

# Learning Transformation Synchronization

Xiangru Huang  
UT Austin

Zhenxiao Liang  
UT Austin

Xiaowei Zhou  
Zhejiang University

Yao Xie  
Georgia Tech

Leonidas Guibas  
Stanford University

Qixing Huang\*  
UT Austin

## Abstract

Reconstructing the 3D model of a physical object typically requires us to align the depth scans obtained from different camera poses into the same coordinate system. Solutions to this global alignment problem usually proceed in two steps. The first step estimates relative transformations between pairs of scans using an off-the-shelf technique. Due to limited information presented between pairs of scans, the resulting relative transformations are generally noisy. The second step then jointly optimizes the relative transformations among all input depth scans. A natural constraint used in this step is the cycle-consistency constraint, which allows us to prune incorrect relative transformations by detecting inconsistent cycles. The performance of such approaches, however, heavily relies on the quality of the input relative transformations. Instead of merely using the relative transformations as the input to perform transformation synchronization, we propose to use a neural network to learn the weights associated with each relative transformation. Our approach alternates between transformation synchronization using weighted relative transformations and predicting new weights of the input relative transformations using a neural network. We demonstrate the usefulness of this approach across a wide range of datasets.

## 1. Introduction

Transformation synchronization, i.e., estimating consistent rigid transformations across a collection of images or depth scans, is a fundamental problem in various computer vision applications, including multi-view structure from motion [5, 30, 39, 37], geometry reconstruction from depth scans [21, 9], image editing via solving jigsaw puzzles [8], simultaneous localization and mapping [4], and re-assembling fractured surfaces [15], to name just a few. A common approach to transformation synchronization proceeds in two phases. The first phase establishes the relative rigid transformations between pairs of objects in iso-

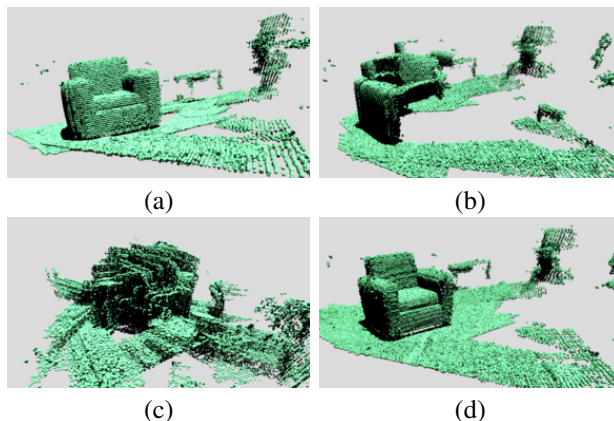


Figure 1: Reconstruction results from 30 RGBD images of an indoor environment using different transformation synchronization methods. (a) Our approach. (b) Rotation Averaging [6]. (c) Geometric Registration[10]. (d) Ground Truth.

lation. Due to incomplete information presented in isolated pairs, the estimated relative transformations are usually quite noisy. The second phase improves the relative transformations by jointly optimizing them across all input objects. This is usually made possible by utilizing the so-called *cycle-consistency* constraint, which states that the composite transformation along every cycle should be the identity transformation, or equivalently, the data matrix that stores pair-wise transformations in blocks is low-rank (c.f. [16]). The cycle-consistency constraint allows us to jointly improve relative transformations by either detecting inconsistent cycles [8] or performing low-rank matrix recovery [16, 38].

However, the success of existing transformation synchronization [38, 5, 2, 19] and more general map synchronization [16, 32, 31, 7, 34, 19] techniques heavily depends on the alignment between the loss function and the noise pattern of the input data. For example, approaches based on robust norms (e.g., L1 [16, 7]) can tolerate either a constant fraction of adversarial noise (c.f.[16, 19]) or a sub-linear outlier ratio when the noise is independent (c.f.[7, 34]).

\*huangqx@cs.utexas.edu

Such assumptions, unfortunately, deviate from many practical settings, where the majority of the input relative transformations may be incorrect (e.g., when the input scans are noisy), and/or the noise pattern in relative transformations is highly correlated (there are a quadratic number of measurements from a linear number of sources). This motivates us to consider the problem of *learning transformation synchronization*, which seeks to learn a suitable loss function that is compatible with the noise pattern of specific datasets.

In this paper, we introduce an approach that formulates transformation synchronization as an end-to-end neural network. Our approach is motivated by reweighted least squares and their application in transformation synchronization (c.f. [5, 2, 10, 19]), where the loss function dictates how we update the weight associated with each input relative transformation during the synchronization process. Specifically, we design a recurrent neural network that reflects this reweighted scheme. By learning the weights from data directly, our approach implicitly captures a suitable loss function for performing transformation synchronization.

We have evaluated the proposed technique on two real datasets: Redwood [11] and ScanNet [12]. Experimental results show that our approach leads to considerable improvements compared to the state-of-the-art transformation synchronization techniques. For example, on Redwood and Scannet, the best combination of existing pairwise matching and transformation synchronization techniques lead to mean angular rotation errors  $22.4^\circ$  and  $64.4^\circ$ , respectively. In contrast, the corresponding statistics of our approach are  $6.9^\circ$  and  $42.9^\circ$ , respectively. We also perform an ablation study to evaluate the effectiveness of our approach.

Code is publicly available at <https://github.com/xiangruhuang/Learning2Sync>.

## 2. Related Works

Existing techniques on transformation synchronization fall into two categories. The first category of methods [21, 15, 40, 29, 43] uses combinatorial optimization to select a subgraph that only contains consistent cycles. The second category of methods [38, 25, 18, 16, 17, 7, 44, 34, 26, 20] can be viewed from the perspective that there is an equivalence between cycle-consistent transformations and the fact that the map collection matrix that stores relative transformations in blocks is semidefinite and/or low-rank (c.f. [16]). These methods formulate transformation synchronization as low-rank matrix recovery, where the input relative transformations are considered noisy measurements of this low-rank matrix. In the literature, people have proposed convex optimization [38, 16, 17, 7], non-convex optimization [5, 44, 26, 20], and spectral techniques [25, 18, 32, 31, 34, 36] for solving various low-rank matrix recovery formulations. Compared with the first category of methods, the second category of methods is computationally more efficient. Moreover, tight exact recovery conditions of many methods have been established.

A message from these exact recovery conditions is that existing methods only work if the fraction of noise in the input relative transformations is below a threshold. The magnitude of this threshold depends on the noise pattern. Existing results either assume adversarial noise [16, 20] or independent random noise [38, 7, 34, 3]. However, as relative transformations are computed between pairs of objects, it follows that these relative transformations are dependent (i.e., between the same source object to different target objects). This means there are a lot of structures in the noise pattern of relative transformations. Our approach addresses this issue by optimizing transformation synchronization techniques to fit the data distribution of a particular dataset. To best of our knowledge, this work is the first to apply supervised learning to the problem of transformation synchronization.

Our approach is also relevant to utilizing recurrent neural networks for solving the pairwise matching problem. Recent examples include learning correspondences between pairs of images [28], predicting the fundamental matrix between two different images of the same underlying environment [33], and computing a dense image flow between an image pair [24]. We study a different problem of transformation synchronization in this paper. In particular, our weighting module leverages problem specific features (e.g., eigen-gap) for determining the weights associated with relative transformations. Learning transformation synchronization also poses great challenges in making the network trainable end-to-end.

## 3. Problem Statement and Approach Overview

In this section, we describe the problem statement of transformation synchronization (Section 3.1) and present an overview of our approach (Section 3.2).

### 3.1. Problem Statement

Consider  $n$  input scans  $\mathcal{S} = \{S_i, 1 \leq i \leq n\}$  capturing the same underlying object/scene from different camera poses. Let  $\Sigma_i$  denote the local coordinate system associated with  $S_i$ . The input to transformation synchronization can be described as a model graph  $\mathcal{G} = (\mathcal{S}, \mathcal{E})$  [22]. Each edge  $(i, j) \in \mathcal{E}$  of the model graph is associated with a relative transformation  $T_{ij}^{in} = (R_{ij}^{in}, \mathbf{t}_{ij}^{in}) \in SE(3)$ , where  $R_{ij}^{in}$  and  $\mathbf{t}_{ij}^{in}$  are rotational and translational components of  $T_{ij}^{in}$ , respectively.  $T_{ij}^{in}$  is usually pre-computed using an off-the-shelf algorithm (e.g., [27, 41]). For simplicity, we impose the assumption that  $(i, j) \in \mathcal{E}$  if and only if (i)  $(j, i) \in \mathcal{E}$ , and (ii) their associated transformations are compatible, i.e.,

$$R_{ji}^{in} = R_{ij}^{inT}, \quad \mathbf{t}_{ji}^{in} = -R_{ij}^{inT} \mathbf{t}_{ij}^{in}.$$

It is expected that many of these relative transformations are incorrect, due to limited information presented between pairs of scans and limitations of the off-the-shelf method being used. The goal of transformation synchronization

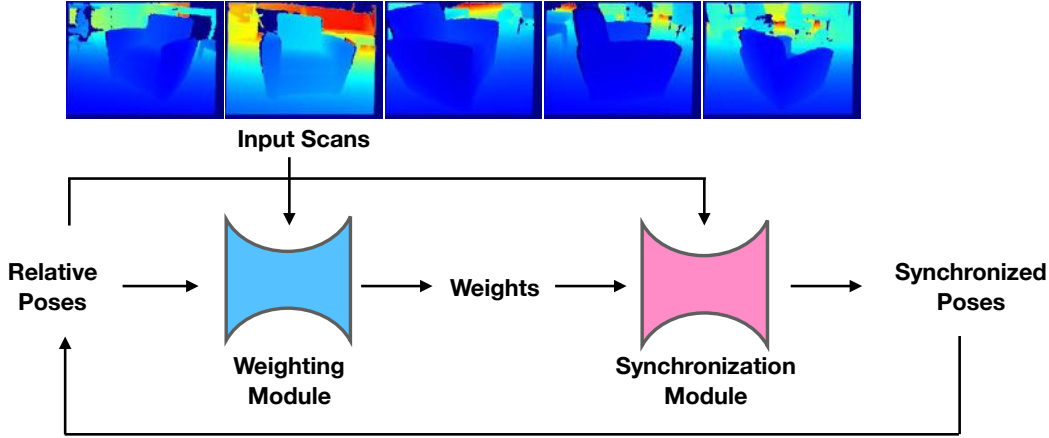


Figure 2: Illustration of our network design.

is to recover the absolute pose  $T_i = (R_i, \mathbf{t}_i) \in SE(3)$  of each scan  $S_i$  in a world coordinate system  $\Sigma$ . Without losing generality, we assume the world coordinate system is given by  $\Sigma := \Sigma_1$ . Note that unlike traditional transformation synchronization approaches that merely use  $T_{ij}^{in}$  (e.g., [5, 38, 2]), our approach also incorporates additional information extracted from the input scans  $S_i, 1 \leq i \leq n$ .

### 3.2. Approach Overview

Our approach is motivated from iteratively reweighted least squares (or IRLS)[13], which has been applied to transformation synchronization (e.g. [5, 2, 10, 19]). The key idea of IRLS is to maintain an edge weight  $w_{ij}, (i, j) \in \mathcal{E}$  for each input transformation  $T_{ij}^{in}$  so that the objective function becomes quadratic in the variables, and transformation synchronization admits a closed-form solution. One can then use the closed-form solution to update the edge weights. Under a special weighting scheme (c.f.[19]), it has been shown that when the fraction of incorrect measurements is below a constant, the weights associated with these incorrect measurements eventually become 0. One way to understand reweighting schemes is that when the weights converged, the reweighted square loss becomes the actual robust loss function that is used to solve the corresponding transformation synchronization problem. In contrast to using a generic weighting scheme, we propose to learn the weighting scheme from data by designing a recurrent network that replicates the reweighted transformation synchronization procedure. By doing so, we implicitly learn a suitable loss function for transformation synchronization.

As illustrated in Figure 2, the proposed recurrent module combines a synchronization layer and a weighting module. At the  $k$ th iteration, the synchronization layer takes as input the initial relative transformations  $T_{ij}^{in} \in SE(3), \forall (i, j) \in \mathcal{E}$  and their associated weights  $w_{ij}^{(k)} \in (0, 1)$  and outputs synchronized poses  $T_i^{(k)} : \Sigma_i \rightarrow \Sigma$  for the input objects

$S_i, 1 \leq i \leq n$ . Initially, we set  $w_{ij}^{(1)} = 1, \forall (i, j) \in \mathcal{E}$ . The technical details of the synchronization layer are described in Section 4.1.

The weighting module operates on each object pair in isolation. For each edge  $(i, j) \in \mathcal{E}$ , the input to the proposed weighting module consists of (1) the input relative transformation  $T_{ij}^{in}$ , (2) the induced relative transformation at the  $k$ th iteration

$$\hat{T}_{ij}^{(k)} := T_j^{(k)} T_i^{(k)-1},$$

(3) features extracted from the initial alignment of the two input scans, and (4) a status vector  $\mathbf{v}^{(k)}$  that collects global signals from the synchronization layer at the  $k$ th iteration (e.g., spectral gap). The output is the associated weight  $w_{ij}^{(k+1)}$  at the  $k+1$ th iteration.

The network is trained end-to-end by penalizing the differences between the ground-truth poses and the output of the last synchronization layer. The technical details of this end-to-end training procedure are described in Section 4.3.

## 4. Approach

In this section, we introduce the technical details of our learning transformation synchronization approach. In Section 4.1, we introduce details of the synchronization layer. In Section 4.2, we describe the weighting module. Finally, we show how to train the proposed network end-to-end in Section 4.3. Note that the proofs of the propositions introduced in this section are deferred to the supplemental material.

### 4.1. Synchronization Layer

For simplicity, we ignore the superscripts  $k$  and  $in$  when introducing the synchronization layer. Let  $T_{ij} = (R_{ij}, \mathbf{t}_{ij})$  and  $w_{ij}$  be the input relative transformation and its weights associated with the edge  $(i, j) \in \mathcal{E}$ . We assume that this

---

**Algorithm 1** Translation Synchronization Layer.

---

**function** SYNC( $(w_{ij}, T_{ij}), \forall (i, j) \in \mathcal{E}$ )  
Form the connection Laplacian  $L$  and vector  $\mathbf{b}$ ;  
Compute first 3 eigenvectors  $U$  of  $L$ ;  
Perform SVD on blocks of  $U$  to obtain  $\{R_i^*, 1 \leq i \leq n\}$  via (3);  
Solve (6) to obtain  $\{\mathbf{t}_i^*, 1 \leq i \leq n\}$ ;  
**return**  $T_i^* = (R_i^*, \mathbf{t}_i^*), 1 \leq i \leq n$ ;  
**end function**

---

weighted graph is connected. The goal of the synchronization layer is to compute the synchronized pose  $T_i^* = (R_i^*, \mathbf{t}_i^*)$  associated with each scan  $S_i$ . Note that a correct relative transformation  $T_{ij} = (R_{ij}, \mathbf{t}_{ij})$  induces two separate constraints on the rotations  $R_i^*$  and translations  $\mathbf{t}_i^*$ , respectively:

$$R_{ij}R_i^* = R_j^*, \quad R_{ij}\mathbf{t}_i^* + \mathbf{t}_{ij} = \mathbf{t}_j^*.$$

We thus perform rotation synchronization and translation synchronization separately.

Our rotation synchronization approach adapts the spectral rotation synchronization approach described in [1]. Specifically, we consider the following optimization problem for rotation synchronization:

$$\underset{R_i \in SO(3), 1 \leq i \leq n}{\text{minimize}} \sum_{(i,j) \in \mathcal{E}} w_{ij} \|R_{ij}R_i - R_j\|_{\mathcal{F}}^2 \quad (1)$$

Solving (1) exactly is difficult. We propose to first relax the constraint  $R_i \in SO(3), 1 \leq i \leq n$  to  $\sum_{i=1}^n R_i R_i^T = nI_3$  when solving (1) and then project each of the resulting solution  $R_i$  to  $SO(3)$ . This leads to the following procedure for rotation synchronization. More precisely, we introduce a connection Laplacian  $L \in \mathbb{R}^{3n \times 3n}$  [35], whose blocks are given by

$$L_{ij} := \begin{cases} \sum_{j \in \mathcal{N}(i)} w_{ij} I_3 & i = j \\ -w_{ij} R_{ij}^T & (i, j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where  $\mathcal{N}(i)$  collects all neighbor vertices of  $i$  in  $\mathcal{G}$ .

Let  $U = (U_1^T, \dots, U_n^T)^T \in \mathbb{R}^{3n \times 3}$  collect the eigenvectors of  $L$  that correspond to the three smallest eigenvalues. We choose the sign of each eigenvector such that  $\sum_{i=1}^n \det(U_i) > 0$ . To compute the absolute rotations, we first perform singular value decomposition (SVD) on each

$$U_i = V_i \Sigma_i W_i^T.$$

We then output the corresponding absolute rotation estimate as

$$R_i^* = V_i W_i^T \quad (3)$$

The following proposition states that although  $\{R_i^*, 1 \leq i \leq n\}$  do not exactly optimize (1), they still provide effective synchronized rotations due to the following robust recovery property:

**Proposition 1. (Informal)** Consider the ground-truth rotations  $R_i^{gt}, 1 \leq i \leq n$ . Suppose  $\forall (i, j) \in \mathcal{E}$  where  $w_{ij} > 0$ ,

$$R_{ij}R_i^{gt} \approx R_j^{gt}. \quad (4)$$

Then  $R_i^*, 1 \leq i \leq n$  in (3) approximately recovers the ground-truth rotations  $R_i^{gt}, 1 \leq i \leq n$ . More precisely, we define

$$\Delta R_{ij} = R_{ij} - R_j^{gt} R_i^{gtT}$$

as the estimation error on  $R_{ij}$ . With  $\Delta R = \{\Delta R_{ij}\}$  we denote the corresponding error matrix. When the constraints in (4) are exact, or equivalently,  $\Delta R = 0$ , then the recovery is also exact. In this case, we have

$$0 = \lambda_1(L) = \lambda_2(L) = \lambda_3(L) < \lambda_4(L) \leq \dots \leq \lambda_{3n}(L).$$

In other words, if the weighting module sets weights of outlier relative transformations to 0, then  $R_i^*$  approximately recover the underlying rotations.

**Translation synchronization** solves the following least square problem to obtain  $\mathbf{t}_i$ :

$$\underset{\mathbf{t}_i, 1 \leq i \leq n}{\text{minimize}} \sum_{(i,j) \in \mathcal{E}} w_{ij} \|R_{ij}\mathbf{t}_i + \mathbf{t}_{ij} - \mathbf{t}_j\|^2 \quad (5)$$

Let  $\mathbf{t} = (\mathbf{t}_1^T, \dots, \mathbf{t}_n^T)^T \in \mathbb{R}^{3n}$  collect the translation components of the synchronized poses in a column vector. Introduce a column vector  $\mathbf{b} = (\mathbf{b}_1^T, \dots, \mathbf{b}_n^T)^T \in \mathbb{R}^{3n}$  where

$$\mathbf{b}_i := - \sum_{j \in \mathcal{N}(i)} w_{ij} R_{ij}^T \mathbf{t}_{ij}.$$

Then an<sup>1</sup> optimal solution  $\mathbf{t}^*$  to (5) is given by

$$\mathbf{t}^* = L^+ \mathbf{b}. \quad (6)$$

Similar to the case of rotation synchronization, we have the following robust recovery property:

**Proposition 2. (Informal)** Consider the underlying synchronized poses  $T_i^{gt} = (R_i^{gt}, \mathbf{t}_i^{gt}), 1 \leq i \leq n$ . Denote  $\mathbf{t}^{gt} = (\mathbf{t}_1^{gtT}, \dots, \mathbf{t}_n^{gtT})^T$ . Suppose  $\forall (i, j) \in \mathcal{E}$  where  $w_{ij} > 0$ ,

$$R_{ij}R_i^{gt} \approx R_j^{gt}, \quad R_{ij}\mathbf{t}_i^{gt} + \mathbf{t}_{ij} \approx \mathbf{t}_j^{gt}. \quad (7)$$

Then  $\mathbf{t}^*$  in (6) approximately recovers  $\mathbf{t}^{gt}$ . In particular, when the constraints in (7) are exact, then the recovery is also exact.

Similar to the case of rotation synchronization, if the pairwise matching module sets the weights of outlier relative transformations to 0, then  $\mathbf{t}_i^*$  approximately recover the underlying translations.

---

<sup>1</sup>When  $L$  is positive semidefinite, then the solution is unique, and (6) gives one optimal solution.

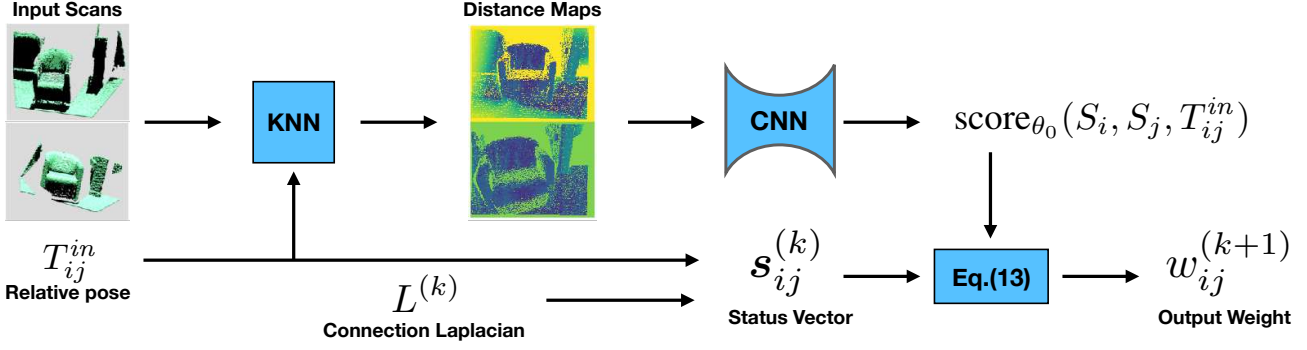


Figure 3: Illustration of network design of the weighting module. We first compute k-nearest neighbor distances between a pair of depth images, which form the images (shown as heat maps) in the middle. We then apply a classic convolutional neural network to output a score between (0, 1), which is then combined with status vector to produce the weight of this relative pose according to (13).

## 4.2. Weighting Module

We define the weighting module as the following function:

$$w_{ij}^{(k+1)} \leftarrow \text{Weight}_{\theta}(S_i, S_j, T_{ij}^{in}, \mathbf{s}_{ij}^{(k)}) \quad (8)$$

where the input consists of (i) a pair of scans  $S_i$  and  $S_j$ , (ii) the input relative transformation  $T_{ij}^{in}$  between them, and (iii) a status vector  $\mathbf{s}_{ij}^{(k)} \in \mathbb{R}^4$ . The output of this weighting module is given by the new weight  $w_{ij}^{(k+1)}$  at the  $k+1$ th iteration. With  $\theta$  we denote the trainable weights of the weighting module. In the following, we first introduce the definition of the status vector  $\mathbf{s}_{ij}^{(k)}$ .

**Status vector.** The purpose of the status vector  $\mathbf{s}_{ij}^{(k)}$  is to collect additional signals that are useful for determining the output of the weighting module. Define

$$s_{ij1}^{(k)} := \|R_{ij}^{in} - R_j^{(k)} R_i^{(k)T}\|_{\mathcal{F}}, \quad (9)$$

$$s_{ij2}^{(k)} := \|R_{ij}^{in} \mathbf{t}_i^{(k)} + \mathbf{t}_{ij}^{in} - \mathbf{t}_j^{(k)}\|. \quad (10)$$

$$s_{ij3}^{(k)} := \lambda_4(L^{(k)}) - \lambda_3(L^{(k)}), \quad (11)$$

$$s_{ij4}^{(k)} := \sum_{(i,j) \in \mathcal{E}} w_{ij}^{(k)} \|\mathbf{t}_{ij}^{(k)}\|^2 - \mathbf{b}^{(k)T} L^{(k)+} \mathbf{b}^{(k)}, \quad (12)$$

Essentially,  $s_{ij1}^{(k)}$  and  $s_{ij2}^{(k)}$  characterize the difference between current synchronized transformations and the input relative transformations. The motivation for using them comes from the fact that for a standard reweighted scheme for transformation synchronization (c.f. [19]), one simply sets  $w_{ij}^{(k+1)} = \rho(s_{ij1}^{(k)}, s_{ij2}^{(k)})$  for a weighting function  $\rho$  (c.f. [13]). This scheme can already recover the underlying ground-truth in the presence of a constant fraction of adversarial incorrect relative transformations (Please refer to the supplemental material for a rigorous analysis). In contrast, our approach seeks to go beyond this limit by leveraging additional information. The definition of  $s_{ij3}^{(k)}$  comes from Prop.1.  $s_{ij4}^{(k)}$  equals to the residual of (5). Intuitively, when

both  $s_{ij3}^{(k)}$  and  $s_{ij4}^{(k)}$  are small, the weighted relative transformations  $(w_{ij}^{(k)}, T_{ij}^{in})$  will be consistent, from which we can recover accurate synchronized transformations  $T_i^{(k)}$ . We now describe the network design.

**Network design.** As shown in Figure 3, the key component of our network design is a sub-network  $\text{score}_{\theta_0}(S_i, S_j, T_{ij}^{in})$  that takes two scans  $S_i$  and  $S_j$  and a relative transformation  $T_{ij}^{in}$  between them and output a score in  $[0, 1]$  that indicates whether this is a good scan alignment or not, i.e., 1 means a good alignment, and 0 means an incorrect alignment.

We design  $\text{score}_{\theta_0}$  as a feed-forward network. Its input consists of two color maps that characterize the alignment patterns between the two input scans. The value of each pixel represents the distance of the corresponding 3D point to the closest points on the other scan under  $T_{ij}^{in}$  (See the second column of images in Figure 3). We then concatenate these two color images and feed them into a neural network (we used a modified AlexNet architecture), which outputs the final score.

With this setup, we define the output weight  $w_{ij}^{(k+1)}$  as

$$w_{ij}^{(k+1)} := \frac{e^{\theta_1 \theta_2}}{e^{\theta_1 \theta_2} + (\text{score}_{\theta_0}(S_i, S_j, T_{ij}^{in}) \mathbf{s}_{ij}^{(k)T} \theta_3)^{\theta_2}} \quad (13)$$

Here  $\theta_1, \theta_2 \in \mathbb{R}$  adopts form of traditional reweighting function and  $\theta_3 \in \mathbb{R}^4$  encode the importance of the elements of  $\mathbf{s}_{ij}^{(k)}$ . With  $\theta = (\theta_0, \theta_1, \theta_2, \theta_3)$  we collect all trainable parameters of (13).

## 4.3. End-to-End Training

Let  $\mathcal{D}$  denote a dataset of scan collections with annotated ground-truth poses. Let  $k_{\max}$  be the number of recurrent steps (we used four recurrent steps in our experiments). We define the following loss function for training the weighting

module  $\text{Weight}_\theta$ :

$$\min_{\theta} \sum_{S \in \mathcal{D}} \sum_{1 \leq i < j \leq |S|} \left( \|R_j^{k_{\max}} R_i^{k_{\max} T} - R_j^{gt} R_i^{gt T}\|_{\mathcal{F}}^2 + \lambda \|t_i^{k_{\max}} - t_i^{gt}\|^2 \right) \quad (14)$$

where we set  $\lambda = 10$  in all of our experiments. Note that we compare relative rotations to factor out the global orientation among the poses. The global shift in translation is already handled by (6).

We perform back-propagation to optimize (14). The technical challenges are to compute the derivatives that pass through the synchronization layer, including 1) the derivatives of  $R_j^* R_i^{* T}$  with respect to the elements of  $L$ , 2) the derivatives of  $t_i^*$  with respect to the elements of  $L$  and  $\mathbf{b}$ , and 3) the derivatives of each status vector with respect to the elements of  $L$  and  $\mathbf{b}$ . In the following, we provide explicit expressions for computing these derivatives.

We first present the derivative between the output of rotation synchronization and its input. To make the notation uncluttered, we compute the derivative by treating  $L$  is a matrix function. The derivative with respect to  $w_{ij}$  can be easily obtained via chain-rule.

**Proposition 3.** *Consider the setup of Prop. 1. Let  $\mathbf{u}_i$  and  $\lambda_i$  be the  $i$ -th eigenvector and eigenvalue of  $L$ . Expand the SVD of  $U_i = V_i \Sigma_i W_i^T$  as follows:*

$$V_i = (\mathbf{v}_{i,1}, \mathbf{v}_{i,2}, \mathbf{v}_{i,3}), \quad \Sigma_i = \text{diag}(\sigma_{i,1}, \sigma_{i,2}, \sigma_{i,3}), \\ W_i = (\mathbf{w}_{i,1}, \mathbf{w}_{i,2}, \mathbf{w}_{i,3}).$$

Let  $\mathbf{e}_j^t \in \mathbb{R}^t$  be the  $j$ th canonical basis of  $\mathbb{R}^t$ . We then have

$$d(R_j^* R_i^{* T}) = dR_j \cdot R_i^{* T} + R_j^* \cdot dR_i^T,$$

where

$$dR_i := \sum_{1 \leq s, t \leq 3} \frac{\mathbf{v}_{i,s}^T dU_i \mathbf{w}_{i,t} - \mathbf{v}_{i,t}^T dU_i \mathbf{w}_{i,s}}{\sigma_{i,s} + \sigma_{i,t}} \mathbf{v}_{i,s} \mathbf{w}_{i,t}^T,$$

where  $dU_i$  is defined by  $\forall 1 \leq j \leq 3$ ,

$$dU_i \mathbf{e}_j^{(3)} = (\mathbf{e}_i^{(n)T} \otimes I_3) \sum_{l=4}^{3n} \frac{\mathbf{u}_l \mathbf{u}_l^T}{\lambda_j(L) - \lambda_l(L)} dL \mathbf{u}_j.$$

The following proposition specifies the derivative of  $t^*$  with respect to the elements of  $L$  and  $\mathbf{b}$ :

**Proposition 4.** *The derivatives of  $t^*$  are given by*

$$dt^* = L^+ dLL^+ + L^+ d\mathbf{b}.$$

Regarding the status vectors, the derivatives of  $s_{ij,1}$  with respect to the elements of  $L$  are given by Prop. 3; The derivatives of  $s_{ij,2}$  and  $s_{ij,4}$  with respect to the elements of  $L$  are given by Prop. 4. It remains to compute the derivatives of  $s_{ij,3}$  with respect to the elements of  $L$ , which can be easily obtained via the derivatives of the eigenvalues of  $L$  [23], i.e.,  $d\lambda_i = \mathbf{u}_i^T dL \mathbf{u}_i$ .

## 5. Experimental Results

This section presents an experimental evaluation of the proposed learning transformation synchronization approach. We begin with describing the experimental setup in Section 5.1. In Section 5.2, we analyze the results of our approach and compare it against baseline approaches. Finally, we present an ablation study in Section 5.3.

### 5.1. Experimental Setup

**Datasets.** We consider two datasets in this paper, Redwood [11] and ScanNet [12]:

- **Redwood** contains RGBD sequences of individual objects. We uniformly sample 60 sequences. For each sequence, we sample 30 RGBD images that are 20 frames away from the next one, which cover 600 frames of the original sequence. For experimental evaluation, we use the poses associated with the reconstruction as the ground-truth. We use 35 sequences for training and 25 sequences for testing. Note that the temporal order among the frames in each sequence is discarded in our experiments.
- **ScanNet** contains RGBD sequences, as well as reconstruction, camera pose, for 706 indoor scenes. Each scene contains 2-3 sequences of different trajectories. We randomly sample 100 sequences from ScanNet. We use 70 sequences for training and 30 sequences for testing. Again the temporal order among the frames in each sequence is discarded in our experiments.

More details about the sampled sequences are given in the appendix.

**Pairwise methods.** We consider two state-of-the-art pairwise methods for generating the input to our approach:

- **Super4PCS** [27] applies sampling to find consistent matches of four point pairs.
- **FastGR** [41] utilizes feature correspondences and applies reweighted non-linear least squares to extract a set of consistent feature correspondences and fit a rigid pose. We used the Open3D implementation [42].

**Baseline approaches.** We consider the following baseline approaches that are introduced in the literature for transformation synchronization:

- **Robust Relative Rotation Averaging (RotAvg)**[6] is a scalable algorithm that performs robust rotation averaging of relative rotations. To recover translations, we additionally apply a state-of-the-art translation synchronization approach [19]. We use default setting of its publicly accessible code. [19] is based on our own Matlab implementation.
- **Geometric Registration** [10] solve multi-way registration via pose graph optimization. We modify

Methods	Redwood											ScanNet												
	Rotation Error						Translation Error					Rotation Error						Translation Error						
	3°	5°	10°	30°	45°	Mean	0.05	0.1	0.25	0.5	0.75	Mean	3°	5°	10°	30°	45°	Mean	0.05	0.1	0.25	0.5	0.75	Mean
FastGR (all)	29.4	40.2	52.0	63.8	70.4	37.4°	22.0	39.6	53.0	60.3	67.0	0.68	9.9	16.8	23.5	31.9	38.4	76.3°	5.5	13.3	22.0	29.0	36.3	1.67
FastGR (good)	33.9	45.2	57.2	67.4	73.2	34.1°	26.7	45.7	58.8	65.9	71.4	0.59	12.4	21.4	29.5	38.6	45.1	68.8°	7.7	17.6	28.2	36.2	43.4	1.43
Super4PCS (all)	6.9	10.1	16.7	39.6	52.3	55.8°	4.2	8.9	18.2	31.0	43.5	1.14	0.5	1.3	4.0	17.4	25.2	98.5°	0.3	1.2	5.3	13.3	21.6	2.11
Super4PCS (good)	10.3	14.9	23.9	48.0	60.0	49.2°	6.4	13.3	26.2	41.2	53.2	0.93	0.8	2.3	6.4	23.0	31.7	90.8°	0.6	2.2	8.9	19.5	29.5	1.80
RotAvg (FastGR)	30.4	42.6	59.4	74.4	82.1	22.4°	23.3	43.2	61.8	72.4	80.7	0.42	6.0	10.4	17.3	36.1	46.1	64.4°	3.7	9.2	19.5	34.0	45.6	1.26
multi-FastGR (FastGR)	17.8	28.7	47.5	74.2	83.2	27.7°	4.9	18.4	50.2	72.6	81.4	0.93	0.2	0.6	2.8	16.4	27.1	87.2°	0.1	0.7	4.8	16.4	28.4	1.80
RotAvg (Super4PCS)	5.4	8.7	17.4	45.1	59.2	49.6°	3.2	7.4	17.0	32.3	46.3	0.95	0.3	0.8	3.0	15.4	23.3	96.8°	0.2	1.0	5.8	16.5	27.6	1.70
multi-FastGR (Super4PCS)	2.1	4.1	10.2	33.1	48.3	60.6°	1.1	3.1	10.3	21.5	31.8	1.25	1.9	5.1	13.9	36.6	47.1	72.9°	0.4	2.1	9.8	23.2	34.5	1.82
Our Approach (FastGR)	<b>67.5</b>	<b>77.5</b>	<b>85.6</b>	<b>91.7</b>	<b>94.4</b>	<b>6.9°</b>	20.7	40.0	70.9	<b>88.6</b>	<b>94.0</b>	0.26	<b>34.4</b>	<b>41.1</b>	<b>49.0</b>	<b>58.9</b>	<b>62.3</b>	<b>42.9°</b>	2.0	7.3	22.3	36.9	48.1	1.16
Our Approach (Super4PCS)	2.3	5.1	13.2	42.5	60.9	46.7°	1.1	4.0	13.8	29.0	42.3	1.02	0.4	1.7	6.8	29.6	43.5	66.9°	0.1	0.8	5.6	16.6	27.0	1.90
Transf. Sync. (FastGR)	27.1	37.7	56.9	74.4	82.4	22.1°	17.4	34.4	55.9	70.4	81.3	0.43	3.2	6.5	14.6	35.8	47.4	63.5°	1.6	5.6	15.5	30.9	43.4	1.31
Input Only (FastGR)	36.7	51.4	68.1	87.7	91.7	13.7°	25.1	<b>49.3</b>	73.2	86.4	91.6	0.26	11.7	19.4	30.5	50.7	57.7	51.7°	<b>5.9</b>	<b>15.4</b>	<b>30.5</b>	43.7	52.2	1.03
No Recurrent (FastGR)	37.8	52.8	71.1	87.7	91.7	12.9°	<b>26.3</b>	51.1	<b>77.3</b>	87.1	92.0	<b>0.24</b>	8.6	15.3	26.9	51.4	58.2	49.8°	3.9	11.1	27.3	<b>43.7</b>	<b>53.9</b>	<b>1.01</b>

Figure 4: Benchmark evaluations on Redwood [11] and ScanNet [12]. Quality of absolute poses are evaluated by computing errors to pairwise ground truth poses. Angular distances between rotation matrices are computed via  $\text{angular}(R_{ij}, R_{ij}^*) = \arccos(\frac{\text{tr}(R_{ij}^T R_{ij}^*) - 1}{2})$ . Translation distances are computed by  $\|t_{ij} - t_{ij}^*\|$ . We collect statistics on percentages of rotation and translation errors that are below a varying threshold. I) The 4th to 7th rows contain evaluations for upstream algorithms. (all) refers to statistics among all pairs where (good) refers to the statistics computed among relative poses with good quality overlap regions with respect to our pretrained weighting module. II) For the second part, we report results of all baselines computed from this good set of relative poses, which is consistently better than the results from all relative poses. Since there are two input methods, we report the results of each transformation synchronization approach on both inputs. III) The third parts contain results for ablation study performed only on FastGR[41] inputs. The first row reports state-of-the-art rotation and translation synchronization results, followed by variants of our approach.

the Open3D implementation to take inputs from Super4PCS or FastGR.

Note that our approach utilizes a weighting module to score the input relative transformations. To make fair comparisons, we apply our pretrained weighting module to filter all input transformations, whose associated scores are below 0.1. We then feed these filtered input transformations to each baseline approach for experimental evaluation.

**Evaluation protocol.** We employ the evaluation protocols of [5] and [19] for evaluating rotation synchronization and translation synchronization, respectively. Specifically, for rotations, we first solve the best matching global rotation between the ground-truth and the prediction, we then report the cumulative distribution function (or CDF) of angular deviation  $\arccos(\frac{\|\log(R^T R^{gt})\|_{\mathcal{F}}}{\sqrt{2}})$  between a prediction  $R$  and its corresponding ground-truth  $R^{gt}$ . For translations, we report the CDF of  $\|t - t^{gt}\|$  between each pair of prediction  $t$  and its corresponding ground-truth  $t^{gt}$ .

## 5.2. Analysis of Results

Figure 4 and Figure 5 present quantitative and qualitative results, respectively. Overall, our approach yielded fairly accurate results. On Redwood, the mean errors in rotations/translations of FastGR and our result from FastGR are  $34.1^\circ/0.58m$  and  $6.9^\circ/0.26m$ , respectively. On ScanNet, the mean errors in rotations/translations of FastGR and our result from FastGR are  $68.8^\circ/1.43m$  and  $42.9^\circ/1.16m$ , respectively. Note that in both cases, our approach leads to

salient improvements from the input. The final results of our approach on ScanNet are less accurate than those on Redwood. Besides the fact that the quality of the initial relative transformations is lower on ScanNet than that on Redwood, another factor is that the depth scans from ScanNet are quite noisy, leading to noisy input (and thus less signals) for the weighting module. Still, the improvements of our approach on ScanNet are salient.

Our approach still requires reasonable initial transformations to begin with. This can be understood from the fact that our approach seeks to perform synchronization by selecting a subset of input relative transformations. Although our approach utilizes learning, its performance shall decrease when the quality of the initial relative transformations drops. An evidence is that our approach only leads to modest performance gains when taking the output of Super4PCS as input.

**Comparison with state-of-the-art approaches.** Although all the two baseline approaches improve from the input relative transformations, our approach exhibits significant further improvements from all baseline approaches. On Redwood, the mean rotation and translation errors of the top performing method RotAvg from FastGR are  $22.4^\circ$  and  $0.418m$ , respectively. The reductions in mean error of our approach are 69.2% and 39.0% for rotations and translations, respectively, which are significant. The reductions in mean errors of our approach on ScanNet are also noticeable, i.e., 33.3% and 7.4% in rotations and translations, respectively.

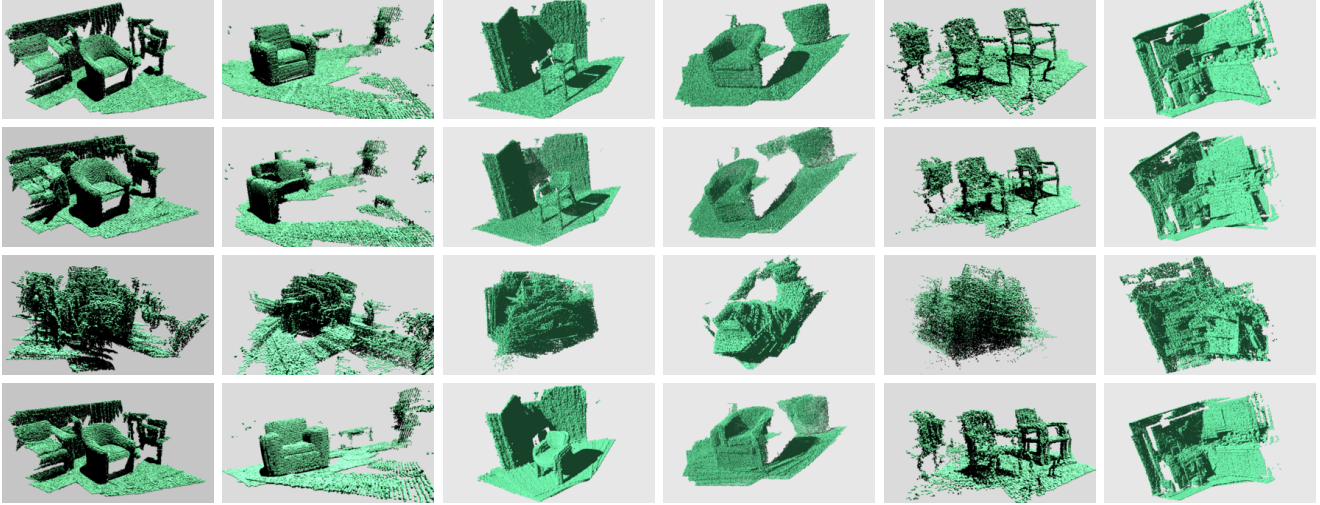


Figure 5: For each block, we show the results of our approach (row I), Rotation Averaging [6]+Translation Sync. [19] (row II), Geometric Registration [10] (row III), and Ground Truth (row IV) (Top). The left four scenes are from Redwood[11] and the right two scenes are from ScanNet[12]

Our approach also achieved relative performance gains from baseline approaches when taking the output of Super4PCS as input. In particular, for mean rotation errors, our approach leads to reductions of 5% and 9% on Redwood and ScanNet, respectively.

When comparing rotations and translations, the improvements on mean rotation errors are bigger than those on mean translation errors. One explanation is that there are a lot of planar structures in Redwood and ScanNet. When aligning such planar structures, rotation errors easily lead to a large change in nearest neighbor distances and thus can be detected by our weighting module. In contrast, translation errors suffer from the gliding effects on planar structures (c.f.[14]), and our weighting module becomes less effective.

### 5.3. Ablation Study

In this section, we present two variants of our learning transformation synchronization approach to justify the usefulness of each component of our system. Due to space constraint, we perform ablation study only using FastGR.

**Input only.** In the first experiment, we simply learn to classify the input maps, and then apply transformation synchronization techniques on the filtered input transformations. In this setting, state-of-the-art transformation synchronization techniques achieves mean rotation/translation errors of  $22.1^\circ/0.43m$  and  $63.5^\circ/1.25m$  on Redwood and ScanNet, respectively. By applying our learning approach to fixed initial map weights, e.g., we fix  $\theta_0$  of the weighting module during, our approach reduced the mean errors to  $13.7^\circ/0.255m$  and  $51.7^\circ/1.031m$  on Redwood and ScanNet, respectively. Although the improvements are noticeable, there are still gaps between this reduced approach and our full approach. This justifies the importance of learning the weighting module together.

**No recurrent module.** Another reduced approach is to directly combine the weighting module and one synchronization layer. Although this approach can improve from the input transformations. There is still a big gap between this approach and our full approach (See the last row in Figure 4). This shows the importance of using weighting modules to gradually reduce the error while simultaneously make the entire procedure trainable end-to-end.

## 6. Conclusions

In this paper, we have introduced a supervised transformation synchronization approach. It modifies a reweighted non-linear least square approach and applies a neural network to automatically determine the input pairwise transformations and the associated weights. We have shown how to train the resulting recurrent neural network end-to-end. Experimental results show that our approach is superior to state-of-the-art transformation synchronization techniques on ScanNet and Redwood for two state-of-the-art pairwise scan matching methods.

There are ample opportunities for future research. So far we have only considered classifying pairwise transformations, it would be interesting to study how to classifying high-order matches. Another interesting direction is to install ICP alignment into our recurrent procedure, i.e., we start from the current synchronized poses and perform ICP between pairs of scans to obtain more signals for transformation synchronization. Moreover, instead of maintaining one synchronized pose per scan, we can maintain multiple synchronized poses, which offer more pairwise matches between pairs of scans for evaluation. Finally, we would like to apply our approach to synchronize dense correspondences across multiple images/shapes.

**Acknowledgement:** The authors wish to thank the sup-



port of NSF grants DMS-1546206, DMS-1700234, CHS-1528025, a DoD Vannevar Bush Faculty Fellowship, a Google focused research award, a gift from adobe research, a gift from snap research, a hardware donation from NVIDIA, and an Amazon AWS AI Research gift.

## References

- [1] M. Arie-Nachimson, S. Z. Kovalsky, I. Kemelmacher-Shlizerman, A. Singer, and R. Basri. Global motion estimation from point matches. In *Proceedings of the 2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission, 3DIMPVT '12*, pages 81–88, Washington, DC, USA, 2012. IEEE Computer Society.
- [2] F. Arrigoni, A. Fusiello, B. Rossi, and P. Fragneto. Robust rotation synchronization via low-rank and sparse matrix decomposition. *CoRR*, abs/1505.06079, 2015.
- [3] C. Bajaj, T. Gao, Z. He, Q. Huang, and Z. Liang. SMAC: simultaneous mapping and clustering using spectral decompositions. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholm, Sweden, July 10-15, 2018*, pages 334–343, 2018.
- [4] L. Carlone, R. Tron, K. Daniilidis, and F. Dellaert. Initialization techniques for 3d SLAM: A survey on rotation estimation and its use in pose graph optimization. In *ICRA*, pages 4597–4604. IEEE, 2015.
- [5] A. Chatterjee and V. M. Govindu. Efficient and robust large-scale rotation averaging. In *ICCV*, pages 521–528. IEEE Computer Society, 2013.
- [6] A. Chatterjee and V. M. Govindu. Robust relative rotation averaging. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):958–972, 2018.
- [7] Y. Chen, L. J. Guibas, and Q. Huang. Near-optimal joint object matching via convex relaxation. In *ICML*, pages 100–108, 2014.
- [8] T. S. Cho, S. Avidan, and W. T. Freeman. The patch transform. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(8):1489–1501, 2010.
- [9] S. Choi, Q.-Y. Zhou, and V. Koltun. Robust reconstruction of indoor scenes. In *CVPR*, pages 5556–5565. IEEE Computer Society, 2015.
- [10] S. Choi, Q.-Y. Zhou, and V. Koltun. Robust reconstruction of indoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [11] S. Choi, Q.-Y. Zhou, S. Miller, and V. Koltun. A large dataset of object scans. *arXiv:1602.02481*, 2016.
- [12] A. Dai, A. X. Chang, M. Savva, M. Halber, T. A. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. *CoRR*, abs/1702.04405, 2017.
- [13] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk. Iteratively re-weighted least squares minimization for sparse recovery. Report, Program in Applied and Computational Mathematics, Princeton University, Princeton, NJ, USA, June 2008.
- [14] N. Gelfand, S. Rusinkiewicz, L. Ikemoto, and M. Levoy. Geometrically stable sampling for the ICP algorithm. In *3DIM*, pages 260–267. IEEE Computer Society, 2003.
- [15] Q. Huang, S. Flöry, N. Gelfand, M. Hofer, and H. Pottmann. Reassembling fractured objects by geometric matching. *ACM Trans. Graph.*, 25(3):569–578, July 2006.
- [16] Q. Huang and L. Guibas. Consistent shape maps via semidefinite programming. In *Proceedings of the Eleventh Eurographics/ACMSIGGRAPH Symposium on Geometry Processing, SGP '13*, pages 177–186, Aire-la-Ville, Switzerland, Switzerland, 2013. Eurographics Association.
- [17] Q. Huang, F. Wang, and L. J. Guibas. Functional map networks for analyzing and exploring large shape collections. *ACM Trans. Graph.*, 33(4):36:1–36:11, 2014.
- [18] Q. Huang, G. Zhang, L. Gao, S. Hu, A. Butscher, and L. J. Guibas. An optimization approach for extracting and encoding consistent maps in a shape collection. *ACM Trans. Graph.*, 31(6):167:1–167:11, 2012.
- [19] X. Huang, Z. Liang, C. Bajaj, and Q. Huang. Translation synchronization via truncated least squares. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 1459–1468. Curran Associates, Inc., 2017.
- [20] X. Huang, Z. Liang, C. Bajaj, and Q. Huang. Translation synchronization via truncated least squares. In *NIPS*, 2017.
- [21] D. Huber. *Automatic Three-dimensional Modeling from Reality*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, December 2002.
- [22] D. F. Huber and M. Hebert. Fully automatic registration of multiple 3d data sets. *Image and Vision Computing*, 21:637–650, 2001.
- [23] M. K. Kadalbajoo and A. Gupta. An overview on the eigenvalue computation for matrices. *Neural, Parallel Sci. Comput.*, 19(1-2):129–164, Mar. 2011.
- [24] S. Kim, S. Lin, S. R. JEON, D. Min, and K. Sohn. Recurrent transformer networks for semantic correspondence. In *NIPS*, page to appear, 2018.
- [25] V. Kim, W. Li, N. Mitra, S. DiVerdi, and T. Funkhouser. Exploring collections of 3d models using fuzzy correspondences. *ACM Trans. Graph.*, 31(4):54:1–54:11, July 2012.
- [26] S. Leonardos, X. Zhou, and K. Daniilidis. Distributed consistent data association via permutation synchronization. In *ICRA*, pages 2645–2652. IEEE, 2017.
- [27] N. Mellado, D. Aiger, and N. J. Mitra. Super 4pcs fast global pointcloud registration via smart indexing. *Comput. Graph. Forum*, 33(5):205–215, Aug. 2014.
- [28] K. Moo Yi, E. Trulls, Y. Ono, V. Lepetit, M. Salzmann, and P. Fua. Learning to find good correspondences. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [29] A. Nguyen, M. Ben-Chen, K. Welnicka, Y. Ye, and L. J. Guibas. An optimization approach to improving collections of shape maps. *Comput. Graph. Forum*, 30(5):1481–1491, 2011.
- [30] O. Özyesil and A. Singer. Robust camera location estimation by convex programming. *CoRR*, abs/1412.0165, 2014.
- [31] D. Pachauri, R. Kondor, G. Sargur, and V. Singh. Permutation diffusion maps (PDM) with application to the image association problem in computer vision. In *NIPS*, pages 541–549, 2014.

- [32] D. Pachauri, R. Kondor, and V. Singh. Solving the multi-way matching problem by permutation synchronization. In *NIPS*, pages 1860–1868, 2013.
- [33] R. Ranftl and V. Koltun. Deep fundamental matrix estimation. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part I*, pages 292–309, 2018.
- [34] Y. Shen, Q. Huang, N. Srebro, and S. Sanghavi. Normalized spectral map synchronization. In *NIPS*, pages 4925–4933, 2016.
- [35] A. Singer and H. tieng Wu. Vector diffusion maps and the connection laplacian. *Communications in Pure and Applied Mathematics*, 65(8), Aug. 2012.
- [36] Y. Sun, Z. Liang, X. Huang, and Q. Huang. Joint map and symmetry synchronization. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part V*, pages 257–275, 2018.
- [37] C. Sweeney, T. Sattler, T. Höllerer, M. Turk, and M. Pollefeys. Optimizing the viewing graph for structure-from-motion. In *ICCV*, pages 801–809. IEEE Computer Society, 2015.
- [38] L. Wang and A. Singer. Exact and stable recovery of rotations for robust synchronization. *Information and Inference: A Journal of the IMA*, 2:145193, December 2013.
- [39] K. Wilson and N. Snavely. Robust global translations with ldsfm. In D. J. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *ECCV (3)*, volume 8691 of *Lecture Notes in Computer Science*, pages 61–75. Springer, 2014.
- [40] C. Zach, M. Klopschitz, and M. Pollefeys. Disambiguating visual relations using loop constraints. In *CVPR*, pages 1426–1433. IEEE Computer Society, 2010.
- [41] Q. Zhou, J. Park, and V. Koltun. Fast global registration. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II*, pages 766–782, 2016.
- [42] Q. Zhou, J. Park, and V. Koltun. Open3d: A modern library for 3d data processing. *CoRR*, abs/1801.09847, 2018.
- [43] T. Zhou, Y. J. Lee, S. X. Yu, and A. A. Efros. Flowweb: Joint image set alignment by weaving consistent, pixel-wise correspondences. In *CVPR*, pages 1191–1200. IEEE Computer Society, 2015.
- [44] X. Zhou, M. Zhu, and K. Daniilidis. Multi-image matching via fast alternating minimization. *CoRR*, abs/1505.04845, 2015.

## A. Overview

We organize this supplemental material as follows. In Section B, we provide more detailed experimental results. In Section C, we describe the technical proofs for all the propositions in the main paper. In Section D, we show the scenes we used in this paper.

## B. More Experimental Results

### B.1. More Visual Comparison Results

Figure 6 shows more visual comparisons between our approach and baseline approaches. Again, our approach produces alignments that are close to the underlying ground-truth. The overall quality of our alignments is superior to that of the baseline approaches.

### B.2. Cumulative Density Function

Figure 7 plots the cumulative density functions of errors in rotations and translations with respect to a varying threshold.

### B.3. Illustration of Dataset

To understand the difficulty of the datasets used in our experiments, we pick a typical scene from each of the Redwood and ScanNet datasets and render 15 out of 30 ground truth point clouds from the same camera view point. From Figure 9 and Figure 8, we can see that ScanNet is generally harder than Redwood, as there is less information that can be extracted by looking at pairs of scans.

## C. Proofs of Propositions

We organize this section as follows. In Section C.1, we provide key lemmas regarding the eigen-decomposition of a connection Laplacian, including stability of eigenvalues/eigenvectors and derivatives of eigenvectors with respect to elements of the connection Laplacian. In Section C.2, we provide key lemmas regarding the projection operator that maps the space of square matrices to the space of rotations. Section C.3 to Section C.6 describe the proofs of all the propositions stated in the main paper. Section C.7 provides an exact recovery condition of a rotation synchronization scheme via reweighted least squares. Finally, Section C.8 provides proofs for new key lemmas introduced in this section.

### C.1. Eigen-Stability of Connection Laplacian

We begin with introducing the problem setting and notations in Section C.1.1. We then present the key lemmas in Section C.1.2.

#### C.1.1 Problem Setting and Notations

Consider a weighted graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  with  $n$  vertices, i.e.,  $|\mathcal{V}| = n$ . We assume that  $\mathcal{G}$  is connected. With  $w_{ij} > 0$  we



Figure 6: We show the results of ground truth result (column I), Rotation Averaging [6]+Translation Sync. [19] (column II), Geometric Registration [10] (column III), and Our Approach (column IV). These scenes are from Redwood Chair dataset.

denote an edge weight associated with edge  $(i, j) \in \mathcal{E}$ . Let  $\bar{L}$  be the weighted adjacency matrix (Note that we drop  $w$  from the expression of  $\bar{L}$  to make the notations uncluttered). It is clear that the leading eigenvector of  $\bar{L}$  is  $\frac{1}{\sqrt{n}}\mathbf{1} \in \mathbb{R}^n$ , and its corresponding eigenvalue is zero. In the following, we shall denote the eigen-decomposition of  $\bar{L}$  as

$$\bar{L} = \bar{U}\bar{\Lambda}\bar{U}^T,$$

where

$$\bar{U} = (\bar{u}_2, \dots, \bar{u}_n) \text{ and } \bar{\Lambda} = \text{diag}(\bar{\lambda}_2, \dots, \bar{\lambda}_n)$$

collect the remaining eigenvectors and their corresponding eigenvalues of  $L(w)$ , respectively. Our analysis will also use a notation that is closely related to the pseudo-inverse

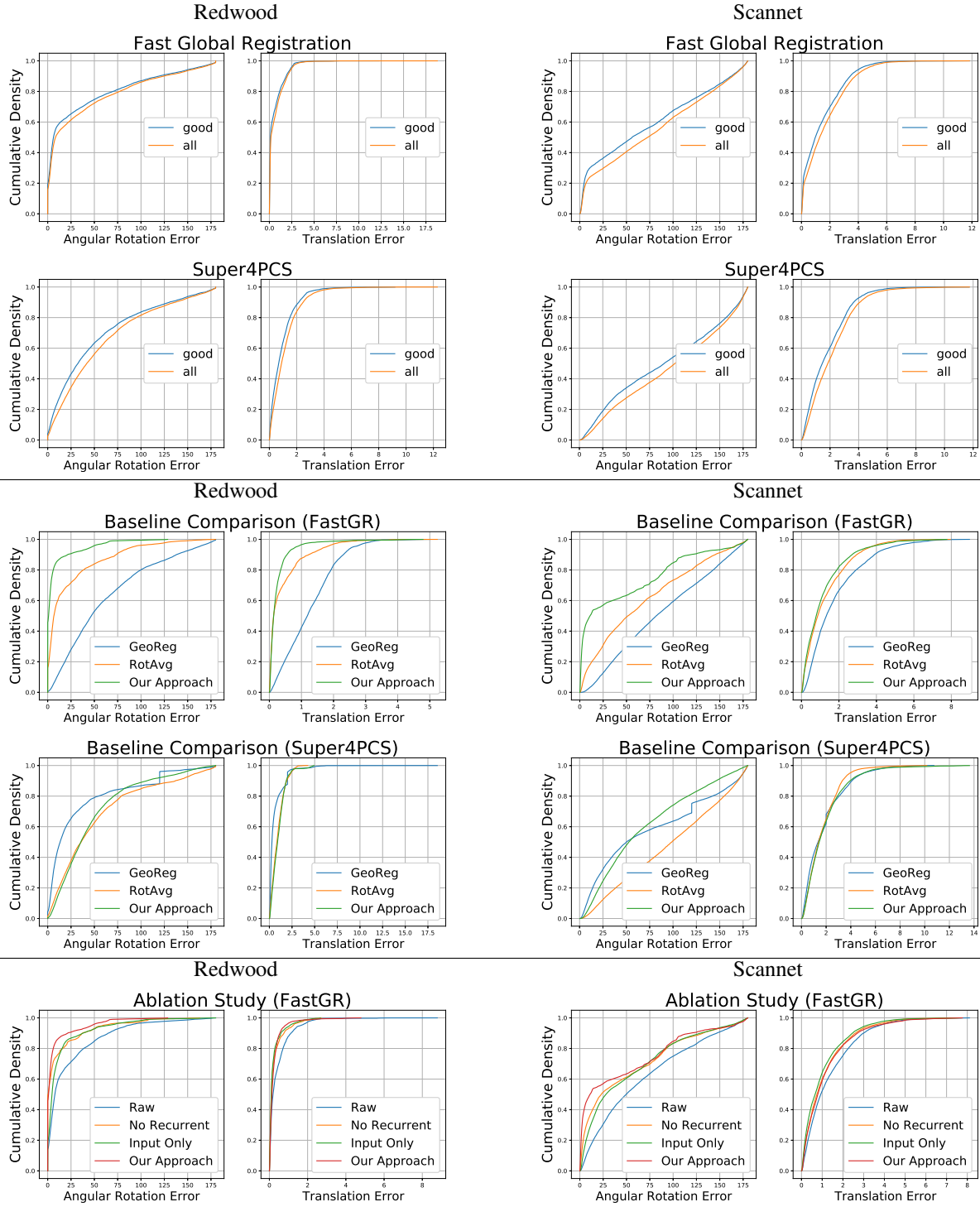


Figure 7: Corresponding cumulative density function (CDF) curves. For the top block, we plot CDF from different input sources. Here "all" corresponds to errors between all pairs and "good" corresponds to errors between selected pairs. The pairs were selected by 1) computing ICP refinement, 2) computing overlapping region by finding points in source point clouds that are close to target point clouds (i.e. by setting a threshold), 3) for these points, we compute their median distance to the target point clouds. For the middle block, we report the comparison of baselines and our approach. Results from different input sources are reported separately. For the bottom block, we report the comparison between variants of our approach using Fast Global Registration as the input pairwise alignments.



Figure 8: A typical example of the a Redwood Chair scene: the 1st, 3rd, 5th, 7th, . . . , 29th of the selected scans are rendered from the same camera view point. Each scan is about 40 frames away from the next one.

of  $\bar{L}$ :

$$\bar{L}_t^+ := \bar{U}(\bar{\Lambda} + tI_{n-1})^{-1}\bar{U}^T, \quad \forall |t| < \bar{\lambda}_2. \quad (15)$$

Our goal is to understand the behavior of the leading eigenvectors of  $\bar{L} \otimes I_k + E^2$  for a symmetric perturbation matrix  $E \in \mathbb{R}^{nk \times nk}$ , which is a  $n \times n$  block matrix whose

<sup>2</sup>Note that when applying the stability results to the problem studied in this paper, we always use  $k = 3$ . However, when assume a general  $k$  when describing the stability results.

blocks are given by

$$E_{ij} = \begin{cases} 0 & i = j \\ -w_{ij}N_{ij} & (i, j) \in \mathcal{E} \end{cases}$$

where  $N_{ij}$  is the perturbation imposed on  $R_{ij}$ .

We are interested in  $U \in \mathbb{R}^{nk \times k}$ , which collects the leading  $k$  eigenvectors of  $\bar{L} \otimes I_k + E$  in its columns. With  $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_k$  we denote the corresponding eigenvalues. Note that due to the property of connection Laplacian,  $\lambda_i \geq 0, 1 \leq i \leq k$ . Our goal is to 1) bound the

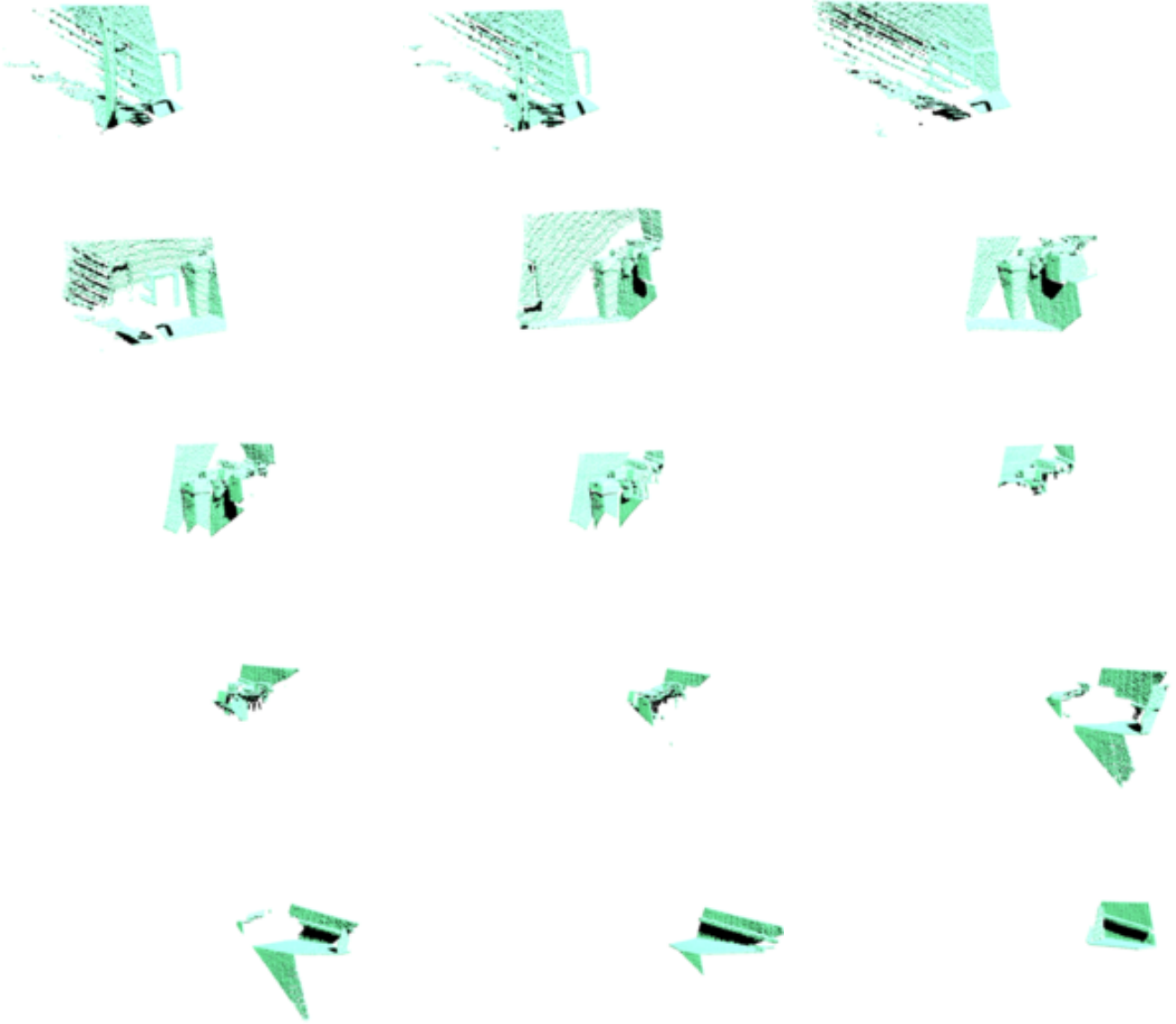


Figure 9: A typical example of the a ScanNet scene: the 1st, 3rd, 5th, 7th, . . . , 29th of the selected scans are rendered from the same camera view point. Each scan is about 40 frames away from the next one.

eigenvalues  $\lambda_i, 1 \leq i \leq k$ , and 2) to provide block-wise bounds between  $U$  and  $\frac{1}{\sqrt{n}}\mathbf{1} \otimes Q$ , for some rotation matrix  $Q \in SO(k)$ .

Besides the notations introduced above that are related to Laplacian matrices, we shall also use a few matrix norms. With  $\|\cdot\|$  and  $\|\cdot\|_{\mathcal{F}}$  we denote the spectral norm and Frobenius norm, respectively. Given a vector  $\mathbf{v} \in \mathbb{R}^n$ , we denote  $\|\mathbf{v}\|_{\infty} = \max_{1 \leq i \leq n} |v_i|$  as the element-wise infinity norm. We will also introduce a norm  $\|\cdot\|_{1,\infty}$  for square matrices,

which is defined as

$$\|A\|_{1,\infty} = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}|, \quad \forall A = (a_{ij})_{1 \leq i,j \leq n} \in \mathbb{R}^{n \times n}.$$

We will also use a similar norm defined for  $n \times n$  block matrices  $E \in \mathbb{R}^{nk \times nk}$  (i.e., each block is a  $k \times k$  matrix):

$$\|E\|_{1,\infty} = \max_{1 \leq i \leq n} \sum_{j=1}^n \|E_{ij}\|, \quad \forall E = (E_{ij})_{1 \leq i,j \leq n} \in \mathbb{R}^{nk \times nk}.$$

### C.1.2 Key Lemmas

This section presents a few key lemmas that will be used to establish main stability results regarding matrix eigenvectors and matrix eigenvalues. We begin with the classical result of the Weyl's inequality:

**Lemma C.1. (Eigenvalue stability)** For  $1 \leq i \leq k$ , we have

$$\lambda_i \leq \|E\|. \quad (16)$$

We proceed to describe tools for controlling the eigenvector stability. To this end, we shall rewrite  $U$  as follows:

$$U = \frac{1}{\sqrt{n}} \mathbf{1} \otimes X + Y.$$

Our goal is to bound the deviation between  $X$  and a rotation matrix and blocks of  $Y$ .

We begin with controlling  $X$ , which we adopt a result described in [3]:

**Lemma C.2. (Controlling  $X$ [3])** If

$$\|E\| < \frac{\bar{\lambda}_2}{2},$$

then there exists  $Q \in SO(k)^3$  such that

$$\|X - Q\| \leq 1 - \sqrt{1 - \left( \frac{\|E\|}{\bar{\lambda}_2 - \|E\|} \right)^2}.$$

In particular,

$$\|X - Q\| \leq \left( \frac{\|E\|}{\bar{\lambda}_2 - \|E\|} \right)^2$$

It remains to control the blocks of  $Y$ . We state a formulation that expresses the column of  $Y$  using a series:

**Lemma C.3.** Suppose  $\|E\| < \frac{\bar{\lambda}_2}{2}$ , then  $\forall 1 \leq j \leq k$ ,

$$Y e_j^{(k)} = -\frac{1}{\sqrt{n}} \sum_{l=1}^{\infty} \left( (\bar{L}_{-\lambda_j}^+ \otimes I_k) E \right)^l (\mathbf{1} \otimes X) e_j^{(k)}. \quad (17)$$

We conclude this section by providing an explicit expression for computing the derivative of the leading eigenvectors of a connection Laplacian with its elements:

**Lemma C.4.** Let  $L$  be an  $N \times N$  non-negative definite matrix and its eigen-decomposition is

$$L = \sum_{i=1}^N \lambda_i \mathbf{u}_i \mathbf{u}_i^T \quad (18)$$

<sup>3</sup>If not, we can always negate the last column of  $U$ .

where  $0 \leq \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_N$ .

Suppose  $\lambda_k < \lambda_{k+1}$ . Collect the eigenvectors corresponding to the smallest  $k$  eigenvalues of  $L$  as the columns of matrix  $U_k$ . Namely,  $U_k = [\mathbf{u}_1, \dots, \mathbf{u}_k]$  where  $\sigma_1, \dots, \sigma_k$  are the smallest  $k$  eigenvalues of  $L$ .

Notice that  $L$  can have different decompositions in (18) when there are repetitive eigenvalues. But in our case where  $\lambda_k < \lambda_{k+1}$ , we claim that  $U_k U_k^T$  is unique under different possible decomposition of  $L$  so that  $d(U_k U_k^T)$  can be well defined and has an explicit expression

$$d(U_k U_k^T) = \sum_{i=1}^k \sum_{j=k+1}^N \frac{\mathbf{u}_j^T dL \mathbf{u}_i}{\sigma_i - \sigma_j} (\mathbf{u}_i \mathbf{u}_j^T + \mathbf{u}_j \mathbf{u}_i^T) \quad (19)$$

Moreover, the differentials of eigenvalues are

$$d\sigma_i = \mathbf{u}_i^T dL \mathbf{u}_i. \quad (20)$$

### C.2. Key Lemma Regarding the Projection Operator

This section studies the projection operator which maps the space of square matrices to the space of rotation matrices. We begin with formally defining the projection operator as follows:

**Definition 1.** Suppose  $\det(M) > 0$ . Let  $M = \sum_{i=1}^n \sigma_i \mathbf{u}_i \mathbf{v}_i^T$  be the singular value decomposition of square matrix  $M$  where  $U = [\mathbf{u}_1, \dots, \mathbf{u}_n]$  and  $V = [\mathbf{v}_1, \dots, \mathbf{v}_n]$  are both orthogonal matrices, and all coefficients  $\sigma_i$  are non-negative. Then we define the rotation approximation of  $M$  as

$$R(M) := \sum_{i=1}^n \mathbf{u}_i \mathbf{v}_i^T = UV^T.$$

It is clear that  $R(M)$  is a rotation matrix, since 1) both  $U$  and  $V^T$  are rotations, and 2)  $\det(UV^T) > 0$ .

**Lemma C.5.** Let  $A \in \mathbb{R}^{n \times k}$  be a block matrix of form

$$A = \begin{bmatrix} A_1 \\ \vdots \\ A_n \end{bmatrix}$$

where  $A_i \in \mathbb{R}^{k \times k}$ . Use  $a_{ij}$  to denote the element on position  $i, j$  in  $A$ . Then we have

$$\sum_{i=1}^n \|A_i\|^2 \leq k \|A\|^2$$

We then present the following key lemma regarding the stability of the projection operator:

**Lemma C.6.** Let  $M$  be a square matrix and  $\epsilon = \|M - I\|$ . Suppose  $\epsilon < \frac{1}{3}$ , then

$$\|R(M) - I\| \leq \epsilon + \epsilon^2.$$

**Lemma C.7.** Regarding  $R(M)$  as a function about  $M$ , then the differential of  $R(M)$  would be

$$dR(M) = \sum_{i \neq j} \frac{\mathbf{u}_i^T dM \mathbf{v}_j - \mathbf{u}_j^T dM \mathbf{v}_i}{\sigma_i + \sigma_j} \mathbf{u}_i \mathbf{v}_j^T$$

where all notations follow Definition (1).

### C.3. Proof of Prop. 4.1

We first present a formal version of Prop. 4.1 in the main paper.

**Proposition 5.** Suppose the underlying rotations are given by  $R_i^*$ ,  $1 \leq i \leq n$ . Modify the definition of  $E$  such that

$$E_{ij} = \begin{cases} -w_{ij} R_j^* (R_j^{*T} R_{ij} R_i^* - I_k) R_i^{*T} & (i, j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}$$

Define

$$\epsilon_1 := \frac{2\|E\|_{1,\infty}}{\lambda_2}, \quad \epsilon_2 := \|\bar{L}^+\|_{1,\infty} \|E\|_{1,\infty}. \quad (21)$$

Suppose  $\epsilon_1 < 1$ ,  $\epsilon_2 < 1$ , and

$$\delta := \left(\frac{\epsilon_1}{2 - \epsilon_1}\right)^2 + \sqrt{k} \cdot \left(1 + \left(\frac{\epsilon_1}{2 - \epsilon_1}\right)^2\right) \cdot \frac{\epsilon_2(1 + \epsilon_2)}{1 - \epsilon_2(1 + \epsilon_2)} < \frac{1}{3}.$$

Then the optimal solution  $R_i$ ,  $1 \leq i \leq n$  to the rotation synchronization step satisfies that there exists  $Q \in SO(k)$ ,

$$\max_{1 \leq i \leq n} \|R_i - R_i^* Q\| \leq \delta + \delta^2. \quad (22)$$

Note that Prop. 5 induces Prop.4.1 in the main paper, because when  $R_{ij} \approx R_j^* R_i^{*T}$ ,  $\forall w_{ij} > 0$ , we have  $\|E\|_{1,\infty} \approx 0$ , meaning  $\epsilon_1 \approx 0$  and  $\epsilon_2 \approx 0$ .

**Proof of Prop. 5:** Without losing generality, we assume  $R_i^* = I_k$ ,  $1 \leq i \leq k$  when proving Prop. 5. In fact, we can always apply an unitary transform to obtain  $\text{diag}(R_1^*, \dots, R_n^*)^T L \text{diag}(R_1^*, \dots, R_n^*)$ , which does not impact the structure of the eigen-decomposition, and which satisfies the assumption.

Before proving Prop.5, we shall utilize two Lemmas, whose proofs are deferred to Section C.8.

**Lemma C.8.** Under the assumptions described above, we have

$$\|\bar{L}_{-\lambda_j}^+\|_{1,\infty} \leq \|\bar{L}^+\|_{1,\infty} (1 + \|\bar{L}^+\|_{1,\infty} \|E\|_{1,\infty}). \quad (23)$$

**Lemma C.9.** Given a  $k \times k$  matrix  $A$ , we have

$$\|A\| \leq \sqrt{k} \max_{1 \leq j \leq k} \|A \mathbf{e}_j^{(k)}\|. \quad (24)$$

Now we proceed to complete the proof of Prop.5. First of all, applying Lemma C.2, we obtain that there exists  $Q \in SO(k)$  such that

$$\|X - Q\| \leq \left(\frac{\epsilon_1}{2 - \epsilon_1}\right)^2. \quad (25)$$

Applying Lemma C.3, we have  $\forall 1 \leq j \leq k$ ,

$$\begin{aligned} & \sqrt{n} \|(\mathbf{e}_i^{(n)} \otimes I_k) Y \mathbf{e}_j^{(k)}\| \\ & \leq \sum_{l=1}^{\infty} \|(\bar{L}_{-\lambda_j}^+)^l E\|_{1,\infty}^l \|X\| \\ & \leq \sum_{l=1}^{\infty} (\|\bar{L}_{-\lambda_j}^+\|_{1,\infty} \|E\|_{1,\infty})^l \|X\| \\ & = \frac{\|\bar{L}_{-\lambda_j}^+\|_{1,\infty} \|E\|_{1,\infty}}{1 - \|\bar{L}_{-\lambda_j}^+\|_{1,\infty} \|E\|_{1,\infty}} \|X\| \\ (23) \quad & \frac{\|\bar{L}^+\|_{1,\infty} \|E\|_{1,\infty} (1 + \|\bar{L}^+\|_{1,\infty} \|E\|_{1,\infty})}{1 - \|\bar{L}^+\|_{1,\infty} \|E\|_{1,\infty} (1 + \|\bar{L}^+\|_{1,\infty} \|E\|_{1,\infty})} \cdot \|X\| \\ & \leq \frac{\epsilon_2(1 + \epsilon_2)}{1 - \epsilon_2(1 + \epsilon_2)} \cdot \left(1 + \left(\frac{\epsilon_1}{2 - \epsilon_1}\right)^2\right). \end{aligned} \quad (27)$$

We can now conclude the proof by combining (25), (27), Lemma C.9, and Lemma C.6.  $\square$

### C.4. Proof of Prop. 4.2

In the same spirit as the preceding section, we assume the underlying ground-truth satisfies

$$R_i^* = I_k, \mathbf{t}_i^* = \mathbf{0}, \quad 1 \leq i \leq n. \quad (28)$$

In other words, a correct measurement along edge  $(i, j) \in \mathcal{E}$  should satisfy  $R_{ij} = I_k, \mathbf{t}_{ij} = \mathbf{0}$ . As we will see later, this assumption makes the error bound easier to parse. It is easy to see that the more general setting can always be converted into this simple setup through factoring out the rigid transformations among the coordinate systems associated with the input objects.

We present a formal statement of Prop. 4.2 of the main paper as follows:

**Proposition 6.** Consider the assumption of (28). Define

$$\epsilon_3 := \max_{1 \leq i \leq n} \sum_{j \in \mathcal{N}(i)} w_{ij} \|t_{ij}\| \quad (29)$$

Intuitively,  $\epsilon_3$  measures the cumulative transformation error associated with each object. Under the same assumption as



Prop. 5 for the connection Laplacian  $L = \bar{L} \otimes I_k + E$ , we can bound the error of the translation synchronization step as

$$\max_{1 \leq i \leq n} \|\mathbf{t}_i\| \leq \frac{\|\bar{L}^+\|_{1,\infty} \epsilon_3}{1 - 4\epsilon_2}. \quad (30)$$

**Proof of Lemma 6:** First of all, note that  $(\mathbf{1} \otimes I_k)^T \mathbf{b} = 0$ . Thus we can factor out the component in  $E$  that corresponds to the subspace spanned by  $\mathbf{1} \otimes I_k$ . Specifically, define

$$E' = \left( (I - \frac{1}{n} \mathbf{1}\mathbf{1}^T) \otimes I_k \right) E \left( (I - \frac{1}{n} \mathbf{1}\mathbf{1}^T) \otimes I_k \right).$$

It is easy to check that

$$\|E'\|_{1,\infty} \leq 4\|E\|_{1,\infty}.$$

Moreover,

$$\mathbf{t} = (\bar{L} \otimes I_k + E')^+ \mathbf{b}.$$

This means

$$\max_{1 \leq i \leq n} \|\mathbf{t}_i\| \leq \|(\bar{L} \otimes I_k + E')^+\|_{1,\infty} \epsilon_3.$$

Note that

$$(\bar{L} \otimes I_k + E')^+ = \sum_{l=0}^{\infty} ((\bar{L}^+ \otimes I_k) E')^l (\bar{L}^+ \otimes I_k).$$

It follows that

$$\begin{aligned} & \|(\bar{L} \otimes I_k + E')^+\|_{1,\infty} \\ & \leq \sum_{l=0}^{\infty} \|(\bar{L}^+ \otimes I_k) E'\|_{1,\infty}^l \|(\bar{L}^+ \otimes I_k)\|_{1,\infty} \\ & = \sum_{l=0}^{\infty} (\|\bar{L}^+\|_{1,\infty} \|E'\|_{1,\infty})^l \|\bar{L}^+\|_{1,\infty} \\ & \leq \frac{\|\bar{L}^+\|_{1,\infty}}{1 - 4\|\bar{L}^+\|_{1,\infty} \|E\|_{1,\infty}}. \end{aligned}$$

### C.5. Proof of Prop. 4.3

Applying Lemma C.7, we have

$$dR_i = \sum_{1 \leq s, t \leq k} \frac{\mathbf{v}_s^{(i)T} dU_i \mathbf{w}_t^{(i)} - \mathbf{v}_t^{(i)T} dU_i \mathbf{w}_s^{(i)}}{\sigma_s^{(i)} + \sigma_t^{(i)}} \mathbf{v}_s^{(i)} \mathbf{w}_t^{(i)T}. \quad (31)$$

We further divide the computation of  $dU_i$  into two parts.

Consider the  $j$ -th column of  $dU_i$ :

$$\begin{aligned} dU_i \mathbf{e}_j^{(k)} &= (\mathbf{e}_i^{(n)T} \otimes I_k) dU \mathbf{e}_j^{(k)} \\ &= (\mathbf{e}_i^{(n)T} \otimes I_k) d\mathbf{u}_j \\ &= (\mathbf{e}_i^{(n)T} \otimes I_k) \sum_{l \neq j} \mathbf{u}_l \frac{\mathbf{u}_j^T dL \mathbf{u}_l}{\lambda_j - \lambda_l} \\ &= (\mathbf{e}_i^{(n)T} \otimes I_k) \left( \sum_{\substack{l=1 \\ l \neq j}}^k \frac{\mathbf{u}_j^T dL \mathbf{u}_l}{\lambda_j - \lambda_l} \mathbf{u}_l + \sum_{l=k+1}^{kn} \frac{\mathbf{u}_j^T dL \mathbf{u}_l}{\lambda_j - \lambda_l} \mathbf{u}_l \right) \\ &= dU_i^{(inner)} \mathbf{e}_j^{(k)} + dU_i^{(outer)} \mathbf{e}_j^{(k)} \end{aligned} \quad (32)$$

where

$$dU_i^{(inner)} \mathbf{e}_j^{(k)} = (\mathbf{e}_i^{(n)T} \otimes I_k) \sum_{\substack{l=1 \\ l \neq j}}^k \frac{\mathbf{u}_l \mathbf{u}_l^T}{\lambda_j - \lambda_l} dL \mathbf{u}_j \quad (33)$$

$$= \sum_{\substack{l=1 \\ l \neq j}}^k (U_i \mathbf{e}_l^{(k)}) \frac{\mathbf{u}_l \mathbf{u}_l^T}{\lambda_j - \lambda_l} dL \mathbf{u}_j \quad (34)$$

$$dU_i^{(outer)} \mathbf{e}_j^{(k)} = (\mathbf{e}_i^{(n)T} \otimes I_k) \sum_{l=k+1}^{kn} \frac{\mathbf{u}_l \mathbf{u}_l^T}{\lambda_j - \lambda_l} dL \mathbf{u}_j \quad (35)$$

In (33), we used the fact that  $(\mathbf{e}_i^{(n)T} \otimes I_k) \mathbf{u}_l$  is just  $U_i \mathbf{e}_l^{(k)}$  by definition of  $U_i$ .

Since  $dR_i$  is linear with respect to  $dU_i$ , we can divide  $dR_i$  similarly:

$$dR_i = dR_i^{(inner)} + dR_i^{(outer)}$$

$$dR_i^{(inner)} :=$$

$$\sum_{1 \leq s, t \leq k} \frac{\mathbf{v}_s^{(i)T} dU_i^{(inner)} \mathbf{w}_t^{(i)} - \mathbf{v}_t^{(i)T} dU_i^{(inner)} \mathbf{w}_s^{(i)}}{\sigma_s^{(i)} + \sigma_t^{(i)}} \mathbf{v}_s^{(i)} \mathbf{w}_t^{(i)T} \quad (36)$$

$$dR_i^{(outer)} :=$$

$$\sum_{1 \leq s, t \leq k} \frac{\mathbf{v}_s^{(i)T} dU_i^{(outer)} \mathbf{w}_t^{(i)} - \mathbf{v}_t^{(i)T} dU_i^{(outer)} \mathbf{w}_s^{(i)}}{\sigma_s^{(i)} + \sigma_t^{(i)}} \mathbf{v}_s^{(i)} \mathbf{w}_t^{(i)T} \quad (37)$$

Then the derivative we would like to compute can be written as

$$\begin{aligned} d(R_i R_j^T) &= dR_i R_j^T + R_i dR_j^T \\ &= dR_i^{(inner)} R_j^T + R_i dR_j^{(inner)T} \\ &\quad + dR_i^{(outer)} R_j^T + R_i dR_j^{(outer)T}. \end{aligned} \quad (38)$$

From (37) and (35) it can be easily checked that the formula in Prop 4.3 that we want to prove is just (38) except the extra terms  $dR_i^{(inner)} R_j^T + R_i dR_j^{(inner)T}$ . Hence in the remaining proof it suffices to show that

$$dR_i^{(inner)} R_j^T + R_i dR_j^{(inner)T} = 0.$$

To this end, we define a  $k$ -by- $k$  auxiliary matrix  $C$  as

$$C_{lj} = \frac{\mathbf{u}_j^T dL \mathbf{u}_l}{\lambda_j - \lambda_l}$$

for all  $l \neq j$  and  $C_{jj} = 0$ . Since  $L$  is symmetric,  $C$  would be skew-symmetric that means  $C + C^T = 0$ . First of all, notice that

$$dU_i^{(inner)} = \sum_{\substack{l=1 \\ l \neq k}}^k (U_i \mathbf{e}_l^{(k)}) \frac{\mathbf{u}_j^T dL \mathbf{u}_l}{\lambda_j - \lambda_l} = U_i C.$$

Also it is clear that

$$\mathbf{v}_s^{(i)T} U_i = \sigma_s \mathbf{w}_s^T$$

by using simple properties of SVD. It follows that

$$dR_i^{(inner)} = \sum_{1 \leq s, t \leq k} \frac{\sigma_s \mathbf{w}_s^{(i)T} C \mathbf{w}_t^{(i)} - \sigma_t \mathbf{w}_t^{(i)T} C \mathbf{w}_s^{(i)}}{\sigma_s^{(i)} + \sigma_t^{(i)}} \mathbf{v}_s^{(i)} \mathbf{w}_t^{(i)T} \quad (39)$$

$$= \sum_{1 \leq s, t \leq k} \mathbf{w}_s^{(i)T} C \mathbf{w}_t^{(i)} \mathbf{v}_s^{(i)} \mathbf{w}_t^{(i)T} \quad (40)$$

$$= \sum_{1 \leq s, t \leq k} \mathbf{v}_s^{(i)} \mathbf{w}_s^{(i)T} C \mathbf{w}_t^{(i)} \mathbf{w}_t^{(i)T} \quad (41)$$

$$= V^{(i)} W^{(i)T} C \quad (42)$$

$$= R_i C. \quad (43)$$

In the derivations above, we used the fact that  $C$  is a skew-symmetric matrix for deriving the first equality (39). In addition, we used the fact that  $\mathbf{v}_s^{(i)T} C \mathbf{v}_t^{(i)}$  is a scalar for deriving the second equality (40). When deriving (41), we used the expansion of  $U^{(i)} V^{(i)T}$  and the fact that  $\{\mathbf{v}_1^{(i)}, \dots, \mathbf{v}_k^{(i)}\}$  form an orthonormal basis:

$$U^{(i)} V^{(i)T} = \sum_{s=1}^k \mathbf{u}_s^{(i)} \mathbf{v}_s^{(i)T}, \quad I = \sum_{t=1}^k \mathbf{v}_t^{(i)} \mathbf{v}_t^{(i)T}.$$

(42) uses the definition of  $R_i$ . Finally, plugging (43) into (38) gives

$$dR_i^{(inner)} R_j^T + R_i dR_j^{(inner)T} = R_i C R_j^T + R_i C^T R_j^T = 0,$$

which completes our proof.  $\square$

## C.6. Proof of Prop. 4.4

The proof of Prop. 4.4 is straightforward, since

$$0 = d(L \cdot L^{-1}) = dL \cdot L^{-1} + L \cdot d(L^{-1}),$$

meaning

$$dL = -L^{-1} dL L^{-1}.$$

In the degenerate case, we replace  $L^{-1}$  by  $L^+$ . This is proper since the only null space of  $L$  is  $\mathbf{1} \otimes I_k$ , which does not affect the solution  $\mathbf{t}$ .  $\square$

## C.7. Exact Recovery Condition of Rotation Synchronization

Similar to [19], we can derive a truncated rotation synchronization scheme (the generalization to transformation synchronization is straight-forward). Specifically, consider an observation graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ . Let  $\mathcal{E}_{bad} \subset \mathcal{E}$  be the edge set associated with incorrect rotation measurements. Starting from  $\mathcal{G}$ , at each iteration, we use the solution  $R_i^{(k)}, 1 \leq i \leq n$  at the  $k$ th iteration to prune input rotations whenever  $\|R_{ij} - R_j^{(k)} R_i^{(k)T}\| \geq 2\gamma^k$ , where  $\gamma < 1$  is a constant. Using Prop. 5, we can easily derive the following exact recovery condition:

**Proposition 7.** *The truncated rotation synchronization scheme recovers the underlying ground-truth if*

$$\|L_{\mathcal{G}}^+\|_{1, \infty} d_{\max}(\mathcal{E}_{bad}) \leq \frac{1}{16}, \quad \gamma > 0.95, \quad (44)$$

where  $L_{\mathcal{G}}$  is the graph Laplacian of  $\mathcal{G}$ , and  $d_{\max}(\mathcal{E}_{bad})$  is the maximum number of bad edges per vertex. Note that the constants in (44) are not optimized.

*Proof:* Denote  $\epsilon_4 := \|L_{\mathcal{G}}^+\|_{1, \infty} d_{\max}(\mathcal{E}_{bad})$ . Consider an arbitrary set  $\mathcal{E}_r \subseteq \mathcal{E}_{bad}$ . Introduce the graph that collects the corresponding remaining observations  $\mathcal{G}_{cur} = (\mathcal{V}, E_{cur})$ , where  $E_{cur} = \mathcal{E} \setminus (\mathcal{E}_{bad} \setminus \mathcal{E}_r)$ . Suppose we apply rotation synchronization step to  $\mathcal{G}_{cur}$  and the associated observations, it is easy to show that (c.f.[19])

$$\epsilon_2 \leq 2 \frac{\epsilon_4}{1 - \epsilon_4} \cdot \max_{(i,j) \in \mathcal{E}_{cur}} \|N_{ij}\|, \quad \epsilon_1 \leq 2\epsilon_2.$$

Using Prop. 5 and after simple calculations, we can derive that the truncated scheme described above will never remove good measurements, which end the proof.  $\square$

**Remark 1.** This exact recovery condition suggests that if we simply let the weighting function to be small when the residual is big, then if the ratio of the incorrect measurements is small. It is guaranteed to remove all the incorrect measurement. Yet to maximize the effectiveness of the weighting scheme, it is suggested to learn the optimal weighting scheme from data. The approach presented in the main paper is one attempt in this direction.

## C.8. Proofs of Key Lemmas

### C.8.1 Proof of Lemma C.3

We first introduce the following notations, which essentially express  $E$  in the coordinate system spanned by  $\frac{1}{\sqrt{n}}(\mathbf{1} \otimes I_k)$  and  $\bar{U} \otimes I_k$ :

$$\begin{aligned} E_{11} &:= \frac{1}{n}(\mathbf{1} \otimes I_k)^T E(\mathbf{1} \otimes I_k), \\ E_{12} &:= \frac{1}{\sqrt{n}}(\mathbf{1} \otimes I_k)^T E(\bar{U} \otimes I_k), \\ E_{21} &:= \frac{1}{\sqrt{n}}(\bar{U} \otimes I_k)^T E(\mathbf{1} \otimes I_k), \\ E_{22} &:= (\bar{U} \otimes I_k)^T E(\bar{U} \otimes I_k). \end{aligned}$$

Let  $Y := (U \otimes I_k)\bar{Y}$ . Substituting  $U = \frac{1}{\sqrt{n}}(\mathbf{1} \otimes I_k) + (\bar{U} \otimes I_k)\bar{Y}$  into

$$(\bar{L} \otimes I_k + E)U = U\Lambda,$$

we obtain

$$\begin{aligned} &(\bar{L} \otimes I_k + E)\left(\frac{1}{\sqrt{n}}(\mathbf{1} \otimes I_k) + (\bar{U} \otimes I_k)\bar{Y}\right) \\ &= \left(\frac{1}{\sqrt{n}}(\mathbf{1} \otimes I_k) + (\bar{U} \otimes I_k)\bar{Y}\right)\Lambda. \end{aligned}$$

Multiply both sides by  $(U \otimes I_k)^T$ , it follows that

$$E_{21}X + (\bar{\Lambda} \otimes I_k)\bar{Y} + E_{22}\bar{Y} = \bar{Y}\Lambda.$$

Since  $\|E\| < \frac{\bar{\lambda}_2}{2}$ , we have

$$\begin{aligned} Y e_j^{(k)} &:= (\bar{U} \otimes I_k)\bar{Y} e_j^{(k)} \\ &= -(\bar{U} \otimes I_k)((\bar{\Lambda} - \lambda_j) \otimes I_k - E_{22})^{-1} E_{21} X e_j^{(k)} \\ &= -\sum_{l=0}^{\infty} (\bar{U} \otimes I_k) \left( ((\bar{\Lambda} - \lambda_j)^{-1} \otimes I_k) E_{22} \right)^l \\ &\quad \cdot ((\bar{\Lambda} - \lambda_j)^{-1} \otimes I_k) E_{21} X e_j^{(k)} \\ &= -\sum_{l=1}^{\infty} \left( (\bar{U}(\bar{\Lambda} - \lambda_j)^{-1} \bar{U}^T) \otimes I_k E \right)^l \frac{1}{\sqrt{n}} \mathbf{1} \otimes X e_j^{(k)} \\ &= -\frac{1}{\sqrt{n}} \sum_{l=1}^{\infty} (L_{-\lambda_j}^+ E)^l (\mathbf{1} \otimes X e_j^{(k)}). \end{aligned}$$

□

### C.8.2 Proof of Lemma C.4

Let

$$L = \sum_{i=1}^N \sigma_i \mathbf{u}_i \mathbf{u}_i^T = \sum_{i=1}^N \sigma_i \mathbf{u}'_i \mathbf{u}'_i{}^T$$

be two different decompositions of  $L$ . It can be written in matrix form

$$L = U\Lambda U^T = U'\Lambda U'^T$$

where  $U = [\mathbf{u}_1, \dots, \mathbf{u}_N]$ ,  $\Lambda = \text{diag}(\sigma_1, \dots, \sigma_N)$ ,  $U' = [\mathbf{u}'_1, \dots, \mathbf{u}'_N]$ . Then we have

$$(U'^T U)\Lambda = \Lambda(U'^T U)$$

Let  $A = U'^T U$  and the element of position  $(i, j)$  on  $A$  be  $a_{ij}$ , then we have

$$a_{ij}\sigma_j = \sigma_i a_{ij},$$

which means  $a_{ij} = 0$  for all  $\sigma_i \neq \sigma_j$ .

Since we have assumed  $\sigma_1 \leq \dots \leq \sigma_k < \sigma_{k+1} \leq \dots \leq \sigma_N$ , the matrix  $A$  would have form

$$\begin{bmatrix} A_{k,k} & O_{k,N-k} \\ O_{N-k,k} & A_{N-k,N-k} \end{bmatrix}.$$

But we have known that  $A$  is an orthogonal matrix, thus  $A_{k,k}$  is also an orthogonal matrix. In this way  $A_{k,k} = U_k'^T U_k$  can be rewritten as

$$U_k = U_k' A_{k,k}$$

and furthermore we have

$$U_k U_k^T = U_k' (A_{k,k} A_{k,k}^T) U_k'^T = U_k' U_k'^T.$$

Since eigen-decomposition is a special case of SVD when dealing with symmetric matrix, (56) gives

$$\begin{aligned} d\mathbf{u}_i &= \sum_{j \neq i} \frac{\sigma_i \mathbf{u}_j^T dL \mathbf{u}_i + \sigma_j \mathbf{u}_i^T dL \mathbf{u}_j}{\sigma_i^2 - \sigma_j^2} \mathbf{u}_j \\ &= \sum_{j \neq i} \frac{\mathbf{u}_i^T dM \mathbf{u}_j}{\sigma_i - \sigma_j} \mathbf{u}_j \end{aligned}$$

in which we used the fact that  $dL$  is also symmetric in the last step.

Finally the differential of  $U_k U_k^T$  can be written as

$$\begin{aligned}
& d(U_k U_k^T) \\
&= d\left(\sum_{i=1}^k \mathbf{u}_i \mathbf{u}_i^T\right) \\
&= \sum_{i=1}^k (d\mathbf{u}_i \mathbf{u}_i^T + \mathbf{u}_i d\mathbf{u}_i^T) \\
&= \sum_{i=1}^k \sum_{\substack{j=1 \\ j \neq i}}^N \frac{\mathbf{u}_i^T dM \mathbf{u}_j}{\sigma_i - \sigma_j} (\mathbf{u}_j \mathbf{u}_i^T + \mathbf{u}_i \mathbf{u}_j^T) \\
&= \sum_{i=1}^k \sum_{j=k+1}^N \frac{\mathbf{u}_i^T dM \mathbf{u}_j}{\sigma_i - \sigma_j} (\mathbf{u}_j \mathbf{u}_i^T + \mathbf{u}_i \mathbf{u}_j^T) + \\
&\quad \sum_{i=1}^k \sum_{j=i+1}^k \left( \frac{\mathbf{u}_i^T dM \mathbf{u}_j}{\sigma_i - \sigma_j} + \frac{\mathbf{u}_i^T dM \mathbf{u}_j}{\sigma_j - \sigma_i} \right) (\mathbf{u}_j \mathbf{u}_i^T + \mathbf{u}_i \mathbf{u}_j^T) \\
&= \sum_{i=1}^k \sum_{j=k+1}^N \frac{\mathbf{u}_i^T dM \mathbf{u}_j}{\sigma_i - \sigma_j} (\mathbf{u}_j \mathbf{u}_i^T + \mathbf{u}_i \mathbf{u}_j^T)
\end{aligned}$$

As for formula (20), taking differential of equation  $L\mathbf{u}_i = \sigma_i \mathbf{u}_i$ , we obtain

$$dL\mathbf{u}_i + Ld\mathbf{u}_i = d\sigma_i \mathbf{u}_i + \sigma_i d\mathbf{u}_i$$

Let us multiply both sides by  $\mathbf{u}_i^T$  and notice that  $\mathbf{u}_i^T \mathbf{u}_i = 1$ ,  $L\mathbf{u}_i = \sigma_i \mathbf{u}_i$ , and  $\mathbf{u}_i^T d\mathbf{u}_i = 0$ , we conclude that the equation above can be simplified to

$$d\sigma_i = \mathbf{u}_i^T dL\mathbf{u}_i.$$

□

### C.8.3 Proof of Lemma C.5

It is well-known that  $\|X\| \leq \|X\|_F$  for any matrix  $X$  where  $\|\cdot\|_F$  represents the Frobenius norm. Thus

$$\begin{aligned}
k\|A\|^2 &= \sum_{j=1}^k \|A\|^2 \geq \sum_{j=1}^k \sum_{i=1}^{kn} a_{ij}^2 \\
&= \sum_{i=1}^{kn} \sum_{j=1}^k a_{ij}^2 = \sum_{i=1}^n \|A_i\|_F^2 \\
&\geq \sum_{i=1}^n \|A_i\|
\end{aligned}$$

completes our proof. □

### C.8.4 Proof of Lemma C.6

Suppose  $M = \sum_{i=1}^n \sigma_i \mathbf{u}_i \mathbf{v}_i^T$  is the SVD decomposition of  $M$ . By definition of  $R(\cdot)$ ,

$$R(M) = \sum_{i=1}^n \mathbf{u}_i \mathbf{v}_i^T.$$

First we have a simple lower bound on  $\epsilon$ :

$$\epsilon = \|M - I\| \geq |(M - I)\mathbf{v}_i| = |\sigma_i - 1|. \quad (45)$$

It is enough to show that for any unit vector  $\mathbf{p} \in \mathbb{R}^n$  we have

$$\|R(M)\mathbf{p} - \mathbf{p}\| \leq (1 + \epsilon)\|M\mathbf{p} - \mathbf{p}\|. \quad (46)$$

In fact, if (46) is true, then

$$\begin{aligned}
\|R(M) - I\| &= \max_{\|\mathbf{p}\|=1} \|R(M)\mathbf{p} - \mathbf{p}\| \\
&\leq \max_{\|\mathbf{p}\|=1} (1 + \epsilon)\|M\mathbf{p} - \mathbf{p}\| \\
&\leq (1 + \epsilon)\epsilon \quad (\text{by definition of } \epsilon).
\end{aligned}$$

By noting  $\{\mathbf{v}_1, \dots, \mathbf{v}_n\}$  are a set of basis on  $\mathbb{R}^n$ , we can decompose  $R(M)$  and  $M$  into

$$\begin{aligned}
R(M)\mathbf{p} - \mathbf{p} &= \sum_{i=1}^n \mathbf{u}_i \mathbf{v}_i^T \mathbf{p} - \sum_{i=1}^n \mathbf{v}_i (\mathbf{v}_i^T \mathbf{p}) \\
&= \sum_{i=1}^n (\mathbf{u}_i - \mathbf{v}_i) (\mathbf{v}_i^T \mathbf{p}) \\
M\mathbf{p} - \mathbf{p} &= \sum_{i=1}^n \sigma_i \mathbf{u}_i \mathbf{v}_i^T \mathbf{p} - \sum_{i=1}^n \mathbf{v}_i (\mathbf{v}_i^T \mathbf{p}) \\
&= \sum_{i=1}^n (\sigma_i \mathbf{u}_i - \mathbf{v}_i) (\mathbf{v}_i^T \mathbf{p})
\end{aligned}$$

To prove (46), it suffices to show that

$$|\mathbf{u}_i - \mathbf{v}_i| \leq (1 + \epsilon)|\sigma_i \mathbf{u}_i - \mathbf{v}_i|.$$

Let  $\delta = \mathbf{u}_i^T \mathbf{v}_i$ . The case that  $\mathbf{u}_i = \mathbf{v}_i$  is trivial. Also, if  $\sigma_i = 0$ , then  $\epsilon \geq 1$  and the resulting inequality

$$|\mathbf{u}_i - \mathbf{v}_i| \leq 2|\mathbf{v}_i|$$

is trivial. Thus we can always assume  $|\sigma_i \mathbf{u}_i - \mathbf{v}_i| \neq 0$  and  $\sigma_i > 0$ . Then by the laws of cosines we have

$$\begin{aligned}
\frac{|\mathbf{u}_i - \mathbf{v}_i|}{|\sigma_i \mathbf{u}_i - \mathbf{v}_i|} &= \sqrt{\frac{2 - 2\delta}{1 + \sigma_i^2 - 2\sigma_i\delta}} \\
&= \sqrt{\sigma_i^{-1} + \frac{2 - (\sigma_i^{-1} + \sigma_i)}{1 + \sigma_i^2 - 2\sigma_i\delta}} \quad (47)
\end{aligned}$$

In (47) it is clear that  $2 - (\sigma_i^{-1} + \sigma_i) \leq 0$  and  $1 + \sigma_i^2 \geq 2\sigma_i\delta$ . Hence by monotonicity (47) reaches its maximum when  $\delta = -1$  and then

$$\frac{|\mathbf{u}_i - \mathbf{v}_i|}{|\sigma_i \mathbf{u}_i - \mathbf{v}_i|} \leq \frac{2}{1 + \sigma_i} = 1 + \frac{1 - \sigma_i}{1 + \sigma_i} \leq 1 + |\sigma_i - 1| \leq 1 + \epsilon$$

### C.8.5 Proof of Lemma C.7

For the sake of brevity we simply write  $R$  instead of  $R(M)$  in the following proof. It is easy to see that

$$M\mathbf{v}_i = \sigma_i \mathbf{u}_i \quad \mathbf{u}_i^T M = \sigma_i \mathbf{v}_i^T$$

for  $i = 1, \dots, n$ . Taking the differential on both sides we obtain

$$dM\mathbf{v}_i + M d\mathbf{v}_i = d\sigma_i \mathbf{u}_i + \sigma_i d\mathbf{u}_i \quad (48)$$

$$d\mathbf{u}_i^T M + \mathbf{u}_i^T dM = d\sigma_i \mathbf{v}_i^T + \sigma_i d\mathbf{v}_i^T \quad (49)$$

Left multiplying both sides of (48) by  $\mathbf{u}_j$  with  $j \neq i$  and observing that  $\mathbf{u}_j^T \mathbf{u}_i = 0$ , we obtain

$$\mathbf{u}_j^T dM\mathbf{v}_i + \mathbf{u}_j^T M d\mathbf{v}_i = \sigma_i \mathbf{u}_j^T d\mathbf{u}_i \quad (50)$$

Similarly right multiplying both sides of (49) by  $\mathbf{v}_j^T$  with  $j \neq i$  gives

$$d\mathbf{u}_i^T M\mathbf{v}_j + \mathbf{u}_i^T dM\mathbf{v}_j = \sigma_i d\mathbf{v}_i^T \mathbf{v}_j \quad (51)$$

Since  $\mathbf{u}_j^T M = \sigma_j \mathbf{v}_j^T$ ,  $M\mathbf{v}_j = \sigma_j \mathbf{u}_j^T$ , we have

$$\mathbf{u}_j^T dM\mathbf{v}_i + \sigma_j \mathbf{v}_j^T d\mathbf{v}_i = \sigma_i \mathbf{u}_j^T d\mathbf{u}_i \quad (52)$$

$$d\mathbf{u}_i^T \sigma_j \mathbf{u}_j + \mathbf{u}_i^T dM\mathbf{v}_j = \sigma_i d\mathbf{v}_i^T \mathbf{v}_j \quad (53)$$

for all  $i \neq j$ .

Observe that  $\mathbf{u}_j^T d\mathbf{u}_i = d\mathbf{u}_i^T \mathbf{u}_j$ ,  $\mathbf{v}_j^T d\mathbf{v}_i = d\mathbf{v}_i^T \mathbf{v}_j$ . Combining (52) and (53) and regarding them as a linear equation group about  $\mathbf{u}_j^T d\mathbf{u}_i$  and  $\mathbf{v}_j^T d\mathbf{v}_i$  they can be solved out as

$$\mathbf{u}_j^T d\mathbf{u}_i = \frac{\sigma_i \mathbf{u}_j^T dM\mathbf{v}_i + \sigma_j \mathbf{u}_i^T dM\mathbf{v}_j}{\sigma_i^2 - \sigma_j^2} \quad (54)$$

$$\mathbf{v}_j^T d\mathbf{v}_i = \frac{\sigma_i \mathbf{u}_i^T dM\mathbf{v}_j + \sigma_j \mathbf{u}_j^T dM\mathbf{v}_i}{\sigma_i^2 - \sigma_j^2} \quad (55)$$

Since  $\mathbf{u}_i^T \mathbf{u}_i = \|\mathbf{u}_i\| = 1$ , we have  $\mathbf{u}_i^T d\mathbf{u}_i = 0$ . As  $\{\mathbf{u}_1, \dots, \mathbf{u}_n\}$  form a set of orthogonal basis of  $\mathbb{R}^n$ , we can write  $\mathbf{u}_i$  as

$$d\mathbf{u}_i = \sum_{j \neq i} \frac{\sigma_i \mathbf{u}_j^T dM\mathbf{v}_i + \sigma_j \mathbf{u}_i^T dM\mathbf{v}_j}{\sigma_i^2 - \sigma_j^2} \mathbf{u}_j \quad (56)$$

Similarly for  $d\mathbf{v}_i$  we have

$$d\mathbf{v}_i = \sum_{j \neq i} \frac{\sigma_i \mathbf{u}_i^T dM\mathbf{v}_j + \sigma_j \mathbf{u}_j^T dM\mathbf{v}_i}{\sigma_i^2 - \sigma_j^2} \mathbf{v}_j \quad (57)$$

Finally we can write  $dR$  as

$$\begin{aligned} dR &= \sum \mathbf{u}_i d\mathbf{v}_i^T + \sum d\mathbf{u}_i \mathbf{v}_i^T \\ &= \sum_{i \neq j} \frac{\sigma_i \mathbf{u}_i^T dM\mathbf{v}_j + \sigma_j \mathbf{u}_j^T dM\mathbf{v}_i}{\sigma_i^2 - \sigma_j^2} \mathbf{u}_i \mathbf{v}_j^T \\ &\quad + \sum_{i \neq j} \frac{\sigma_i \mathbf{u}_j^T dM\mathbf{v}_i + \sigma_j \mathbf{u}_i^T dM\mathbf{v}_j}{\sigma_i^2 - \sigma_j^2} \mathbf{u}_j \mathbf{v}_i^T \\ &= \sum_{i \neq j} \frac{(\sigma_i - \sigma_j) \mathbf{u}_i^T dM\mathbf{v}_j - (\sigma_i - \sigma_j) \mathbf{u}_j^T dM\mathbf{v}_i}{\sigma_i^2 - \sigma_j^2} \mathbf{u}_i \mathbf{v}_j^T \\ &= \sum_{i \neq j} \frac{\mathbf{u}_i^T dM\mathbf{v}_j - \mathbf{u}_j^T dM\mathbf{v}_i}{\sigma_i + \sigma_j} \mathbf{u}_i \mathbf{v}_j^T \end{aligned}$$

□

### C.8.6 Proof of Lemma C.8

Since  $\|E\|_{1,\infty} \leq \frac{\bar{\lambda}_2}{2}$ , we have

$$\begin{aligned} \bar{L}_{-\lambda_j}^+ &= \bar{U}(\bar{\Lambda} - \lambda_j)^{-1} \bar{U}^T \\ &= \bar{U} \sum_{l=0}^{\infty} \bar{L}^{-(l+1)} \lambda_j^l \bar{U}^T \\ &= \sum_{l=0}^{\infty} (\bar{L}^+)^{l+1} \lambda_j^l, \quad 1 \leq j \leq k. \end{aligned}$$

As  $\|E\|_{1,\infty} \|\bar{L}^+\|_{1,\infty} < 1$ , it follows that

$$\begin{aligned} \|\bar{L}_{-\lambda_j}^+\|_{1,\infty} &\leq \sum_{l=0}^{\infty} \|\bar{L}^+\|_{1,\infty}^{l+1} \lambda_j^l \\ &\leq \sum_{l=0}^{\infty} \|\bar{L}^+\|_{1,\infty}^{l+1} \|E\|_{1,\infty}^l \\ &= \|\bar{L}^+\|_{1,\infty} (1 + \|\bar{L}^+\|_{1,\infty} \|E\|_{1,\infty}). \end{aligned}$$

□

### C.8.7 Proof of Lemma C.9

In fact,  $\forall \mathbf{x} \in \mathbb{R}^k$ , where  $\|\mathbf{x}\| = 1$ , we have

$$\begin{aligned} \|A\mathbf{x}\| &\leq \sum_{j=1}^k \|A\mathbf{e}_j^{(k)}\| |x_j| \\ &\leq \max_{1 \leq j \leq k} \|A\mathbf{e}_j^{(k)}\| \sum_{j=1}^k |x_j| \\ &\leq \max_{1 \leq j \leq k} \|A\mathbf{e}_j^{(k)}\| \sqrt{k} \left( \sum_{j=1}^k x_j^2 \right)^{\frac{1}{2}}. \end{aligned}$$

□

## **D. Scenes used in this paper**

For completeness, we show the scenes we used in this paper. Including 100 scenes from ScanNet dataset and 60 scenes from Redwood Chair dataset. Fig. 10 and fig. 11 shows the scenes we used in the paper from ScanNet and Redwood chair dataset, respectively.

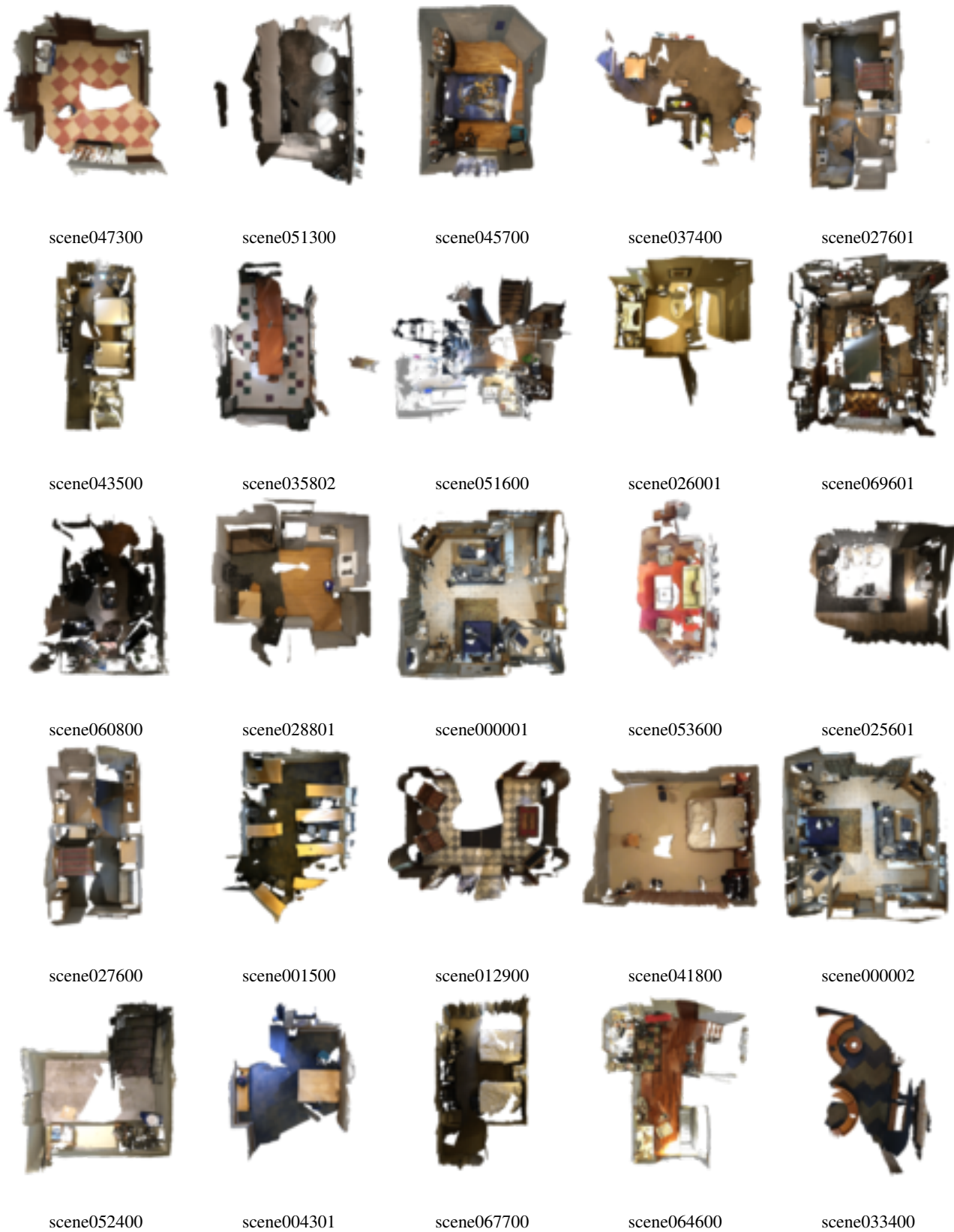


Figure 10: ScanNet Train Dataset (1st to 25th)



Figure 10: ScanNet Train Dataset (26th to 50th)





scene069201



scene031701



scene004700



scene019702



scene013401



scene062500



scene033501



scene035400



scene062900



scene043402



scene009200



scene060901



scene020600



scene000601



scene066900



scene027401



scene067901



scene017702



scene062200

Figure 10: ScanNet Train Dataset (51st to 69th)

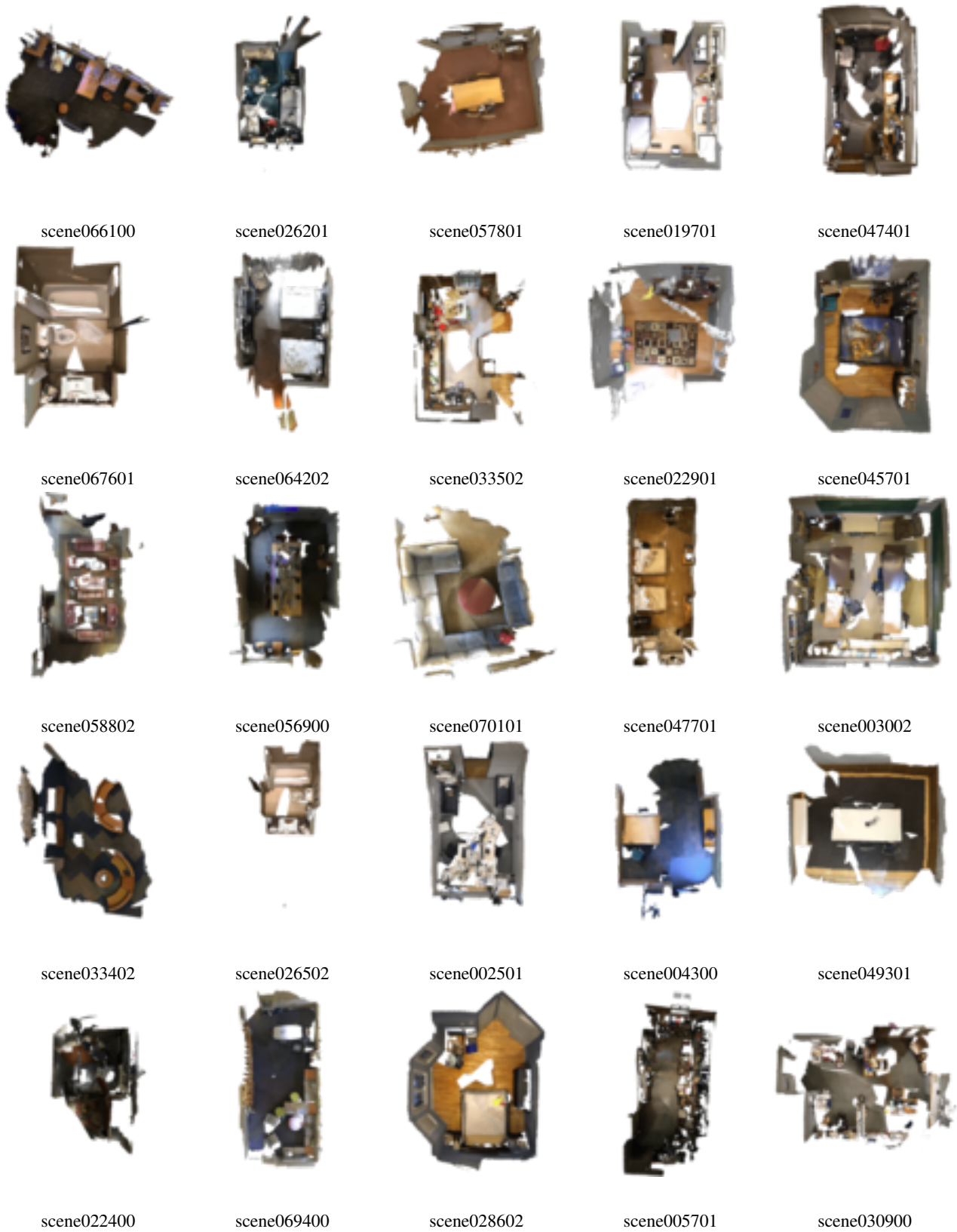


Figure 10: ScanNet Test Dataset (1st to 25th)



scene040602



scene035300



scene022300



scene014602



scene020800



scene057502



scene023101

Figure 10: ScanNet Test Dataset (26th to 32nd)



Figure 11: Redwood Chair Train Dataset (1st to 25th)



06160



01053



05703



05702



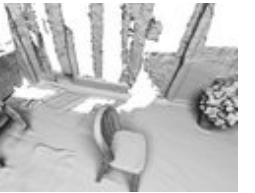
01668



01194



00037



05324

Figure 11: Redwood Chair Train Dataset (26th to 33th)

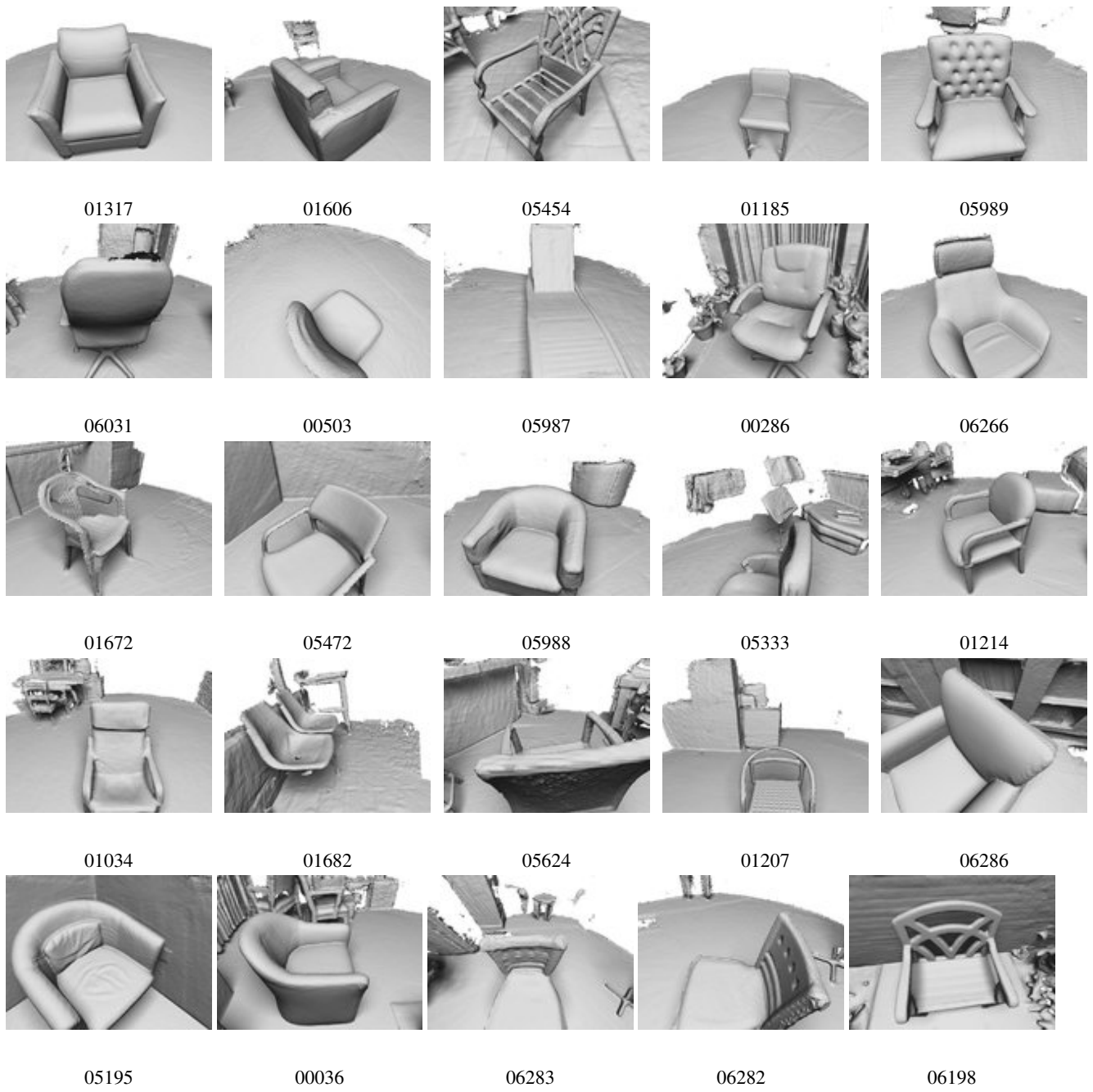


Figure 11: Redwood Chair Test Dataset