhi Yu, Xiaojun Chang, and Xiaodan Liang. Soon: Scenario oriented object navigation with graph-based exploration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12689–12699, 2021. 1

[56] Lifa Zhu, Haining Guan, Changwei Lin, and Renmin Han. Leveraging inlier correspondences proportion for point cloud registration. *arXiv preprint arXiv:2201.12094*, 2022. 7, 3

SG-PGM: Partial Graph Matching Network with Semantic Geometric Fusion for 3D Scene Graph Alignment and Its Downstream Tasks

Supplementary Material

Abstract

In the supplemental material, we provide additional details about the following:

- *Details on implementation. (Section A),*
- *Evaluation metrics of 3D scene graph alignment and downstream tasks (Secion B),*
- *Evaluation on scene graph alignment with controlled semantic noise and with predicted 3D scene graph (Section C),*
- *Additional ablation study on registration strategy and network variants (Section D),*
- *Visualisation on point cloud registration and point cloud mosaicking (Section E).*

A. Implementation Details

Data Generation for Alignment in Dynamics: To evaluate scene graph alignment in the changing environment Section 4.3, we generate the samples using the sub-scenes in the validation split and the original 3D scene maps from [41, 46]. The dynamics between scan and rescan of the same indoor scene consist of three types: "non-rigid", "removed" and "rigid". We ignore small rigid object changes, whose Euler angles $\alpha + \beta + \gamma < 3°$, and mark them as aligned node ground truth. Thus, the sample numbers of scenarios (i), (ii) and (iii) are 819, 354, and 1,635.

Network and Training: We take the fine-level geometric feature of the KPConv-FPN as the input of our P2SG Fusion module. Same as suggested in [44], the input node embeddings of the AFA-U module are set to **zero vectors** for one graph and **one-hot vectors** for the other graph. Unlike in [44], we train the AFA-U module together with the other parts of the network in one stage. We employ the matching rescoring on the super-point matching stage of [31] because the fine-level points within a super-point are considered most likely to belong to the same object. The training procedure takes 10 epochs with the ADAM optimizer and an initial learning rate of $1e^{-4}$, which decreases by 0.1 every 4 epochs. If not specified, we mask out the unmatched objects of the scene fragments and conduct registration on the overlap region as a whole instead of registration traverse through all matched pairs.

B. Evaluation Metrics

We give the definition of evaluation metrics used in the main paper here. For the same evaluation metric used in multiple tasks, its definition will be adjusted based on input.

B.1. Scene Graph Alignment

Hits@K describes the fraction of true entities that appear in the first k entities of the sorted rank list R of the alignment prediction $\tilde{\mathbf{S}}$. Denoting the set of individual ranks as r_i, it is given as:

$$H_k(r_1, ..., r_n) = \frac{1}{n}\sum_{i}^{n}[r_i < k] \quad \in [0, 1] \tag{1}$$

where $[\cdot]$ is the Iversion bracket.

Mean Reciprocal Rank (MRR) is the arithmetic mean over the reciprocals of ranks of true triples:

$$MRR(r_1, ..., r_n) = \frac{1}{n}\sum_{i}^{n}\frac{1}{r_i} \quad \in (0, 1] \tag{2}$$

F1-score is the harmonic mean of the precision and recall. More specifically, the F1 score for graph matching is defined as:

$$\begin{aligned} tp,\ fp,\ fn &= \tilde{\mathbf{S}}\mathbf{S},\ \tilde{\mathbf{S}}(1-\mathbf{S}),\ (1-\tilde{\mathbf{S}})\mathbf{S} \\ F1 &= \frac{2tp}{2tp + fp + fn} \quad \in [0, 1]\,. \end{aligned} \tag{3}$$

B.2. Overlap Checking

Overlap checking of two 3D scenes is a binary classification problem that checks whether two 3D scenes overlap or not. Metrics (Precision, Recall, and F1-score) are given as:

$$\begin{aligned} Prec. &= \frac{TP}{TP + FP} \quad \in [0, 1]\,, \\ Recall &= \frac{TP}{TP + FN} \quad \in [0, 1]\,, \\ F1 &= 2\frac{Prec. \times Recall}{Prec. + Recall} \quad \in [0, 1]\,, \end{aligned} \tag{4}$$

in which TP is true positive, FP is false positive and FN as false negative.

B.3. Point Cloud Registration

Registration Recall (RR) is the fraction of successfully registered point cloud pairs. A point cloud pair is successfully registered when its transformation error is lower than threshold $\tau_1 = 0.2m$. In addition, the transformation error is the root mean square error of the ground truth correspondence C, to which the estimated transformation $\tilde{\mathbf{T}}$ has applied:

$$\begin{aligned} RMSE &= \sqrt{\frac{1}{|C|}\sum_{(p_x, q_y)\in C}\left\|\tilde{\mathbf{T}}(\mathbf{p}_x) - \mathbf{q}_y\right\|_2^2}, \\ RR &= \frac{1}{M}\sum_{i=1}^{M}[RMSE < \tau_1] \quad \in [0, 1]\,, \end{aligned} \tag{5}$$

where p_x and q_y denote the x-th point in source P and y-th point in reference Q, respectively; $[\cdot]$ is the inerson bracket; and M is the number of all point cloud pairs.

Feature Matching Recall (FMR) is the fraction of point cloud pairs whose Inlier Ration (IR) is above $\tau_3 = 0.05$. FMR measures the potential success during the registration, while Inlier Ratio is the fraction of inlier correspondences among all hypothesized correspondences $\tilde{C}$:

$$IR = \frac{1}{\left|\tilde{C}\right|} \sum_{(p_x, q_y) \in \tilde{C}} \left[\left\| \mathbf{T}(\mathbf{p}_x) - \mathbf{q}_y \right\|_2 < \tau_2 \right] \quad \in [0,1] \,,$$
$$FMR = \frac{1}{M} \sum_{i=1}^{M} [IR > \tau_3] \quad \in [0,1] \,, \tag{6}$$

in which an inlier is defined as the distance between the two points is lower than a certain threshold τ_2 under the ground-truth transformation $\mathbf{T}$.

Relative Rotation Error (RRE) measures the geodesic distance in degrees between the estimated $\tilde{\mathbf{R}}$ and ground truth rotation $\mathbf{R}$ matrices:

$$RRE = \arccos(\frac{trace(\mathbf{R}^T \tilde{\mathbf{R}}) - 1)}{2}). \tag{7}$$

Relative Translation Error (RTE) measures the Euclidean distance between the estimated $\tilde{\mathbf{t}}$ and ground truth translation $\mathbf{t}$ vectors:

$$RTE = \left\| \mathbf{t} - \tilde{\mathbf{t}} \right\| . \tag{8}$$

Modified Chamfer Distance measures the average of the pair-wise nearest distance between two point sets P and Q:

$$CD = \frac{1}{|P|} \sum_{p \in P} \min_{q \in Q} \left\| \tilde{\mathbf{T}}(\mathbf{p}) - \mathbf{q} \right\|_2^2 +$$
$$\frac{1}{|Q|} \sum_{q \in Q} \min_{p \in P} \left\| \mathbf{q} - \tilde{\mathbf{T}}(\mathbf{p}) \right\|_2^2 \tag{9}$$

B.4. Point Cloud Mosaicking

Having the ground truth point cloud P and reconstructed point cloud P^*. The **Reconstruction Accuracy (Acc)** and **Reconstruction Completeness (Comp)** are defined as:

$$Acc = \frac{1}{n} \sum_{p \in P}^{n} \min_{p^* \in P^*} (\|p - p^*\|)$$
$$Comp = \frac{1}{n} \sum_{p^* \in P^*}^{n} \min_{p \in P} (\|p - p^*\|) \tag{10}$$

And the **Reconstruction Precision (Prec.)** and **recall (Recall)** and the **F1-score** are defined as:

$$Prec. = \frac{1}{n} \sum_{p \in P}^{n} \min_{p^* \in P^*} [\|p - p^*\| < 0.05] \quad \in [0,1] \,,$$
$$Recall = \frac{1}{n} \sum_{p^* \in P^*}^{n} \min_{p \in P} [\|p - p^*\| < 0.05] \quad \in [0,1] \,, \tag{11}$$
$$F1 = 2 \frac{Prec. \times Recall}{Prec. + Recall} \quad \in [0,1] \,.$$

C. Evaluation on Scene Graph Alignment with Controlled Semantic Noise and with Predicted 3D Scene Graph

We also test the robustness of our network against controlled noise on scene graph node alignment. Following the same implementation of SGAligner [34], we evaluate our method with 5 different types of noises: (i) only relationships are removed; (ii) only object(node) are removed their corresponding attributes and any relationships that include them are also removed; (iii) both relationships and object nodes are removed; (iv) object instances assigned with the wrong semantic label); and (v) both relationships and objects are both assigned with wrong semantics. Results are given in Table 8. We also list the noise-free result here as a reference.

Noise Types	Mean RR	F1	Hits @ K=1	Hits @ K=3	Hits @ K=5
(i)	96.70	77.52	94.93	98.56	98.80
(ii)	97.81	78.41	96.02	99.69	99.94
(iii)	96.86	77.15	94.43	99.35	99.89
(iv)	85.18	69.71	77.99	90.69	94.75
(v)	85.14	69.05	77.81	90.57	95.02
noise-free	97.91	88.39	96.24	99.66	99.93

Table 8. **Evaluation on node matching with different variants of controlled semantic noise**.

Our method shows very strong robustness against missing relationships (edges) and missing instances (nodes). In (iv) and (v), wrong instance semantic information shows relatively strong impacts on the alignment performance compared to wrong relationships. For testing the use of predicted 3D scene graphs instead of ground truth graphs, we generated predicted 3D scene graphs using [41] and tested our network (only trained on the ground truth) on the alignment task. Since the authors of [34] did not publish their code or pre-trained model for using predicted 3D scene graph, we **cannot guarantee a fair comparison** with their results. Table 9 reproduces theirs as in [34] compared with **ours on our validation set**.

Methods	Mean RR	F1	Hits @ K=1	Hits @ K=3	Hits @ K=5
SGA [34]	88.2	-	83.3	91.8	95.1
B+P+K	95.9	86.0	93.1	98.6	99.4

Table 9. **Evaluation on node matching with predicted graph.**

D. Additional Ablation Study

D.1. Object-per-Object Registration with Ours

Same as SGAligner [34], we conduct object-per-object point cloud registration following with RANSAC using the scene graph alignment results of our own network. To further improve the robustness of the object-to-object registration, we propose two methods: (1) The dense scene graph alignment result $\mathbf{S}$ is first filtered with a confidence threshold s, only when the score of object pairs is higher than s will be considered in point cloud registration. If none of the object pairs has a score higher than s, all object pairs are taken for registration, and (2) only top-k-scored object pairs will be used in registration. We also give the registration results of using our network with overlap-to-overlap (O2O) and using SGAligner (S*.) with O2O as references in Table 10. Our network combined with OPO registration performs marginally worse than with O2O registration, while for SGAligner the situation is the converse.

Methods	CD	RRE	RTE	FMR	RR
$s = 0$	0.0544	4.9849	12.31	99.37	96.00
$s = 0.3$	0.0581	4.8246	12.74	99.37	95.74
$s = 0.5$	0.0462	3.9634	9.74	99.26	96.39
$k = 3$	0.0627	5.1250	13.61	99.37	95.95
$k = 5$	0.0514	4.7141	11.76	99.37	96.27
$k = 7$	0.0574	5.0628	12.97	99.37	95.90
O2O	**0.0083**	**0.6252**	**1.32**	**99.73**	**99.57**
S*. + O2O	0.0179	1.3428	2.67	99.26	98.95

Table 10. **Object-per-Object Point Cloud Registration with our method.** Methods with s represent filter object pairs with confidence scores lower than the threshold, while methods with k take only the top-k object pairs for registration.

D.2. Fusion with Different Levels of Point Feature

KPConv-FPN [38] provides multi-level point geometric features of a point cloud. In the original implementation of Geotransformer, there are three levels of geometric features: coarse-level $N_c \times 1024$, middle-level $N_m \times 512$ and fine-level $N_f \times 256$. Here we give a comparison of using different levels of geometric features for the P2SG fusion module in terms of 3D scene graph alignment in Table 11. As the result shows, P2SG fusion with fine-level geometric features performs the best among all listed variants.

Methods	Mean RR	F1	Hits @		
			K=1	K=3	K=5
Coarse	97.00	85.51	94.69	99.33	99.79
Middle	97.85	87.67	96.24	99.58	99.83
Fine	**98.58**	**89.39**	**97.49**	**99.68**	**99.90**

Table 11. **Evaluation on node matching with different levels of point geometric feature.**

D.3. Alignment with Augmented Transformation

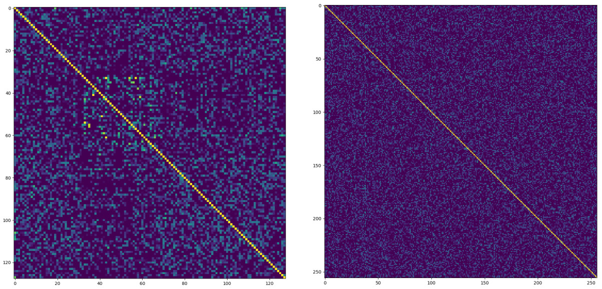
Here we provide the 3D scene graph alignment results with augmented T in Table 12 as the complementary of Figure 6.

Mtds.	Overlap (%)	Mean RR	F1	Hits @		
				K=1	K=3	K=5
SG-PGM (*ours*)	10-30	94.96	74.86	91.23	98.69	99.65
	30-60	97.91	87.95	96.33	99.54	99.87
	60-	99.15	95.21	98.48	99.83	99.93
	overall	**97.81**	**88.18**	**96.16**	**99.49**	**99.85**
SGA* [34]	10-30	79.93	60.46	64.64	86.54	93.50
	30-60	83.20	71.84	71.25	89.61	95.28
	60-	87.24	81.05	78.01	93.75	97.48
	overall	85.92	79.46	77.69	88.07	93.71

Table 12. **Evaluation of our proposed method on node matching per overlap range.** Even in low-overlap cases, our method still provides accurate alignment results with Hit@1 over 90%.

D.4. Analyse of AIS Module

Equation 2 gives the definition of the affinity matrix, in which the affinity of the embeddings from the scene graph and the point cloud is separately computed. In Figure 8, we provide a visualization of the learnable parameters W_s and W_p . As shown in the Figure, the multi-level scene graph embedding is more coupled crossing different feature channels, especially of the first-hop graph embedding, while the geometric feature is relatively more decoupled.



(a) W_s (b) W_p

Figure 8. **The learnable parameters W_s and W_p of the AIS Module.**

D.5. Additional comparison with GCNet on point cloud registration and overlap checking

We tested GCNet [56] on the registration task on our validation set in Table 13. We additionally combined our method with GCNet to mask out the feature points from unmatched objects before the Consistent Voting, which shows improvement compared to GCNet alone.

Methods	RRE	RTE	FMR	RR
GeoTr [31]	1.94	4.96	98.37	98.37
GeoTr + Ours	**1.57**	**3.51**	**99.47**	**98.72**
GCNet [56]	2.24	5.43	98.88	98.51
GCNet + Ours	1.96	4.91	99.09	98.72

Table 13. **Additional evaluation on point cloud registration.**

We also tested GCNet on the overlap checking task, using the average of the top 25% of predicted overlap score vector o and saliency score vector s. In Table 14, we report GCNet with $o_{25\%} \cdot s_{25\%} > 0.45$ as overlap, and the results of using the scene-level score k instead of Eq. 8 in our method. It shows a huge drop in Prec. because our partial graph matching module is only trained with overlapping samples.

Methods	Prec.	Recall	F1
SGA [34]	92.03	90.94	91.48
GCNet [56]	93.43	92.24	92.83
SG-PGM w/ $k > 0.45$	89.94	96.87	93.28
SG-PGM@3 (ours)	**95.41**	**95.01**	**95.21**

Table 14. **Overlap check for point cloud registration.**

E. Qualitative Results

Here we provide some qualitative results by combining our method and GeoTransformer [31] for point cloud registration in Figure 9 and for point cloud mosaicking in Figure 10.

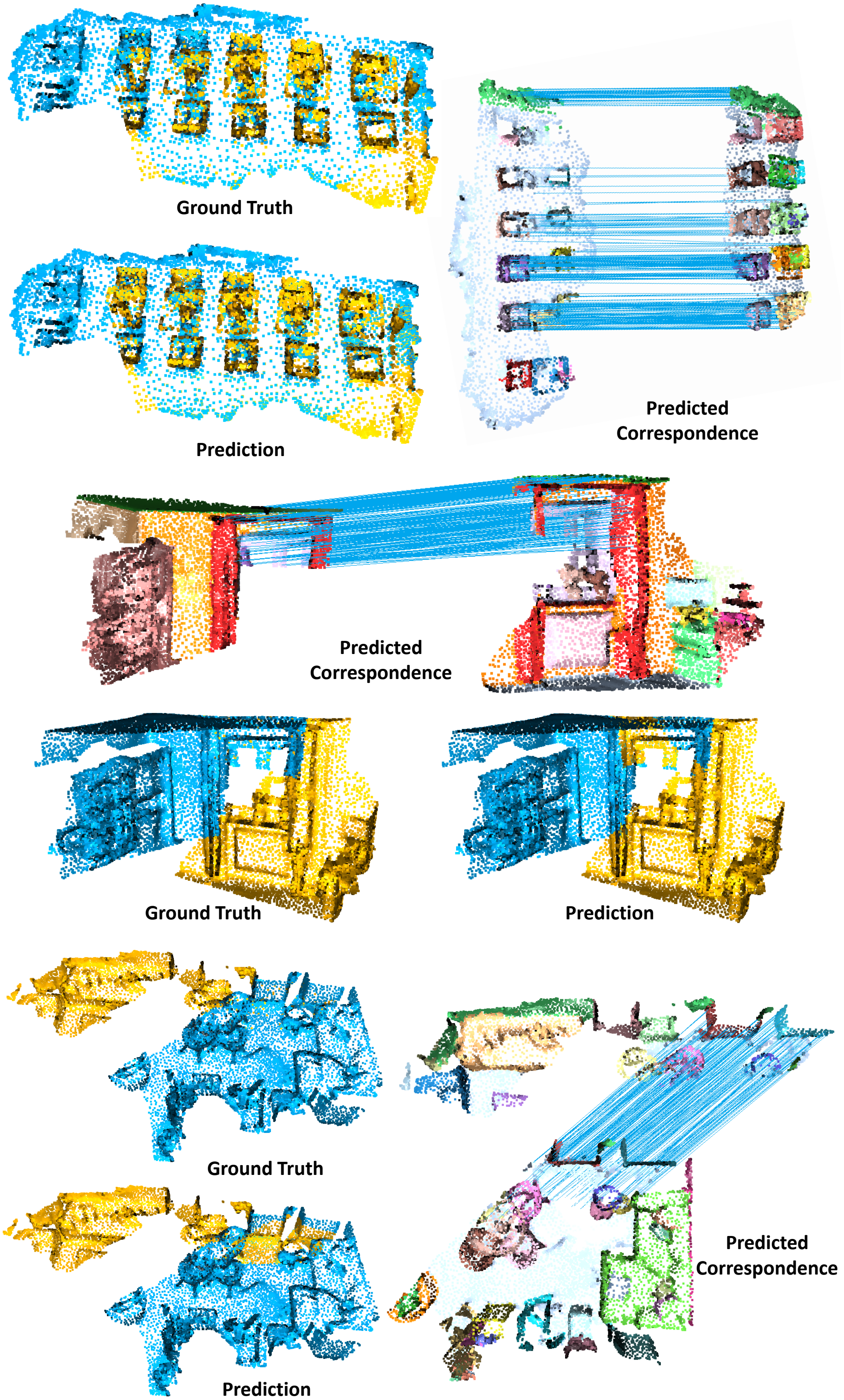


Figure 9. **Qualitative Results on Point Cloud Registration** of our proposed method.

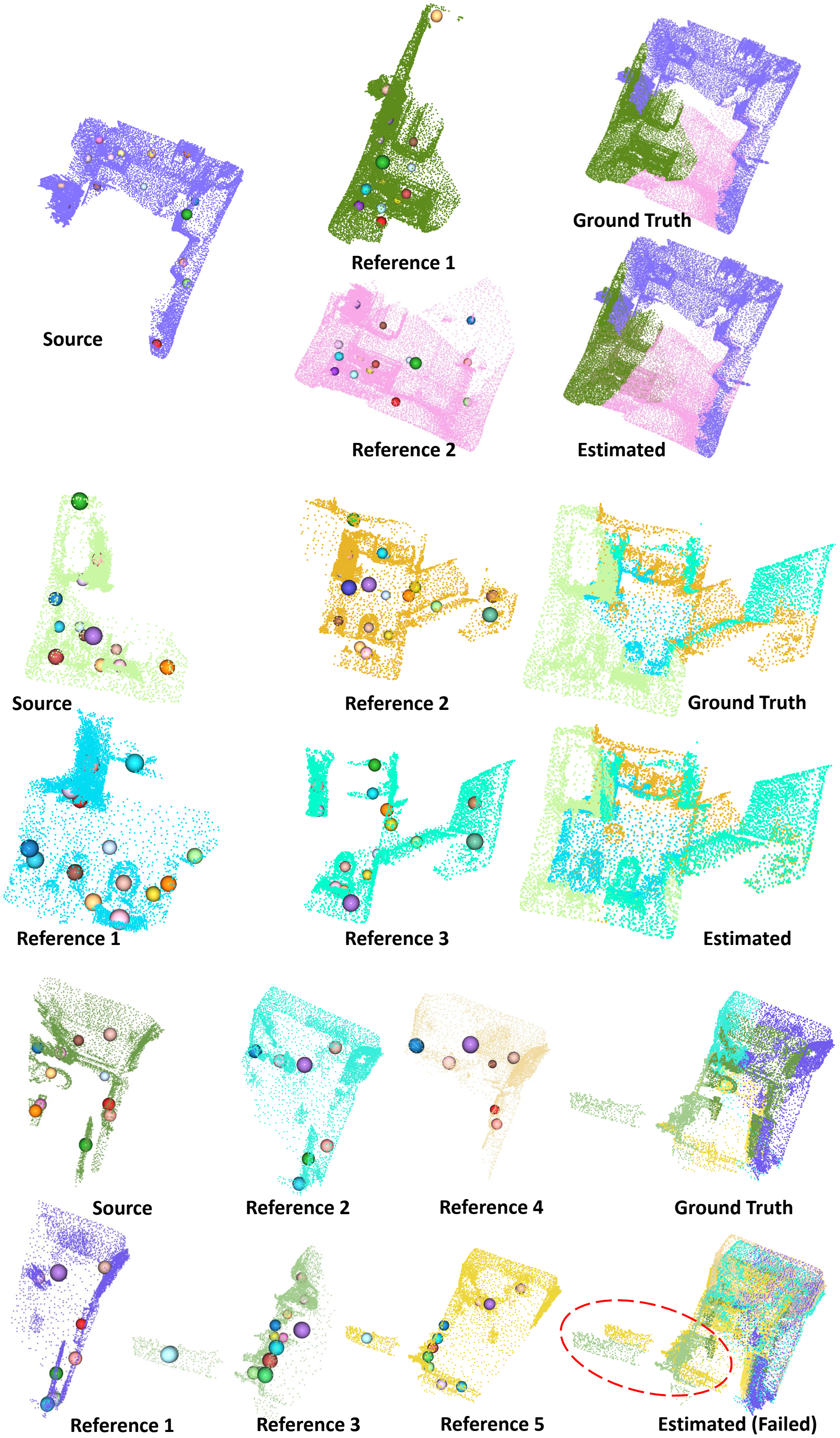


Figure 10. **Qualitative Results on Point Cloud Mosaicking** of our proposed method. Object nodes are visualized as 3D spheres.