Learning multiview 3D point cloud registration

Zan Gojcic∗§  Caifa Zhou∗§  Jan D. Wegner§  Leonidas J. Guibas†  Tolga Birdal†
§ETH Zurich   †Stanford University

Abstract

We present a novel, end-to-end learnable, multiview 3D point cloud registration algorithm. Registration of multiple scans typically follows a two-stage pipeline: the initial pairwise alignment and the globally consistent refinement. The former is often ambiguous due to the low overlap of neighboring point clouds, symmetries and repetitive scene parts. Therefore, the latter global refinement aims at establishing the cyclic consistency across multiple scans and helps in resolving the ambiguous cases. In this paper we propose, to the best of our knowledge, the first end-to-end algorithm for joint learning of both parts of this two-stage problem. Experimental evaluation on well accepted benchmark datasets shows that our approach outperforms the state-of-the-art by a significant margin, while being end-to-end trainable and computationally less costly. Moreover, we present detailed analysis and an ablation study that validate the novel components of our approach. The source code and pretrained models are publicly available under https://github.com/zgojcic/3D_multiview_reg.

1. Introduction

Downstream tasks in 3D computer vision, such as semantic segmentation and object detection typically require a holistic representation of the scene. The capability of aligning and fusing individual point cloud fragments, which cover only small parts of the environment, into a globally consistent holisitic representation is therefore essential and has several use cases in augmented reality and robotics. Pairwise registration of adjacent fragments is a well studied problem and traditional approaches based on geometric constraints [52, 69, 58] and hand-engineered feature descriptors [38, 27, 56, 61] have shown successful results to some extent. Nevertheless, in the recent years, research on local descriptors for pairwise registration of 3D point clouds is centered on deep learning approaches [70, 39, 21, 67, 19, 28] that succeed in capturing and encoding evidence hidden to hand-engineered descriptors. Furthermore, novel end-to-end methods for pairwise point cloud registration were recently proposed [65, 43]. While demonstrating good performance for many tasks, pairwise registration of individual views of a scene has some conceptual drawbacks: (i) low overlap of adjacent point clouds can lead to inaccurate or wrong matches, (ii) point cloud registration has to rely on very local evidence, which can be harmful if 3D scene structure is scarce or repetitive, (iii) separate post-processing is required to combine all pair-wise matches into a global representation. Compared to the pairwise methods, globally consistent multiview alignment of unorganized point cloud fragments is yet to fully benefit from the recent advances achieved by the deep learning methods. State-of-the-art methods typically still rely on a good initialization of the pairwise maps, which they try to refine globally in a subsequent decoupled step [30, 63, 2, 3, 5, 4, 44, 11]. A general drawback of this hierarchical procedure is that global noise distribution over all nodes of the pose graph ends up being far from random, i.e. significant biases persist due to the highly correlated initial pairwise maps.

In this paper, we present, to the best of our knowledge, the first end-to-end data driven multiview point cloud registration algorithm. Our method takes a set of potentially overlapping point clouds as input and outputs a global/absolute transformation matrix per each of the input scans (c.f. Fig. 1). We depart from a traditional two-stage approach where the individual stages are detached from each other and directly learn to register all views of a scene in a globally consistent manner.

The main contributions of our work are:

• We formulate the traditional two-stage approach in an
end-to-end neural network, which in the forward pass solves two differentiable optimization problems: (i) the Procrustes problem for the estimation of the pairwise transformation parameters and (ii) the spectral relaxation of the transformation synchronization.

- We propose a confidence estimation block that uses a novel overlap pooling layer to predict the confidence in the estimated pairwise transformation parameters.
- We cast the multiview 3D point cloud registration problem as an iterative reweighted least squares (IRLS) problem and iteratively refine both the pairwise and absolute transformation estimates.

Resulting from the aforementioned contributions, the proposed multiview registration algorithm (i) is very efficient to compute, (ii) achieves more accurate scan alignments because the residuals are being fed back to the pairwise network in an iterative manner, (iii) outperforms current state-of-the-art on pairwise as well as multiview point cloud registration.

2. Related Work

Pairwise registration The traditional pairwise registration pipeline consists of two stages: the coarse alignment stage, which provides the initial estimate of the relative transformation parameters and the refinement stage that iteratively refines the transformation parameters by minimizing the 3D registration error under the assumption of rigid transformation.

The former is traditionally performed by using either handcrafted \[56, 61, 60\] or learned \[70, 39, 21, 20, 67, 28, 16\] 3D local features descriptors to establish the pointwise candidate correspondences in combination with a RANSAC-like robust estimator \[26, 53, 41\] or geometric hashing \[24, 8, 33\]. A parallel stream of works \[1, 59, 45\] relies on establishing correspondences using the 4-point congruent sets. In the refinement stage, the coarse transformation parameters are often fine-tuned with a variant of the iterative closest point (ICP) algorithm \[6\]. ICP-like algorithms \[42, 66\] perform optimization by alternatively hypothesizing the correspondence set and estimating the new set of transformation parameters. They are known to not be robust against outliers and to converge to a global optimum only when starting with a good prealignment \[9\]. ICP algorithms are often extended to use additional radiometric, temporal or odometry constraints \[72\]. Contemporary to our work, \[65, 43\] propose to integrate coarse and fine pairwise registration stages into an end-to-end learnable algorithm. Using a deep network, \[31\] formulates the object tracking as a relative motion estimation of two point sets.

Multiview registration Multiview, global point cloud registration methods aim at resolving hard or ambiguous cases that arise in pairwise methods by incorporating cues from multiple views. The first family of methods employ a multiview ICP-like scheme to optimize for camera poses as well as 3D point correspondences \[37, 25, 46, 9\]. A majority of these suffer from increased complexity of correspondence estimation. To alleviate this, some approaches only optimize for motion and use the scans to evaluate the registration error \[72, 58, 7\]. Taking a step further, other modern methods make use of the global cycle-consistency and optimize only over the poses starting from an initial set of pairwise maps. This efficient approach is known as synchronization \[10, 63, 2, 58, 3, 5, 44, 72, 7, 36\]. Global structure-from-motion \[17, 73\] aims to synchronize the observed relative motions by decomposing rotation, translation and scale components. \[23\] proposes a global point cloud registration approach using two networks, one for pose estimation and another modelling the scene structure by estimating the occupancy status of global coordinates.

Probably the most similar work to ours is \[36\], where the authors aim to adapt the edge weights for the transformation synchronization layer by learning a data driven weighting function. A major conceptual difference to our approach is that relative transformation parameters are estimated using FPFH \[36\] in combination with FGR \[72\] and thus, unlike ours, are not learned. Furthermore, in each iteration \[36\] has to convert the point clouds to depth images as the weighting function is approximated by a 2D CNN. On the other hand our whole approach operates directly on point clouds, is fully differentiable and therefore facilitates learning a global, multiview point cloud registration in an end-to-end manner.

3. End-to-End Multiview 3D Registration

In this section we derive the proposed multiview 3D registration algorithm as a composition of functions depending upon the data. The network architectures used to approximate these functions are then explained in detail in Sec 4. We begin with a new algorithm for learned pairwise point cloud registration, which uses two point clouds as input and outputs estimated transformation parameters (Sec. 3.1). This method is extended to multiple point clouds by using a transformation synchronization layer amenable to backpropagation (Sec. 3.2). The input graph to this synchronization layer encodes, along with the relative transformation parameters, the confidence in these pairwise maps, which is also estimated using a novel neural network, as edge information. Finally, we propose an IRLS scheme (Sec. 3.3) to refine the global registration of all point clouds by updating the edge weights as well as the pairwise poses.

Consider a set of potentially overlapping point clouds \(S = \{S_i \in \mathbb{R}^{N_i \times 3}, 1 \leq i \leq N_S\}\) capturing a 3D scene from different viewpoints (i.e. poses). The task of multiview registration is to recover the rigid, absolute poses
\{M^*_i \in SE(3)\}_i \text{ given the scan collection, where } SE(3) = \left\{ M \in \mathbb{R}^{4 \times 4} : M = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \right\}, \quad (1)

R_i \in SO(3) \text{ and } t_i \in \mathbb{R}^3. \mathcal{S} \text{ can be augmented by connectivity information resulting in a finite graph } \mathcal{G} = (\mathcal{S}, \mathcal{E}), \text{ where each vertex represents a single point set and the edges } (i, j) \in \mathcal{E} \text{ encode the information about the relative rotation } R_{ij} \text{ and translation } t_{ij} \text{ between the vertices. These relative transformation parameters satisfy } R_{ij} = R_{ji}^T \text{ and } t_{ij} = -R_{ji}^T t_{ji} \text{ as well as the compatibility constraint } [4]

R_{ij} \approx R_i R_j^T \quad t_{ij} \approx -R_i R_j^T t_j + t_i \quad (2)

In current state-of-the-art \cite{72, 36, 7} edges \mathcal{E} \text{ of } \mathcal{G} \text{ are initialized with (noisy) relative transformation parameters } \{M_{ij}\}, \text{ obtained by an independent, auxiliary pairwise registration algorithm. Global scene consistency is enforced via a subsequent synchronization algorithm. In contrast, we propose a joint approach where pairwise registration and transformation synchronization are tightly coupled as one fully differentiable component, which leads to an end-to-end learnable, global registration pipeline.}

3.1. Pairwise registration of point clouds

In the following, we introduce a differentiable, pairwise registration algorithm that can easily be incorporated into an end-to-end multiview 3D registration algorithm. Let \{\mathcal{P}, \mathcal{Q}\} := \{S_i, S_j | i \neq j\} \subset \mathcal{S} \text{ denote a pair of point clouds where } (\mathcal{P})_l := p_l \in \mathbb{R}^3 \text{ and } (\mathcal{Q})_l := q_l \in \mathbb{R}^3 \text{ represent the coordinate vectors of individual points in point clouds } \mathcal{P} \in \mathbb{R}^{N_p \times 3} \text{ and } \mathcal{Q} \in \mathbb{R}^{N_q \times 3}, \text{ respectively. The goal of pairwise registration is to retrieve optimal } \hat{R}_{ij} \text{ and } \hat{t}_{ij}.

\hat{R}_{ij}, \hat{t}_{ij} = \arg \min_{R_{ij}, t_{ij}} \sum_{l=1}^{N_p} \| R_{ij} p_l + t_{ij} - \phi(p_l, Q) \|^2 \quad (3)

where \phi(p, Q) \text{ is a correspondence function} \text{ that maps the points } p \text{ to their corresponding points in point cloud } Q. \text{ The formulation of Eq. 3 facilitates a differentiable closed-form solution, which is—subject to the noise distribution—close to the ground truth solution [57]. However, least square solutions are not robust and thus Eq. 3 will yield wrong transformation parameters in case of high outlier ratio. In practice, the mapping } \phi(p, Q) \text{ is far from ideal and erroneous correspondences typically dominate. To circumvent that, Eq. 3 can be robustified against outliers by introducing a heteroscedastic weighting matrix } [62, 57]:

\hat{R}_{ij}, \hat{t}_{ij} = \arg \min_{R_{ij}, t_{ij}} \sum_{l=1}^{N_p} w_l \| R_{ij} p_l + t_{ij} - \phi(p_l, Q) \|^2 \quad (4)

where } w_l := (w)_l \text{ is the weight of the putative correspondence } \gamma_l \in \mathbb{R}^6 = \{p_l, \phi(p_l, Q)\} \text{ computed by some weighting function } w = \psi_{\text{init}}(\Gamma), \text{ where } \Gamma := \{\gamma_l\} := \{\mathcal{P}, \{\phi(p_l, Q)\}_l\} \text{ and } \psi_{\text{init}} : \mathbb{R}^{N_p \times 6} \rightarrow \mathbb{R}^{N_p}. \text{ Assuming that } w_l \text{ is close to one when the putative correspondence is an inlier and close to zero otherwise, Eq. 4 will yield the correct transformation parameters while retaining a differentiable closed-form solution [57]. Hereinafter we denote this closed-form solution as weighted least squares transformation WLS trans. and for the sake of completeness, its derivation is provided in the supp. material.}

3.2. Differentiable transformation synchronization

Returning to the task of multiview registration, we again consider the initial set of point clouds \mathcal{S}. If no prior connectivity information is given, graph \mathcal{G} \text{ can be initialized by forming } \binom{\mathcal{S}}{2} \text{ point cloud pairs and estimating their relative transformation parameters as described in Sec. 3.1. The global transformation parameters can be estimated either jointly (transformation synchronization) [30, 5, 4, 11] or by dividing the problem into rotation synchronization [2, 3] and translation synchronization [35]. Herein, we opt for the latter approach, which under the spectral relation admits a differentiable closed-form solution [2, 3, 35].}

\textbf{Rotation synchronization} \text{ The goal of rotation synchronization is to retrieve global rotation matrices } \{R^*_i\} \text{ by solving the following minimization problem based on their observed ratios } \{\hat{R}_{ij}\}

R^*_i = \arg \min_{R_i \in SO(3)} \sum_{(i,j) \in \mathcal{E}} c_{ij} \| R_i - R_i R_j^T \|^2 \quad (5)

where the weights } c_{ij} := \zeta_{\text{init}}(\Gamma) \text{ represent the confidence in the relative transformation parameters } \hat{M}_{ij}. \text{ Under the spectral relaxation Eq. 5 admits a closed-form solution, which is provided in the supp. material [2, 3].}

\textbf{Translation synchronization} \text{ Similarly, the goal of translation synchronization is to retrieve global translation vectors } \{t^*_i\} \text{ that minimize the following least squares problem}

t^*_i = \arg \min_{t_i} \sum_{(i,j) \in \mathcal{E}} c_{ij} \| \hat{R}_{ij} t_i + \hat{t}_{ij} - t_j \|^2 (6)

The differentiable closed-form solution to Eq. 6 is again provided in the supp. material.

3.3. Iterative refinement of the registration

The above formulation (Sec. 3.1 and 3.2) facilitates an implementation in an iterative scheme, which in turn can be viewed as an IRLS algorithm. We can start each subsequent iteration \(k + 1\) by pre-aligning the point cloud pairs using the synchronized estimate of the relative transforma-
tion parameters \( M_{ij}^{*(k)} = M_{i}^{*(k)} M_{ij}^{*(k)^{-1}} \) from iteration \( (k) \) such that \( Q^{(k+1)} := M_{ij}^{*(k)} \otimes Q \), where \( \otimes \) denotes applying the transformation \( M_{ij}^{*(k)} \) to point cloud \( Q \). Additionally, weights \( w^{(k)} \) and residuals \( r^{(k)} \) of the previous iteration can be used as a side information in the correspondence weighting function. Therefore, \( \psi_{\text{init}}(\cdot) \) is extended to

\[
\psi_{\text{init}}(\Gamma^{(k+1)}, w^{(k)}, r^{(k)}) = \exp(-\frac{d_{ij}}{t}) / \sum_{k=1}^{N_t} \exp(-d_{ij}/t),
\]

where \( \Gamma^{(k+1)} := \{ \gamma_i^{(k+1)} \} := \{ \mathcal{P}, \{ \phi(p, Q^{(k+1)}) \} \} \).

Analogously, the difference between the input \( M_{ij}^{(k)} \) and the synchronized \( M_{ij}^{*(k)} \) transformation parameters of the \( (k) \)-th iteration can be used as an additional cue for estimating the confidence \( c_{ij}^{(k+1)} \). Thus, \( \psi_{\text{iter}}(\cdot) \) can be extended to

\[
c_{ij}^{(k+1)} = \psi_{\text{iter}}(\Gamma^{(k+1)}, M_{ij}^{(k)}, M_{ij}^{*(k)}).
\]

4. Network Architecture

We implement our proposed multiview registration algorithm as a deep neural network (Fig. 2). In this section, we first describe the architectures used to approximate \( \phi(\cdot), \psi_{\text{init}}(\cdot), \psi_{\text{iter}}(\cdot), \psi_{\text{init}}(\cdot) \) and \( \psi_{\text{iter}}(\cdot) \), before integrating them into one fully differentiable, end-to-end trainable algorithm.

Learned correspondence function Our approximation of the correspondence function \( \phi(\cdot) \) extends a recently proposed fully convolutional 3D feature descriptor FCGF [16] with a soft assignment layer. FCGF operates on sparse tensors [15] and computes 32 dimensional descriptors for each point of the sparse point cloud in a single pass. Note that the function \( \phi(\cdot) \) could be approximated with any of the recently proposed learned feature descriptors [39, 20, 21, 28], but we choose FCGF due to its high accuracy and low computational complexity.

Let \( F_P \) and \( F_Q \) denote the FCGF embeddings of point clouds \( P \) and \( Q \) obtained using the same network weights, respectively. Pointwise correspondences \( \{ \phi(\cdot) \} \) can then be established by a nearest neighbor (NN) search in this high dimensional feature space. However, the selection rule of such hard assignments is not differentiable. We therefore form the NN-selection rule in a probabilistic manner by computing a probability vector \( s \) of the categorical distribution [50]. The stochastic correspondence of the point \( p \) in the point cloud \( Q \) is then defined as

\[
\phi(p, Q) := s^T Q, \quad (s)_i := \frac{\exp(-d_i/t)}{\sum_{j=1}^{N_t} \exp(-d_j/t)},
\]

where \( d_i := ||f_p - (F_Q)_i||_2 \), \( f_p \) is the FCGF embedding of the point \( p \) and \( t \) denotes the temperature parameter. In the limit \( t \to 0 \) the \( \phi(p, Q) \) converges to the deterministic NN-search [50].

We follow [16] and supervise the learning of \( \phi(\cdot) \) with a correspondence loss \( L_c \), which is defined as the hardest contrastive loss and operates on the FCGF embeddings

\[
L_c = \frac{1}{N_{\text{FCGF}}} \sum_{(i,j) \in \mathcal{P}} \left\{ [d(f_i, f_j) - m_p]_+^2 / |\mathcal{P}| + 0.5[m_n - \min_{k \in \mathcal{N}_i} d(f_i, f_k)]_+^2 / |\mathcal{N}_i| + 0.5[m_n - \min_{k \in \mathcal{N}_j} d(f_j, f_k)]_+^2 / |\mathcal{N}_j| \right\}
\]

where \( \mathcal{P} \) is a set of all the positive pairs in a FCGF mini batch \( N_{\text{FCGF}} \) and \( \mathcal{N} \) is a random subset of all features that is used for the hardest negative mining. \( m_p = 0.1 \) and \( m_n = 1.4 \) are the margins for positive and negative pairs.
respectively. The detailed network architecture of $\phi(\cdot)$ as well as the training configuration and parameters are available in the supp. material.

**Deep pairwise registration** Despite the good performance of the FCGR descriptor, several putative correspondences $\Gamma' \subset \Gamma$ will be false. Furthermore, the distribution of inliers and outliers does not resemble noise but rather shows regularity [54]. We thus aim to learn this regularity from the data using a deep neural network. Recently, several networks representing a complex weighting function for filtering of 2D [47, 54, 71] or 3D [29] feature correspondences have been proposed.

Herein, we propose extending the 3D outlier filtering network [29] that is based on [47] with the order-aware blocks proposed in [71]. Specifically, we create a pairwise registration block $f_\theta : \mathbb{R}^{N_P \times 6} \rightarrow \mathbb{R}^{N_P}$ that takes the coordinates of the putative correspondences $\Gamma$ as input and outputs weights $w := \psi_{\text{init}}(\Gamma) := \text{tanh}(\text{ReLU}(f_\theta(\Gamma)))$ that are fed, along with $\Gamma$, into the closed form solution of Eq. 4 to obtain $R_{ij}$ and $t_{ij}$. Motivated by the results in [54, 71] we add another registration block $\psi_{\text{iter}}(\cdot)$ to our network and append the weights $w$ and the pointwise residuals $r$ to the original input s.t. $w^{(k)} := \psi_{\text{iter}}(\text{cat}([P^{(k)}, w^{(k-1)}, r^{(k-1)}]))$ (see Sec. 3.3). The weights $w^{(k)}$ are then, again fed together with the initial correspondences $\Gamma$ to the closed form solution of Eq. 4 to obtain the refined pairwise transformation parameters. In order to ensure permutation-invariance of $f_\theta(\cdot)$ a PointNet-like [51] architecture that operates on individual correspondences is used in both registration blocks. As each branch only operates on individual correspondences, the local 3D context information is gathered in the intermediate layers using symmetric context normalization [68] and order-aware filtering layers [71]. The detailed architecture of the registration block is available in the supp. material. Training of the registration network is supervised using the confidence loss function $\mathcal{L}_\text{conf}$ defined for a batch with $N_{\text{reg}}$ examples as

$$\mathcal{L}_\text{conf} = \alpha_{\text{reg}} L_{\text{class}} + \beta_{\text{reg}} L_{\text{trans}}$$  \hspace{1cm} (10)$$

loss, where $L_{\text{class}}$ denotes the binary cross entropy loss and

$$L_{\text{trans}} = \frac{1}{N_{\text{reg}}} \sum_{(i,j)} \frac{1}{N_P} \sum_{l=1}^{N_P} \|M_{ij} \otimes p_l - M_{ij}^{\text{GT}} \otimes p_l\|_2$$  \hspace{1cm} (11)$$

is used to penalize the deviation from the ground truth transformation parameters $M_{ij}^{\text{GT}}$. $\alpha_{\text{reg}}$ and $\beta_{\text{reg}}$ are used to control the contribution of the individual loss functions.

**Confidence estimation block** Along with the estimated relative transformation parameters $\hat{M}_{ij}$, the edges of the graph $G$ encode the confidence $c_{ij}$ in those estimates. Confidence encoded in each edge of the graph consist of (i) the local confidence $c_{ij}^{\text{local}}$ of the pairwise transformation estimation and (ii) the global confidence $c_{ij}^{\text{global}}$ derived from the transformation synchronization. We formulate the estimation of $c_{ij}^{\text{local}}$ as a classification task and argue that some of the required information is encompassed in the features of the second-to-last layer of the registration block. Let $x_{ij}^{\text{conf}} = f_p(\cdot)$ denote the output of the second-to-last layer of the registration block, we propose an overlap pooling layer $f_{\text{overlap}}$ that extracts a global feature $x_{ij}^{\text{conf}}$ by performing the weighted average pooling as

$$x_{ij}^{\text{conf}} = w_{ij}^T x_{ij}^{\text{conf}}.$$  \hspace{1cm} (12)$$

The obtained global feature is concatenated with the ratio of inliers $\delta_{ij}$ (i.e., the number of correspondences whose weights are higher than a given threshold) and fed to the confidence estimation network with three fully connected layers $(129 - 64 - 32 - 1)$, followed by a ReLU activation function. The local confidence can thus be expressed as

$$c_{ij}^{\text{local}} := \zeta_{\text{init}}(\Gamma) := \text{MLP}(\text{cat}([x_{ij}^{\text{conf}}, \delta_{ij}]))$$  \hspace{1cm} (13)$$

The training of the confidence estimation block is supervised with the confidence loss function $\mathcal{L}_\text{conf}$ defined as

$$\mathcal{L}_\text{conf} = \frac{1}{N} \sum_{(i,j,k)} \text{BCE}(c_{ij}, c_{ij}^{\text{GT}})$$  \hspace{1cm} (15)$$

where $c_{ij}$ are used to classify correspondences as inliers or outliers. $c_{ij}^{\text{GT}}$ denotes the ground truth local confidence. Inliers and outliers do not resemble noise but rather shows regularity.

Global confidence in the relative transformation parameters $c_{ij}^{\text{global}}$ can be expressed with the Cauchy weighting function $\text{cat}([1 + r_{ij}^*])$ following [34, 4]

$$c_{ij}^{\text{global}} = \frac{1}{1 + r_{ij}^* / b}$$  \hspace{1cm} (14)$$

where $r_{ij}^* := \|\hat{M}_{ij} - M_{ij}^{\text{GT}}\|_F$ and following [34, 4] $b = 1.482 \gamma \text{med}(|r^* - \text{med}(r^*)|)$ with $\text{med}(\cdot)$ denoting the median operator and $r^*$ the vectorization of residuals $r_{ij}^*$. Since local and global confidence provide complementary information about the relative transformation parameters, we combine them into a joined confidence $c_{ij}$ using their harmonic mean:

$$c_{ij} := \zeta_{\text{iter}}(c_{ij}^{\text{local}}, c_{ij}^{\text{global}}) := \frac{1}{\beta_2 c_{ij}^{\text{local}}} + \frac{1}{\beta_2 c_{ij}^{\text{global}}}.$$  \hspace{1cm} (15)$$
where the $\beta$ balances the contribution of the local and global confidence estimates and is learned during training.

**End-to-end multiview 3D registration** The individual parts of the network are connected into an end-to-end multiview 3D registration algorithm as shown in Fig. 2. We pre-train the individual sub-networks (training details available in the supp. material) before fine-tuning the whole model in an end-to-end manner on the 3DMatch data set [70] using the official train/test data split. In fine-tuning we use $N_{\text{FCGF}} = 4$ to extract the FCGF features and randomly sample feature vectors of 2048 points per fragment. These features are used in the soft assignment (softNN) to form the putative correspondences of $N_{\text{pair}}$ point cloud pairs, which are fed to the pairwise registration network. The output of the pairwise registration is used to build the graph, which is input to the transformation synchronization layer. The iterative refinement of the transformation parameters is performed four times. We supervise the fine tuning using the joint multiview registration loss

\[
\mathcal{L} = \mathcal{L}_c + \mathcal{L}_\text{reg} + \mathcal{L}_\text{conf} + \mathcal{L}_\text{sync}
\]  

where the transformation synchronization $\mathcal{L}_\text{sync}$ loss reads

\[
\mathcal{L}_\text{sync} = \frac{1}{N} \sum_{(i,j)} (||R_{ij} - R_{ij}^\text{GT}||_F + ||t_{ij}^\text{GT} - t_{ij}||_2).
\]

We fine-tune the whole network for 2400 iterations using Adam optimizer [40] with a learning rate of $5 \times 10^{-6}$.

**5. Experiments**

We conduct the evaluation of our approach on the publicly available benchmark datasets 3DMatch [70], Redwood [14] and ScanNet [18]. First, we evaluate the performance, efficiency, and the generalization capacity of the proposed pairwise registration algorithm on 3DMatch and

---

Table 1. Registration recall on 3DMatch data set. 1-iter and 4-iter denote the result of the pairwise registration network and input to the 4th Trasnsl-Sync layer, respectively. Best results, except for 4-iter that is informed by the global information, are shown in bold.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Kitchen</th>
<th>Home 1</th>
<th>Home 2</th>
<th>Hotel 1</th>
<th>Hotel 2</th>
<th>Hotel 3</th>
<th>Study</th>
<th>MIT Lab</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.85</td>
<td>0.69</td>
<td>0.58</td>
<td>0.79</td>
<td>0.70</td>
<td>0.58</td>
<td>0.65</td>
<td>0.54</td>
</tr>
<tr>
<td>3DMatch</td>
<td>0.79</td>
<td>0.77</td>
<td>0.71</td>
<td>0.73</td>
<td>0.71</td>
<td>0.78</td>
<td>0.68</td>
<td>0.74</td>
</tr>
<tr>
<td>3DR</td>
<td>0.79</td>
<td>0.77</td>
<td>0.71</td>
<td>0.73</td>
<td>0.71</td>
<td>0.78</td>
<td>0.68</td>
<td>0.74</td>
</tr>
<tr>
<td>3DSN</td>
<td>0.79</td>
<td>0.77</td>
<td>0.71</td>
<td>0.73</td>
<td>0.71</td>
<td>0.78</td>
<td>0.68</td>
<td>0.74</td>
</tr>
<tr>
<td>FCGF</td>
<td>0.79</td>
<td>0.77</td>
<td>0.71</td>
<td>0.73</td>
<td>0.71</td>
<td>0.78</td>
<td>0.68</td>
<td>0.74</td>
</tr>
<tr>
<td>Ours</td>
<td>0.85</td>
<td>0.69</td>
<td>0.58</td>
<td>0.79</td>
<td>0.70</td>
<td>0.58</td>
<td>0.65</td>
<td>0.54</td>
</tr>
</tbody>
</table>

---

Table 2. Average run-time for estimating the pairwise transformation parameters of one fragment pair on 3DMatch dataset. Note, the GPU implementation of the soft assignments is faster than the CPU based kd-tree NN search.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>NN search</th>
<th>Model estimation</th>
<th>Total time</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIT</td>
<td>0.10</td>
<td>0.01</td>
<td>0.11</td>
</tr>
<tr>
<td>Ours (softNN)</td>
<td>0.08</td>
<td>0.07</td>
<td>0.15</td>
</tr>
</tbody>
</table>

---

Redwood dataset respectively (Sec. 5.1). We then evaluate the whole pipeline on the global registration of the point cloud fragments generated from RGB-D images, which are part of the ScanNet dataset [18].

**5.1. Pairwise registration performance**

We begin by evaluating the pairwise registration part of our algorithm on a traditional geometric registration task. We compare the results of our method to the state-of-the-art data-driven feature descriptors 3DMatch [70], CGF [39], PPFNet [21], 3DSmoothNet (3DS) [28], and FCGF [16], which is also used as part of our algorithm, as well as to a recent network based registration algorithm 3DR [22]. Following the evaluation procedure of 3DMatch [70] we complement all the descriptor based methods with the RANSAC-based transformation parameter estimation. For our approach we report the results after the pairwise registration network (1-iter in Tab. 1) as well as the output of the $\psi_{\text{iter}}$ in the 4th iteration (4-iter in Tab. 1). The latter is already informed with the global information and serves primarily as verification that with the iterations our input to the Transl-Sync layer improves. Consistent with the 3DMatch evaluation procedure, we report the average recall per scene as well as for the whole dataset in Tab. 1.

The registration results show that our approach reaches the highest recall among all the evaluated methods. More importantly, it indicates that using the same features (FCGF), our method can outperform RANSAC-based estimation of the transformation parameters, while having a much lower time complexity (Tab. 2). The comparison of the results of 1-iter and 4-iter also confirms the intuition that feeding the residuals and weights of the previous estimation back to the pairwise registration block helps refining the estimated pairwise transformation parameters.

**Generalization to other domains** In order to test if our pairwise registration model can generalize to new datasets and unseen domains, we perform a generalization evaluation on a synthetic indoor dataset Redwood indoor [14]. We follow the evaluation protocol of [14] and report the average registration recall and precision across all four scenes.

We compare our approach to the recent data driven approaches 3DMatch [70], CGF [39]+FGR [72] or CZK [14], RelativeNet (RN) [22], 3DR [22] and traditional methods CZK [14] and Latent RANSAC (LR) [41]. Table 3 shows...
that our approach can achieve \( \approx 4 \) percentage points higher recall than state-of-the-art without being trained on synthetic data and thus confirming the good generalization capacity of our approach. Note that while the average precision across the scenes is low for all the methods, several works [14, 39, 22] show that the precision can easily be increased using pruning without almost any loss in the recall.

### 5.2. Multiview registration performance

We finally evaluate the performance of our complete method on the task of multiview registration using the ScanNet [18] dataset. ScanNet is a large RGBD dataset of indoor scenes. It provides the reconstructions, ground truth camera poses and semantic segmentations for 1513 scenes. To ensure a fair comparison, we follow [36] and use the same 32 randomly sampled scenes for evaluation. For each scene we randomly sample 30 RGBD images that are 20 frames apart and convert them to point clouds. The temporal sequence of the frames is discarded. In combination with the large temporal gap between the frames, this makes the test setting extremely challenging. Different to [36], we do not train our network on ScanNet, but rather perform direct generalization of the network trained on the 3DMatch dataset.

**Evaluation protocol** We use the standard evaluation protocol [13, 36] and report the empirical cumulative distribution function (ECDF) for the angular \( a_e \) and translation \( t_e \) deviations defined as

\[
a_e = \arccos\left(\frac{\langle R_{ij}^T R_{ij}^{GT} \rangle - 1}{2}\right) \quad t_e = ||t_{ij}^{GT} - t_{ij}^*||_2
\]

The ground truth rotations \( R_{ij}^{GT} \) and translations \( t_{ij}^{GT} \) are provided by the authors of ScanNet [18]. In Tab. 3 we report the results for three different scenarios. "FGR (Good)" and "Ours (Good)" denote the scenarios in which we follow [36] and use the computed pairwise registrations to prune the edges before the transformation synchronization if the median point distance in the overlapping\(^4\) region after the transformation is larger than 0.1m (FGR) or 0.05m (ours). The EIGSE3 in "Ours (Good)" is initialized using our pairwise estimates. On the other hand, "all" denotes the scenario in which all \( \binom{N_2}{2} \) pairs are used to build the graph. In all scenarios we prune the edges of the graph if the confidence estimation in the relative transformation parameters of that edge \( c_{ij}^{local} \) drops below \( \tau_p = 0.85 \). This threshold was determined on 3DMatch dataset and its effect on the performance of our approach is analyzed in detail in the supp. material. If during the iterations the pruning of the edges yields a disconnected graph we simply report the last valid values for each node before the graph becomes disconnected. A more sophisticated handling of the edge pruning and disconnected graphs is left for future work.

**Analysis of the results** As shown in Tab. 3 our approach can achieve a large improvement on the multiview registration tasks when compared to the baselines. Not only are the initial pairwise relative transformation parameters estimated using our approach more accurate than the ones of FGR [72], but they can also be further improved in the subsequent iterations. This clearly confirms the benefit of the feed-back loop of our algorithm. Furthermore even when directly considering all input edges our approach still proves dominant, even when considering the results of the scenario "Good" for our competitors. More qualitative results of the multiview registration evaluation, including the failure cases, are available in the supp. material.

**Computational complexity** Low computational costs of pairwise and multiview registration is important for various fields like augmented reality or robotics. We first compare computation time of our pairwise registration component to RANSAC. In Tab. 2 we report the average time needed to register one fragment pair of the 3DMatch dataset as well

\(^4\)The overlapping regions are defined as parts, where after transformation, the points are less than 0.2m away from the other point cloud. [36]
as one whole scene. All timings were performed on a standalone computer with Intel(R) Core(TM) i7-7700K CPU @ 4.20GHz, GeForce GTX 1080, and 32 GB RAM. Average time of performing softNN for a fragment pair is about 0.1s, which is a approximately four times faster than traditional nearest neighbor search (implemented using scikit-learn [49]). An even larger speedup (about 23 times) is gained in the model estimation stage, where our approach requires a single forward pass (constant time) compared to up to 50000 iterations of RANSAC when the inlier ratio is 5% and the desired confidence 0.99\(^5\). This results in an overall run-time of about 80s for our entire multiview approach (including the feature extraction and transformation synchronization) for the Kitchen scene with 1770 fragment pairs. In contrast, feature extraction and pairwise estimation of transformation parameters with RANSAC takes > 1100s. This clearly shows the efficiency of our method, being > 13 times faster to compute (for a scene with 60 fragments).

5.3. Ablation study

To get a better intuition how much the individual novelties in our approach contribute to the final performance, we carry out an ablation study on the ScanNet [18] dataset. In particular, we analyze the proposed edge pruning scheme based on the confidence estimation block and Cauchy function as well as the impact of the iterative refinement of the relative transformation parameters.\(^6\) The results of the ablation study are presented in Fig. 4.

Benefit from the iterative refinement We motivate the iterative refinement of the transformation parameters that are input to the Transf-Sync layer with a notion that the weights and residuals provide additional cues for their estimation. Results in Fig. 4 confirm this assumption. The input relative parameters in the 4-th iteration are approximately 2 percentage points better that the initial estimate. On the other hand, Fig. 4 shows that at the high presence of outliers or inefficient edge pruning (see e.g., the results w/o edge pruning) the weights and the residuals actually provide a negative bias and worsen the results.

Edge pruning scheme There are several possible ways to implement the pruning of the presumable outlier edges. In our experiments we prune the edges based on the output of the confidence estimation block (w-conf.). Other options are to realize this step using the global confidence, i.e. the Cauchy weights defined in (14) (w-Cau.) or not performing this at all (w/o). Fig. 4 clearly shows the advantage of using our confidence estimation block (gain of more than 20 percentage points). Even more, due to preserving a large amount of outliers, alternative approaches perform even worse than the pairwise registration.

6. Conclusions

We have introduced an end-to-end learnable, multiview point cloud registration algorithm. Our method departs from the common two-stage approach and directly learns to register all views in a globally consistent manner. We augment the 3D descriptor FCGF [16] by a soft correspondence layer that pairs all the scans to compute initial matches, which are fed to a differentiable pairwise registration block resulting in transformation parameters as well as weights. A pose graph is constructed and a novel, differentiable iterative transformation synchronization layer globally refines weights and transformations. Experimental evaluation on common benchmark datasets show that our method outperforms state-of-the-art by more than 25 percentage points on average regarding the rotation error statistics. Moreover, our approach is > 13 times faster than RANSAC-based methods in a multiview setting of 60 scans, and generalizes better to new scenes (≈ 4 percentage points higher recall on Redwood indoor compared to state-of-the-art).

Acknowledgements. This work is partially supported by Stanford-Ford Alliance, NSF grant IIS-1763268, Vannevar Bush Faculty Fellowship, Samsung GRO program and the Stanford SAIL Toyota Research Center. We thank NVIDIA Corp. for providing the GPUs used in this work.

\(^5\)We use the CPU-based RANSAC implementation that is provided in the original evaluation code of 3DMatch dataset [70].

\(^6\)Additional results of the ablation study are included in the supp. material.
References

[1] Dror Aiger, Niloy J Mitra, and Daniel Cohen-Or. 4-points congruent sets for robust pairwise surface registration. In *ACM transactions on graphics (TOG)*, number 3, 2008. 2


A. Supplementary Material

B. Supplementary material

In this supplementary material, we provide additional information about the proposed algorithm (Sec. B.1-B.2 and Alg. 1), network architectures and training configurations (Sec. B.3), an extended ablation study (Sec. B.5) as well as additional visualizations (Sec. B.7). The source code and pretrained models are publicly available under https://github.com/zgojcic/3D_multiview_reg.

B.1. Closed-form solution of Eq. 4.

For the sake of completeness we summarize the closed-form differentiable solution of the weighted least square pairwise registration problem

\[ \hat{R}_{ij}, \hat{t}_{ij} = \arg \min_{R_{ij}, t_{ij}} \sum_{l=1}^{N} w_l \| R_{ij} \mathbf{p}_l + t_{ij} - \mathbf{q}_l \|^2. \]  

(22)

Let \( \mathbf{p} \) and \( \mathbf{q} \)

\[
\mathbf{p} := \frac{\sum_{l=1}^{N_p} w_l \mathbf{p}_l}{\sum_{l=1}^{N_p} w_l}, \quad \mathbf{q} := \frac{\sum_{l=1}^{N_q} w_l \mathbf{q}_l}{\sum_{l=1}^{N_q} w_l}
\]

(23)
denote weighted centroids of point clouds \( \mathbf{P} \in \mathbb{R}^{N \times 3} \) and \( \mathbf{Q} \in \mathbb{R}^{N \times 3} \), respectively. The centered point coordinates can then be computed as

\[
\tilde{\mathbf{p}}_l := \mathbf{p}_l - \mathbf{p}, \quad \tilde{\mathbf{q}}_l := \mathbf{q}_l - \mathbf{q}, \quad l = 1, \ldots, N
\]

(24)

Arranging the centered points back to the matrix forms \( \tilde{\mathbf{P}} \in \mathbb{R}^{N \times 3} \) and \( \tilde{\mathbf{Q}} \in \mathbb{R}^{N \times 3} \), a weighted covariance matrix \( \mathbf{S} \in \mathbb{R}^{3 \times 3} \) can be computed as

\[ \mathbf{S} = \tilde{\mathbf{P}}^T \tilde{\mathbf{Q}} \]

(25)

where \( \mathbf{W} = \text{diag}(w_1, \ldots, w_N) \). Considering the singular value decomposition \( \mathbf{S} = \mathbf{U} \Sigma \mathbf{V}^T \) the solution to Eq. 22 is given by

\[
\hat{R}_{ij} = \mathbf{V} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \text{det}(\mathbf{VU}^T) \end{bmatrix} U^T
\]

(26)

where \( \text{det}(\cdot) \) denotes computing the determinant and is used here to avoid creating a reflection matrix. Finally, \( \hat{t}_{ij} \) is computed as

\[ \hat{t}_{ij} = \mathbf{q} - \hat{R}_{ij} \mathbf{p} \]

(27)

B.2. Closed-form solution of Eq. 5 and 6

In this section we summarize the closed form solutions to Eq. 5 and 6 from the main paper describing the rotation and translation synchronization, respectively.

The least squares formulation of the rotation synchronization problem

\[ \hat{R}_i^* = \arg \min_{R_i \in SO(3)} \sum_{(i,j) \in E} c_{ij} \| R_{ij} - R_i R_j^T \|^2_F \]

(28)

admits a closed form solution under spectral relaxation as follows [2, 3]. Consider a symmetric matrix \( \mathbf{L} \in \mathbb{R}^{N_\Theta \times N_\Theta} \) resembling a block Laplacian matrix, defined as

\[ \mathbf{L} = \mathbf{D} - \mathbf{A} \]

(29)

where \( \mathbf{D} \) is the weighted degree matrix constructed as

\[
\mathbf{D} = \begin{bmatrix}
I_3 \sum_i c_{i1} & I_3 \sum_i c_{i2} & \ldots \\
I_3 \sum_i c_{i1} & I_3 \sum_i c_{i2} & \ldots \\
& & \ddots \\
& & & I_3 \sum_i c_{iN_\Theta} \\
\end{bmatrix}
\]

(30)
and $A$ is a block matrix of the relative rotations

$$A = \begin{bmatrix}
0_3 & c_2R_{12} & \cdots & c_{1N_S}R_{1N_S} \\
c_{21}R_{21} & 0_3 & \cdots & c_{2N_S}R_{2N_S} \\
\vdots & \vdots & \ddots & \vdots \\
c_{N_S1}R_{N_S1} & c_{N_S2}R_{N_S2} & \cdots & 0_3
\end{bmatrix} \tag{31}$$

where the weights $c_{ij} := \text{ci}m(\Gamma)$ represent the confidence in the relative transformation parameters $M_{ij}$ and $N_S$ denotes the number of nodes in the graph. The least squares estimates of the global rotation matrices $R_i^z$ are then given, under relaxed orthonormality and determinant constraints, by the three eigenvectors $v_1 \in \mathbb{R}^{3N_S}$ corresponding to the smallest eigenvalues of $L$. Consequently, the nearest rotation matrices under Frobenius norm can be obtained by a projection of the $3 \times 3$ submatrices of $V = [v_1, v_2, v_3] \in \mathbb{R}^{3N_S \times 3}$ onto the orthonormal matrices and enforcing the determinant $\det(R_i^z) = 1$ to avoid the reflections.

Similarly, the closed-form solution to the least squares formulation of the translation synchronization

$$t_i^* = \arg \min_{t_i} \sum_{(i,j) \in E} c_{ij} \| \hat{R}_{ij}t_i + \hat{t}_{ij} - t_j \|^2 \tag{32}$$

can be written as

$$t^* = L^+ b \tag{33}$$

where $t^* = [t_1^T, \ldots, t_{N^S}^T]^T \in \mathbb{R}^{3N_S}$ and $b = [b_1^T, \ldots, b_{N^S}^T]^T \in \mathbb{R}^{3N^S}$ with

$$b_i = - \sum_{j \in N(i)} c_{ij} \hat{R}_{ij}^T \hat{t}_{ij}. \tag{34}$$

where $N(i)$ denotes all the neighboring vertices of $S_i$ in graph $G$.

**B.3. Network architecture and training details**

This section describes the network architecture as well as the training details of the FCGF [16] feature descriptor (Sec. B.3.1) and the proposed registration block (Sec. B.3.2). Both networks are implemented in Pytorch and pretrained using the 3DMatch dataset [70].

**B.3.1 FCGF local feature descriptor**

**Network architecture** The FCGF [16] feature descriptor operates on sparse tensors that represent a point cloud in form of a set of unique coordinates $C$ and their associated features $F$

$$C = \begin{bmatrix} x_1 & y_1 & z_1 & b_1 \\ \vdots & \vdots & \vdots & \vdots \\ x_N & y_N & z_N & b_N \end{bmatrix}, \quad F = \begin{bmatrix} f_1 \\ \vdots \\ f_M \end{bmatrix} \tag{35}$$

where $x_i, y_i, z_i$ are the coordinates of the $i$-th point in the point cloud and $f_i$ is the associated feature (in our case simply 1). FCGF is implemented using the Minkowski Engine, an auto-differentiation library, which provides support for sparse convolutions and implements all essential deep learning layers [15]. We adopt the original, fully convolutional network design of FCGF that is depicted in Fig. 5. It has a U-Net structure [53] and utilizes skip connections and ResNet blocks [32] to extract the per-point $32 \times 32 \times 32$ dim feature descriptors. To obtain the unique coordinates $C$, we use a GPU implementation of the voxel grid downsampling [15] with the voxel size $v := 2.5 \text{ cm}$.

**Training details** We again follow [16] and pre-train FCGF for 100 epochs using the point cloud fragments from the 3DMatch dataset [70]. We optimize the parameters of the network using stochastic gradient descent with a batch size 4 and an initial learning rate of 0.1 combined with an exponential decay with $\gamma = 0.99$. To introduce rotation invariance of the descriptors we perform a data augmentation by randomly rotating each of the fragments along an arbitrary direction, by a different rotation, sampled from the $[0^\circ, 360^\circ]$ interval. The sampling of the positive and negative examples follows the procedure proposed in [16].

**B.3.2 Registration block**

**Network architecture** The architecture of the registration block (same for $\psi_{\text{inl}}(\cdot)$ and $\psi_{\text{out}}(\cdot)$

where $\psi_{\text{inl}}(\cdot)$ and $\psi_{\text{out}}(\cdot)$ are dimensionality reduction functions) follows [71] and is based on the PointNet-like architecture [51] where each of the fully connected layers ($\mathbf{P}$ in Fig. 6) operates on individual correspondences. The local context is then aggregated using the instance normalization layers [64] defined as

$$y_{i}^l = \frac{x_{i}^l - \mu_{i}^l}{\sigma_{i}} \tag{36}$$

where $x_{i}^l$ is the output of the layer $l$ and $\mu_{i}^l$ and $\sigma_{i}$ are per dimension mean value and standard deviation, respectively. Opposed to the more commonly used batch normalization, instance normalization operates on individual training examples and not on the whole batch. Additionally, to retrieve the local context, the order-aware blocks [71] are used to map the correspondences to clusters using the learned soft pooling $S_{\text{pool}} \in \mathbb{R}^{N_c \times M_c}$ and unpooling $S_{\text{unpool}} \in \mathbb{R}^{N_c \times M_c}$ operators as

$$X_{k+1} = S_{\text{pool}}X_k \quad \text{and} \quad X_{k+1}' = S_{\text{unpool}}X_{k+1}' \tag{37}$$

where $N_c$ is the number of correspondences and $M_c$ is the number of clusters. $X_k$ and $X_{k+1}$ are the features at the level $k$ (before clustering) and $k + 1$ (after clustering), respectively (see Fig. 6). Finally, $X_{k+1}'$ denotes the output of the last layer in the level $k + 1$.

**Training details** We pre-train the registration blocks using the same fragments from the 3DMatch dataset. Specifically, we first infer the FCGF descriptors and randomly sample $N_c = 5000$ descriptors per fragment. We use these descriptors to compute the putative correspondences for all fragment pairs $(i, j)$ such that $i \leq j$. Based on the ground truth transformation parameters, we label these correspondences as inliers if the Euclidean distance between the points after the transformation is smaller than 7.5 cm. At the start of the training (first 15000 iterations) we supervise the learning using only the binary cross-entropy loss. Once a meaningful number of correspondences can already be classified correctly we add the transformation loss. We train the network for 500k iterations using Adam [40] optimizer with the initial learning rate of 0.001. We decay the learning rate every 1000 iterations by multiplying it with 0.999. To learn the rotation invariance we perform data augmentation, starting from the 25000th iteration, by randomly sampling an angle from the interval $[0^\circ, \text{na} \cdot 20^\circ]$ where na is initialized with zero and is then increased by 1 every 5000 iteration until the interval becomes $[0^\circ, 360^\circ]$. 

---

For $\psi_{\text{inl}}(\cdot)$ the input dimension is increased from 6 to 8 (weights and residuals added).
Figure 6. Network architecture of the registration block consists of two main modules: i) a PointNet-like ResNet block with instance normalization, and ii) an order-aware block. For each point cloud pair, putative correspondences are feed into three consecutive ResNet blocks followed by a differentiable pooling layer, which maps the \( N_c \) putative correspondences to \( M_c \) clusters \( X_{k+1} \) at the level \( k+1 \). These serve as input to the three order-aware blocks. Their output \( X_{k+1} \) is fed along with \( X_k \) into the differentiable unpooling layer. The recovered features are then used as input to the remaining three ResNet blocks. The output of the registration block are the scores indicating whether the putative correspondence is an outlier or an inlier. Additionally, the 128-dim features (denoted as \( X^{\text{conf}} \)) before the last perceptron layer \( P \) are used as input to the confidence estimation block.

### B.4. Pseudo-code

Alg. 1 shows the pseudo-code of our proposed approach. We iterate \( k = 4 \) times over the network and transformation synchronization (i.e. Transf-Sync) layers and in each of those iterations we execute the Transf-Sync layer four times. Our implementation is constructed in a modular way (each part can be run on its own) and can accept a varying number of input point clouds with or without the connectivity information.

### B.5. Extended ablation study

We extend the ablation study presented in the main paper, by analyzing the impact of edge pruning based on the local confidence (i.e. the output of the confidence estimation block) (Sec. B.5.1) and of the weighting scheme (Sec. B.6) on the angular and translation errors. The ablation study is performed on the point cloud fragments of the ScanNet dataset [18].

#### B.5.1 Impact of the edge pruning threshold

Results depicted in Fig. 7 show that the threshold value used for edge pruning has little impact on the angular and translation errors as long as it is larger than 0.2.

#### B.6. Impact of the harmonic mean weighting scheme

In this work, we have introduced a scheme for combining the local and global confidence using the harmonic mean (HM). In the following, we perform the analysis of this proposal and compare its performance to established methods based only on global information [4]. To this end, we again consider the scenario “Ours (Good)” as the input graph connectivity information. We compare the results of the proposed scheme (HM) to SE3 EIG [4], which proposes using the Cauchy function for computing the global edge confidence [4]. Note, we use the same pairwise transformation parameters, estimated using the method proposed herein, for all methods.

Without edge pruning It turns out that combining the local and global evidence about the graph connectivity is essential to achieve good performance. In fact, merely relying on local confidence estimates without HM weighting (denoted as ours; green) in Fig. 8) the Transf-Sync is unable to recover global transformations from the given graph connectivity evidence that is very noisy. Introducing the HM weighting scheme allows us to reduce the impact of noisy graph connectivity built solely using local confidence and can significantly improve performance after Transf-Sync.

---

**Algorithm 1** Pseudo-code of the proposed approach

Input: a set of potentially overlapping scans \( \{S_i\}_{i=1}^{N_S} \)

Output: globally optimized poses \( \{M_i^*\}_{i=1}^{N_S} \)

# Compute the pairwise transformations

for each pair of scans \( S_i, S_j \subset S \), \( i \neq j \) do

# find the putative correspondences using \( \phi(\cdot) \)

\(- X_{ij} = \text{cat}([S_i, \phi(S_i, S_j)]) \in \mathbb{R}^{N_{S_j} \times 6} \)

# compute the weights \( w_{ij} \in \mathbb{R}^{N_{S_j}} \)

\(- w_{ij} = \psi_{\text{init}}(X_{ij}) \in \mathbb{R}^{N_{S_i}} \)

# calculate \( R_{ij}, t_{ij} \) using SVD according to (4)

# Iterative network for transformation synchronization

\( X_{ij}^{(0)} \leftarrow X_{ij}, w_{ij}^{(0)} \leftarrow w_{ij}, r_{ij}^{(0)} \leftarrow r_{ij} \)

for \( k = 1, 2, \ldots, \text{max iters} \) do

for each pairwise output from \( \psi_{\text{init}} \) do

\(- R_{ij}^{(k)}, t_{ij}^{(k)}, w_{ij}^{(k)} = \psi_{\text{iter}}([X_{ij}^{(k-1)}, w_{ij}^{(k-1)}, r_{ij}^{(k-1)}) \)

# estimate local \( \{c_{ij}^{(k)}\} \) using (16)

# Gather the pairwise estimation as \( R^{(k)}, t^{(k)} \)

# Build the graph and perform the synchronization

if \( k = 1 \) then

\(- c^{(k)} := \text{local}(c^{(k)}) \)

else

\(- c^{(k)} := f_{HM}(\text{local}(c^{(k)}), \text{global}(c^{(k-1)})) \)

\(- R^{(k)}, t^{(k)} = \text{Transf-Sync}(R^{(k)}, t^{(k)}, c^{(k)}) \)

# update step

for each pair of scans \( S_i, S_j \subset S \), \( i \neq j \) do

\(- X_{ij}^{(k+1)} = \text{cat}([S_i, M_{ij}^{(k)} \otimes \phi(S_i, S_j)]) \)

\(- w_{ij}^{(k+1)} = w_{ij}^{(k)} \)

\(- r_{ij}^{(k+1)} = \|S_i - M_{ij}^{(k)} \otimes \phi(S_i, S_j)\|_2 \)
block, which in turn enables us to outperform the SE3 EIG.

**With edge pruning** Fig. 9 shows that pruning the edges can help coping with noisy input graph connectivity built from the pairwise input. In principal, suppression of the edges with low confidence results in discarding the outliers that corrupt the l2 solution and as a result improves the performance of the Transf-Sync block.

### B.7. Qualitative results

We provide some additional qualitative results in form of success and failure cases on selected scenes of 3DMatch (Fig. 10 and 11) and ScanNet (Fig. 12 and 13) datasets. Specifically, we compare the results of our whole pipeline Ours (After Sync.) to the results of SE3 EIG [4], pairwise registration results of our method from the first iteration Ours (1\textsuperscript{st} iter.), and pairwise registration results of our method from the fourth iteration Ours (4\textsuperscript{th} iter.). Both global methods (Ours (After Sync.) and SE3 EIG) use transformation parameters estimated by our proposed pairwise registration algorithm as input to the transformation synchronization. The failure cases of our method predominantly occur on point clouds with low level of structure (planar areas in Fig. 11 bottom) or high level of symmetry and repetitive structures (Fig. 13 top and bottom, respectively).
Figure 10. Selected success cases of our method on 3DMatch dataset. Top: Kitchen and bottom: Hotel 1. Red rectangles highlight interesting areas with subtle changes.
Figure 11. Selected failure cases of our method on 3DMatch dataset. Top: Home 1 and bottom: Home 2. Note that our method still provides qualitatively better results than state-of-the-art.
Figure 12. Selected success cases of our method on ScanNet dataset. Top: scene0057_01 and bottom: scene0309_00. Red rectangles highlight interesting areas with subtle changes.
Figure 13. Selected failure cases of our method on ScanNet dataset. Top: scene0334_02 and bottom: scene0493_01. Note that our method still provides qualitatively better results than state-of-the-art.