PoGO-Net: Pose Graph Optimization with Graph Neural Networks

Xinyi Li *
Magic Leap, Sunnyvale, CA, USA
xinli@magicleap.com

Haibin Ling †
Stony Brook University, Stony Brook, NY, USA
hling@cs.stonybrook.edu

Abstract

Accurate camera pose estimation or global camera relocalization is a core component in Structure-from-Motion (SfM) and SLAM systems. Given pair-wise relative camera poses, pose-graph optimization (PGO) involves solving for an optimized set of globally-consistent absolute camera poses. In this work, we propose a novel PGO scheme fueled by graph neural networks (GNN), namely PoGO-Net, to conduct the absolute camera pose regression leveraging multiple rotation averaging (MRA). Specifically, PoGO-Net takes a noisy view-graph as the input, where the nodes and edges are designed to encode the geometric constraints and local graph consistency. Besides, we address the outlier edge removal by exploiting an implicit edge-dropping scheme where the noisy or corrupted edges are effectively filtered out with parameterized networks. Furthermore, we introduce a joint loss function embedding MRA formulation such that the robust inference is capable of achieving real-time performances even for large-scale scenes. Our proposed network is trained end-to-end on public benchmarks, outperforming state-of-the-art approaches in extensive experiments that demonstrate the efficiency and robustness of our proposed network.

1. Introduction

Visual localization or camera pose regression lies in the heart of many computer vision and robotics tasks, with applications including robot navigation, autonomous driving and augmented reality. Camera pose estimation is the process of self-determining the orientation and position with the aid of sequential information via image retrieval. As the key component in standard camera pose estimation pipelines, pose-graph optimization (PGO) involves iterative estimations of pair-wise camera relative poses and the progressive optimization of the noisy global view-graph. In most of the conventional Structure-from-Motion (SfM) [65, 69] and SLAM [47] systems, PGO is conducted as numerically solving a high-dimensional non-convex approximation problem by leveraging feature-frame correspondences and often yields high computational costs.

Despite the proliferation of research addressing the back-end optimization in SfM systems, many challenges remain open. Firstly, canonical solvers carry a complexity of cubic order with regards to the input size and gradually slows down [67], forfeiting the real-time requirements. Secondly, measurements of pair-wise relative camera poses are often noisy, yielding corrupted and erroneous edges in the view-graph and henceforth impairing the performances of both conventional and learning-based methods [52]. Thirdly, direct regressions of structures and motions with deep learning networks are prone to overfitting [55, 62], hindering the robustness and generality in real-world applications.

Inspired by the recent successes of Graph Neural Networks (GNNs) [53], we herein propose a novel GNN-based PGO scheme to address all the aforementioned issues with a concrete network, namely, *PoGO-Net*. Specifically, we encode the edge messages with pair-wise geometric constraints on the edges of the view-graph, aggregated with the local consistency information. The absolute camera orientations are encoded as node features, updated according to its connected edges and neighboring nodes. As we consider the input as a corrupted graph with erroneous and redundant edges, we address the graph de-noising issue by exploiting topological parameterized network layers to conduct the 'edge dropping', *i.e.*, the outlier edges are removed according to the local graph consistency, resulting a sparser yet preciser sub-graph of the input view-graph. We redefine the message aggregation and design the loss function based on multiple rotation averaging (MRA) algorithm, with the efficient message passing scheme our proposed network is capable of processing in real-time speed even with large-scale datasets. Moreover, our network bares an end-to-end differentiable structure where the parameters of the de-noising layers and the GNN layers are jointly optimized during training.

Our contributions can be summarized as follows:

- We propose a novel PGO formulation fueled with a GNN to conduct the absolute camera pose regression by exploiting the MRA scheme.

*Work primarily conducted during graduate study at Temple Univ.
†Corresponding author.

- We design the de-noise layers to address the outlier edge removal in PGO. Our proposed de-noise layers are iteratively executed with the GNN layers, implicitly exploiting the 'edge-dropping' scheme.
- We train PoGO-Net end-to-end and the network can be easily integrated with both conventional and learning-based SfM systems*. Extensive experiments on public benchmarks demonstrates the accuracy, efficiency and robustness of our proposed network.

2. Related Work

Conventional PGO approaches. Given a 3D scene, pairwise relative camera poses are initially estimated by applying robust methods [21, 50] to reject the matched feature correspondence outliers and thus fits the essential/fundamental matrix [2], followed by the view-graph refinement, *i.e.*, PGO iterations. In standard PGO pipelines of conventional SfM approaches [19, 35, 47, 56], solving the high-dimensional non-convex optimization problem [27, 58] mostly involves adopting iterative non-linear numerical solvers [1, 45, 48, 64] to minimize the reprojection errors with jointly optimizing the 3D scene points, camera orientations and translations [42, 58, 68], namely, bundle adjustment (BA).

As a sub-problem in BA, *rotation averaging* (RA) [26, 29] devotes to solve for the camera orientations given a set of noisy measurements of the relative camera rotations and can be categorized into single rotation averaging [28, 38, 40] and *multiple rotation averaging* (MRA) [4, 7, 20, 44]. The former delivers the optimal solution of one rotation given several estimates whereas the latter can be considered as a synchronization problem with the goal to recover unknown vertex labellings in the graph given noisy edge labellings [3]. In recent years, we have witnessed a surge of research interests on MRA [9–12, 46, 63, 66]. Though MRA is still a computationally difficult problem to solve due to its non-convexity of the rotation group space, it shows the advantages by admitting a lower dimension and complexity compared with conventional BA approaches based on point-frame correspondences [11, 17, 66], enabling faster and lighter solvers. However, the predominant challenge of MRA is associated with the outlier edges, *i.e.*, the accuracy and robustness of MRA is tremendously impaired without the knowledge of the noise distribution over the edges in the view-graph [4, 12, 44, 65]. There have been plentiful recent lines of work toward robust and efficient MRA approaches, which can be further categorized into explicit outlier detection/removal schemes [12, 29, 49] and implicit noise reduction schemes [4, 14, 63].

Learning-based SfM approaches. It was not until recently that research interests focus on incorporating deep neural networks into SfM pipelines and camera pose regression tasks [5, 18, 22, 33, 36, 57, 61, 71]. As one of the earliest work adopting neural networks for camera pose regression, the deep convolutional neural network pose regressor proposed in [33] is trained according to a loss function embedding the absolute camera pose prediction error. While [33] pioneers in fusing the power of neural networks into pose regression frameworks, it does not take the intra-frame constraints or connectivity of the view-graph into optimization and thus barely over-performs conventional counterparts on the accuracy, as improved later in [13, 52, 72]. Other work exploits the algebraic or geometric relations among the given sequential images and train the networks to predict to locate the images [8, 13, 59, 61], among which [13] leverages temporal consistency of the sequential images by equipping bi-directional LSTMs with a CNN-RNN model such that temporal regularity can provide more pose information in the regression. The approach in [8] trains DNNs model with the pair-wise geometric constraints between frames, by leveraging additional measurements from IMU and GPS. Adoption of neural networks also greatly benefits parallel line of studies including 3D registration and point cloud alignment [6, 25].

Recent work [72] is the first study to leverage GNNs in a full absolute camera pose regression framework, where the authors model the view-graph with nodes fused with image features extracted by CNNs. Another recent approach [49] proposes a GNN-based network to address MRA, where the network consists of two sub-networks addressing outlier removal and pose refinement respetively. Though these two GNN-based approaches both achieve satisfactory performance, limitations exist and improvements can be made. For example, the correlation of node features and edge values are treated as purely binary in [72], discarding geometric constraints between frames. Also, the graph is initialized to be fully connected, which might introduce large amounts of redundant and erroneous edges.

In our work, we encode the edge messages with pairwise geometric constraints on the edges of the view-graph, aggregated with the local consistency information. Though inspired by NeuRoRA [49], the proposed network enables the 'edge dropping' scheme by the explicit formulation of the edge message, while the former conducts message aggregation solely on nodes. Moreover, the graph information is preserved more efficiently by allowing node-edge joint message aggregation such that only one single loss is required, thus facilitating the end-to-end training, whereas the additional viewgraph cleaning loss is involved in the network design of NeuRoRA. Especially, we address the robustness of our proposed network by introducing de-noise layers for the efficient outlier removal.

Graph Neural Networks. By virtue of its powerful yet agile data representation, GNNs [34, 53, 60] have achieved

*Code at https://github.com/xxylii/PoGO-Net

exceptional performances on numerous computer vision tasks. Despite their successes, straightforward adoptions of GNNs in solving PGO is not applicable due to GNN's vulnerability against noisy graphs [24, 43, 51, 70, 73]. In our work, we reduce the negative effects of outlier edges by adopting parameterized de-noising layers [41, 43, 51].

3. Problem Statement

3.1. Preliminaries and Notations

Given a 3D scene with n image frames, consider there exists a measurement $\widetilde{\mathbf{R}}_{ij} \in \mathbb{SO}(3)$ of relative rotation between frame I_i and I_j. Assume that in the ideal scenario where $\widetilde{\mathbf{R}}_{ij}$ is noise-free, then the absolute rotations $\mathbf{R}_i, \mathbf{R}_j \in \mathbb{SO}(3)$ of I_i and I_j satisfies $\widetilde{\mathbf{R}}_{ij} = \mathbf{R}_j\mathbf{R}_i^{-1}$. However, in practice the relative measurements are often noisy and contain outliers, the absolute camera orientation estimation is thus to seek a set of camera orientations which are globally consistent with the relative measurements, this process is called *multiple rotation averaging* (MRA).

Formally, MRA [12] is a transformation synchronization problem which involves minimizing a cost function that penalizes the discrepancy between the measurements of relative rotations $\widetilde{\mathbf{R}}_{ij}$ and $\mathbf{R}_j\mathbf{R}_i^{-1}$. That is, to solve the following objective function

$$\underset{\mathbf{R}_i,\mathbf{R}_j,1\leq i,j\leq n}{\arg\min} \sum_{(i,j)} \rho\big(d(\widetilde{\mathbf{R}}_{ij}, \mathbf{R}_j\mathbf{R}_i^{-1})\big), \tag{1}$$

where $\rho(\cdot)$ is a robust cost function and $d(\cdot,\cdot)$ is the distance metric. We adopt the quaternion parameterization and the corresponding metric [29] throughout the paper.

3.2. Pose-Graph Optimization

With the MRA problem defined above, now we are ready to formulate the PGO process. Let a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote the initial view-graph, where the vertex set $\mathcal{V} = \{v_i | i \leq n\}$ represents the set of the absolute camera orientations to be estimated, and the edge set $\mathcal{E} = \{(i,j)|v_i, v_j \in \mathcal{V}\}$ describes the availability of pair-wise measurements of relative camera orientations between image frames. In practice, the view-graph is often noisy regarding edges, preventing us to conduct MRA directly on $\mathcal{G}$. The reasons of $\mathcal{E}$ being noisy are two-fold: 1) In light of the existence of irreducible errors in the image retrieval (*e.g.* feature matching), outlier pair-wise relative measurements are hard to eliminate for both deep-learning based approaches [37, 49] and traditional geometry-constrained approaches [10, 44, 63]. 2) As multiple cameras can share similar views, the view-graph tends to have redundant edges such that MRA defined in Eq. 1 is often 'over-constrained' [12].

In our work, we handle the noise in the view-graph by exploiting an 'edge-dropping' scheme fused by parameterized de-noising layers, such that the noisy/redundant edges are remedied and eradicated, MRA is then veritably operated on the proper sub-graph of $\mathcal{G}$.

4. PoGO-Net Architecture

In this section, we detail the proposed PoGO-Net as shown in Fig. 1. Specifically, we first give the network architecture overview in §4.1, followed by the introduction of our graph structure and feature embedding in §4.2. We then illustrate the novel construction of our message aggregation scheme in §4.3, where the node messages and edge messages are both effectively encoded to gather all the information over the neighborhood of each node. §4.4 depicts the de-noising layers in our proposed network, where the de-noising layers are designed to be iteratively executed with GNN layers such that the outlier edges can be efficiently removed implicitly. In §4.5 and §4.6, we emphasize the graph update rules and the proposed loss function.

4.1. Architecture Overview

As shown in Fig. 1, our PoGO-Net takes noisy view-graphs as the input and output the optimized pose-graphs. Since the absolute camera orientations are unknown in the input, we initialize the node features by seeding a spanning tree at the node with the highest degree (*i.e.* connected with most nodes) and the initialization is propagated over the graph with the aid of our de-noise layers actively removing the outlier edges. The network has a multi-layer feed-forward architecture and consists of de-noise layers and GNN layers. At each iteration, the de-noise layer conducts the 'edge-dropping' scheme on the outlier edges before updating the aggregated messages through the GNN layer. PoGO-Net is fully differentiable and trained end-to-end to jointly optimize the de-noise layers and GNN layers.

4.2. Feature Embedding

For an input view-graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the edge set $\mathcal{E}$ representing the set of relative orientations contains most of the essential information required in the pose aggression. Let $\widetilde{r}_{ij} \in \mathbb{SO}(3), (i,j) \in \mathcal{E}$ represent the feature vector of the edge connecting v_i and v_j. Since the nodes represent the absolute camera orientations which are unknown, let $q_i \in \mathbb{SO}(3), v_i \in \mathcal{V}$ represent the node feature. $\{q_i | v_i \in \mathcal{V}\}$ can be deemed as a set of feature placeholders and is interactively initialized in a spanning-tree manner during the training process, more details are given in §4.5.

In contrast with regular GNNs where the adjacency matrix $\mathcal{A}_\mathcal{G}$ derived from $\mathcal{E}$ is a binary matrix indicating the neighborhood of each node, the adjacency matrix in our work is formed by parameterized variables. Specifically, values of elements consisting $\mathcal{A}_\mathcal{G}$ illustrate whether the corresponding edge-denoted measurements are reliable, *i.e.*, small values imply that the edges are prone to be noisy or even outliers. Details of parameterization of $\mathcal{A}_\mathcal{G}$ is in §4.4.

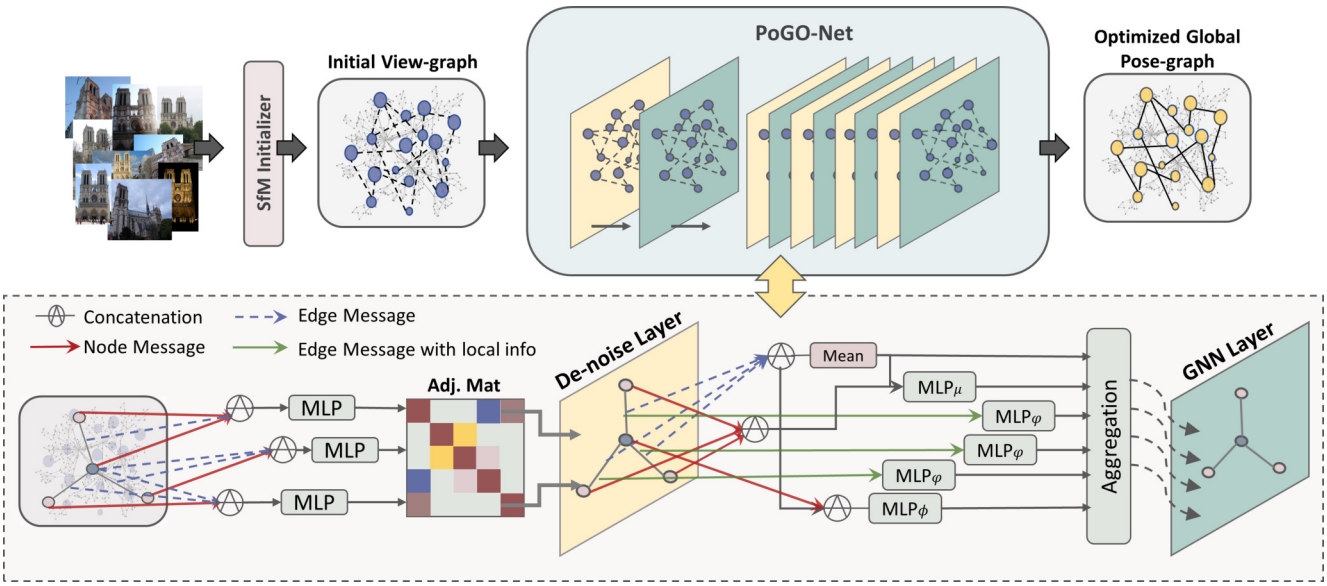


Figure 1: Illustration of the PoGO-Net pipeline. Our proposed network takes a noisy view-graph as the input and the output is the optimized pose-graph. The network adopts a multi-layer feed-forward architecture with the message passing scheme, where the message is aggregated over the connected edges and neighboring nodes of each node. The de-noising layer is designed to remove the outlier edges and is executed iteratively with GNN layers. Best viewed in color.

4.3. Message Aggregation

Our network adopts a multi-layer feed-forward architecture implemented with the message-passing scheme [53], *i.e.*, the aggregated information is propagated over the neighborhood of each node. Since the nodes and edges are interactively updated through network layers, we design a novel joint message aggregation scheme to effectively encode both the node messages and edge messages. In detail, denote $\mathcal{N}_i^l = \{v_j | (i,j) \in \mathcal{E}^l\}$ for the neighborhood of node v_i on the l^{th} layer, the messages are generated as follows

$$m_{q_i}^l = \rho\{\widetilde{r}_{ij} | (i,j) \in \mathcal{E}^l\} +\!\!+\, q_i^l, \tag{2}$$

$$m_{\widetilde{r}_{ij}}^l = q_i^l +\!\!+\, q_j^l +\!\!+\, \widetilde{r}_{ij}^l, \tag{3}$$

$$m_{\pi_i}^l = \text{mean}\{\widetilde{r}_{ij}^l | (i,j) \in \mathcal{E}^l\} +\!\!+\, \{q_j^l | v_j \in \mathcal{N}_i^l\}, \tag{4}$$

where $+\!\!+$ denotes the concatenation and π_i the state of node v_i. For PGO, gathering information from all the neighboring cameras sharing views with a given camera pose is essential, hence we assemble the state feature of v_i with all the connected edge and node features in its neighborhood.

It is noteworthy that, as our proposed network is capable of filtering out the outlier/redundant edges during the training, $\mathcal{E}$ is evolving as sparser yet preciser through different layers (details given in §4.4). The two components of node state message correspond to the all the connected edges an neighboring nodes.

4.4. Graph De-noising

As the input of PoGO-Net is often noisy with the presence of outlier/redundant edges, it is not practical to directly apply GNNs to the PGO task as the message aggregation along edges is likely to propagate and amplify the noise over the whole graph. In our proposed network, we reduce the noise by exploiting 'edge-dropping' de-noising layers along with the GNN layers, such that the edges and nodes are interactively updated according to the corresponding message passing defined in §4.3.

In detail, consider the adjacency matrix $\mathcal{A}_\mathcal{G}^l$ at the l^{th} layer of the network, in our network the elements of $\mathcal{A}_\mathcal{G}^l$ represent the weights of the corresponding edge features $\widetilde{r}_{ij}^l$ in the regression. That is, $\mathcal{A}_\mathcal{G}^l = \mathcal{A}_\mathcal{G} \odot \mathcal{Z}^l$, where $\mathcal{Z}^l$ denotes the binary coefficient matrix $\{z_{ij}^l\}$ and $\odot$ denotes the element-wise multiplication operation. Following [31, 43, 60], we relax the binary elements z_{ij}^l from being purely binary to values of a deterministic function g of the edge message $m_{\widetilde{r}_{ij}}^l$ as defined in Eq. 3, such that the coefficients are continuous and non-binary. Specifically, let ϵ^l be a uniformly distributed random variable independent with $m_{\widetilde{r}_{ij}}^l$, then z_{ij}^l is defined as

$$z_{ij}^l = g(\omega_{\gamma^l}(m_{\widetilde{r}_{ij}}^l), \epsilon^l), \tag{5}$$

where $\omega_{\gamma^l}(\cdot)$ is the MLP parameterized by γ^l. As we encourage the network to remove edges for the optimization, we extend the open domain $(0,1)$ of z_{ij}^l to include 0. Denote u_{ij}^l as the random variable drawn from the binary concrete distribution parametrized by the edge message, *i.e.*

$$u_{ij}^l = \sigma((\log \epsilon^l - \log(1-\epsilon^l) + \omega_{\gamma^l}(m_{\widetilde{r}_{ij}}^l))/\tau), \tag{6}$$

where $\tau > 0$ denotes the temperature parameter [31, 43] and $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function. Since we want

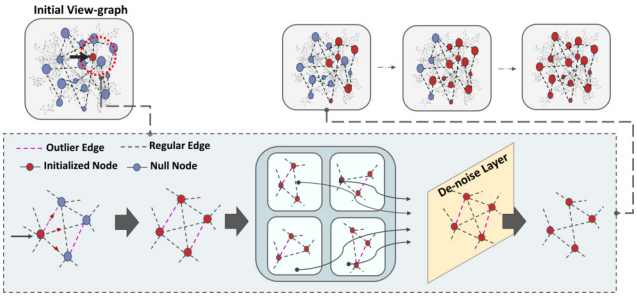


Figure 2: Node initialization. Our de-noising layers are capable of filtering out the outlier edges according to the local consistency during the spanning-tree based initialization, preventing erroneous measurements to be broadcast.

$u_{ij}^l \in (a, b)$ with $a < 0$ and $b > 0$, we thus update u_{ij}^l as $\hat{u}_{ij}^l = (b-a)u_{ij}^l + a$. Now we are ready to finalize z_{ij}^l as

$$z_{ij}^l = \min(1, \max(\hat{u}_{ij}^l, 0)), \tag{7}$$

such that the zero-valued coefficients are enabled.

With the de-noising scheme described above, now the noisy edges can be efficiently removed from the view-graph without explicit outlier detection. In our proposed network, the de-noising and message-passing are executed iteratively, *i.e.*, the input goes through the de-noise layer right before going through GNN layer in each iteration.

4.5. Graph Initialization and Updating

Initialization. Recall that the graph inauguration is equipped with the node set as the collection of node feature placeholders, as the absolute camera orientations are unknown in the input view-graph at the time of the initialization (§4.2). In PoGO-Net, we initialize the nodes by seeding a spanning tree in the view-graph [11, 28], *i.e.*, an initial value is given to the node with the highest degree, followed by the iterations of orientation broadcasting over its neighborhood in a breadth-first manner.

Despite that the initialization with spanning-tree rotation distribution is generally not robust for conventional approaches, as the outlier measurements on the noisy edges get propagated progressively [4, 12, 49], our proposed network is capable of correcting the erroneous measurements dynamically and thus restricting the outlier transmission, by virtue of the utilization of our de-noise layers. Specifically, the de-noise layers are parameterized with the edge messages, which assemble the information of the 'local edge consistency', *i.e.*, the outlier edges generate inconsistent messages within their neighborhoods, thus prone to be removed (§4.4). An illustration of our initialization process is given in Fig. 3.

Graph Update. The view-graph is updated regarding both edges and nodes through the network layers, while the node features are directly updated with reference to the aggregated node messages, the edge structure evolves implicitly regarding the emerging adjacency matrix. In detail, the edge features are aggregated in the edge messages along with the inherent graph connectivity information. In each iteration, outlier edges are dropped before the passing of the edge message aggregated over the updated local region.

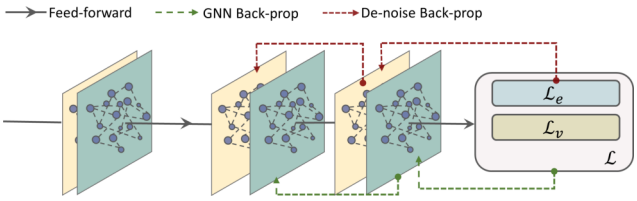


Figure 3: Illustration of back propagation scheme in our network. The de-noise layers are updated based on the edge loss while the GNN parameters are tuned by the total loss.

Formally, denote $\phi(\cdot), \varphi(\cdot)$ and $\mu(\cdot)$ as the differentiable MLPs for the concatenation of the nodes, edges and states, respectively, we update the graph according to the rules

$$q_i{}^{l+1} = \phi_i^l(\rho\{\widetilde{r}_{ij} | (i,j) \in \mathcal{E}^l\}, q_i^l, \pi_i{}^l), \tag{8}$$

$$\widetilde{r}_{ij}^{l+1} = \varphi_{ij}^l(q_i^l, q_j^l, \widetilde{r}_{ij}^l), \tag{9}$$

$$\pi_i{}^{l+1} = \mu_i^l(\text{softmax}\{\widetilde{r}_{ij}^l | (i,j) \in \mathcal{E}^l\}, \{q_j^l | v_j \in \mathcal{N}_i^l\}, \pi_i{}^l). \tag{10}$$

4.6. Loss Function

Loss Function. Our loss function consists of two components with one representing the edge loss and the other one representing the node loss. Intuitively, the edge loss measures the global consistency of the output pose-graph and the node loss evaluates the prediction of the absolute camera orientations. Respectively, denote $\mathcal{L}_e$ the edge loss and $\mathcal{L}_v$ the node loss, let $\mathcal{L}_r$ be the additional l_1 regularization loss corresponding to the weighted sum of node weights regarding the vertex degree along with the edge weights regarding the adjacency coefficients z_{ij}, then

$$\mathcal{L} = \alpha_e \mathcal{L}_e + \alpha_v \mathcal{L}_v + \alpha_r \mathcal{L}_r, \tag{11}$$

where $\alpha_e, \alpha_v, \alpha_r \in (0, 1)$ are the weight parameters. Precisely, denote the groundtruth absolute camera orientations as $\{q_i^*\}$, then we have

$$\mathcal{L}_e = \textstyle\sum_{(i,j)\in\hat{\mathcal{E}}} \|\hat{q}_j^{-1} \hat{r}_{ij} \hat{q}_i\|_d, \tag{12}$$

$$\mathcal{L}_v = \textstyle\sum_{i\in\mathcal{V}} \|\hat{q}_i - q_i^*\|_d, \tag{13}$$

where $\hat{(\cdot)}$ denotes the output variable values, $\|\cdot\|_d$ denotes the norm corresponding to the l_1 quaternion metric d.

Our network is trained jointly end-to-end with de-noise layers and GNN layers parameters optimized concurrently. Particularly, while the GNN layers are tuned with regard to the combined total loss, we enforce the de-noise layer training to be solely dependent on $\mathcal{L}_e$ since the 'edge-dropping' scheme designed for de-noising is based on the edge consistency of the local region.

Table 1: Experiment results on the 7Scenes Dataset [55]. Results are cited directly, the best results are highlighted.

Scene		RelocNet [5]	LsG [71]	MapNet [8]	MapNet+PGO [8]	PoseNet15 [33]	PoseNet17 [32]	PoseNet+LSTM [62]	CNN+GNN [72]	PoGO-Net
Chess	3m x 2m x 1m	4.14°	3.28°	3.25°	3.24°	8.12°	4.48°	5.77°	2.82°	**1.72°**
Office	2.5m x 2m x 1.5m	5.32°	5.45°	5.15°	5.42°	7.68°	5.55°	8.08°	5.08°	**3.93°**
Fire	2.5m x 1m x 1m	10.4°	10.92°	11.69°	9.29°	14.4°	11.30°	11.90°	8.94°	**6.23°**
Pumpkin	2.5m x 2m x 1m	4.17°	3.69°	4.02°	3.96°	8.42°	4.75°	7.00°	**2.77°°**	3.56°
Red Kitchen	4m x 3m x 1.5m	5.08°	4.92°	4.93°	4.94°	8.64°	5.35°	8.83°	4.48°	**3.85°**
Stairs	2.5m x 2m x 1.5m	**7.53°**	11.3°	12.08°	10.62°	13.8°	12.40°	13.70°	8.78°	7.88°
Heads	2m x 0.5m x 1m	10.5°	12.70°	13.25°	8.45°	12.0°	13.0°	13.7°	11.41°	**7.34°**
Average		6.73°	7.47°	7.66°	6.56°	10.4°	8.12°	9.85°	6.33°	**4.93°**

Training. For the training of PoGO-Net, we optimize the network parameters with SGD, where the weight decay is set to be 1e-4, and the learning rate is initialized as 1e-3. We train the network with batch size of 64, maximum epochs are set to be 300. In our experiments, we use parameters $\alpha_e = 0.2, \alpha_v = 0.7, \alpha_r = 0.1$ for the loss function. More details of training are given in §5.1.

5. Experimental Results

Our network is trained end-to-end with SGD for all the datasets. The networks are implemented in Pytorch on a single Nvidia GeForce 1080 GPU with 8GB memory.

Datasets and Metrics. We conduct extensive experiments on multiple benchmark against conventional and learning-based state-of-the-art camera pose regression approaches. We report the median and mean angular errors along with the runtime for the experiments. For the datasets where the measurements of relative camera poses are not available, the initial view-graph is given by manually running the conventional state-of-the-art SfM system VisualSfM [68, 69] with Gaussian noises ($\mu = 20°$, $\sigma = 5°$) added on the edges of the initialized view-graph.

ScanNet [15] is an RGB-D video dataset containing 2.5 million views in more than 1500 indoor scans, the groundtruth includes the absolute camera orientations (given by [16]), triangulated surfaces and semantic segmentations. *The Cambridge dataset* [33] contains over 12000 images with groundtruth absolute camera orientations, taken in 6 outdoor scenes around Cambridge University. The dataset is challenging due to the presence of high amounts of moving objects and changing lightning conditions. *7 Scenes* [55] consists of 7 relatively small indoor scenes, tracked by a Kinect RGB-D camera. While the dataset with less than 10K images is small in scale compared with the other datasets, the view-graphs are highly noisy with the presence of various texture-less objects in scene, thus making it challenging. *The Photo Tourism datasets* [65] are a large collection of 19 outdoor scenes with more than 5k views and over 200K relative measurements on several datasets.

Baselines. We compare performance of PoGO-Net against both conventional and learning-based state-of-the-art approaches to demonstrate the efficiency and robustness of the proposed network. Among the methods, IRLS [11], IRLS-Robust [12], Weiszfeld [28], Arrigoni [4], DISCO [14], CEMP [39], MPLS [54] and Wang [63] are conventional MRA-PGO methods. Learning-based approaches include RelocNet [5], LsG [71], MapNet [8], PoseNet15 [33], PoseNet17 [32], PoseNet+LSTM [62], CNN+GNN [72] and NeuRoRA [49].

5.1. Implementation Details

For the training of PoGO-Net, we adopt SGD optimizer with no dropout. To prevent the 'over-smoothing' of GNNs, we conduct random shuffling within the batch (size = 64) with l_1 regularization. The backbone network adopts the original GNNs [53]. We train PoGO-Net according to the conventional split of the datasets, the learning rate is annealed geometrically starting at 1e-3 and decreases to 1e-5. The view-graph is initialized completely with the conventional spanning tree method, prone to broadcasting erroneous edge measurements. We thus address the de-noise layer parameter tuning by setting the weight of edge loss $\mathcal{L}_e$ (*i.e.* local edge consistency) slightly higher ($\alpha_e = 0.35$) on first 10% of the training data. The loss component weight parameters are set $\alpha_v = 0.7, \alpha_e = 0.2, \alpha_r = 0.1$ for the training of all the datasets. Though we set the maximum epochs to be 300, we have observed that the dropping of validating errors and testing errors terminates around 150-230 epochs in our experiments.

5.2. Performance Comparisons

7 Scenes. We first compare PoGO-Net with recent state-of-the-art learning-based PGO methods on the 7 Scenes dataset, the quantitative results are reported in Table.1. It can be seen that PoGO-Net has achieved best results on most of the scenes, among which on *Fire* and *Heads* datasets PoGO-Net outperforms the other approaches by large margins. On *Pumpkin* and *Stairs* dataset, PoGO-Net slightly falls short to previous approaches. Considering that both scenes hold high amounts of views with repetitive patterns and textureless surfaces, the main factor of errors roots from the exceedingly noisy image retrieval, *i.e.*, the erroneous feature extraction and matching causes the initial view-graph to be highly corrupted on most of the edges.

Table 2: Experiment results on the Cambridge Dataset [33]. Results are cited directly, the best results are highlighted.

Scene		MapNet [8]	PoseNet15 [33]	PoseNet17 [32]	PoseNet+LSTM [62]	CNN+GNN [72]	PoGO-Net *Tourism*	PoGO-Net *7Scenes*	PoGO-Net *ScanNet*	PoGO-Net *Cambridge*
T. G. Court	8.0x10^3 m^2	3.76°	-	3.27°	-	2.79°	3.23°	3.92°	3.66°	**1.96°**
Street	5.0x10^3 m^2	27.55°	-	15.50°	-	22.44°	19.29°	28.33°	23.17°	**11.76°**
K. College	5.6x10^3 m^2	1.89°	4.86°	1.04°	3.65°	**0.65°**	2.04°	3.89°	2.55°	0.94°
O. Hospital	2.0x10^3 m^2	3.91°	4.90°	3.29°	4.29°	2.78°	3.14°	3.65°	2.97°	**1.69°**
S. Facade	8.8x10^3 m^2	4.22°	7.18°	3.78°	7.44°	2.87°	3.93°	4.88°	4.06°	**2.40°**
St. M. Church	4.8x10^3 m^2	4.53°	7.96°	3.32°	6.68°	3.29°	3.66°	5.12°	3.49°	**2.12°**
Average		7.64°	6.23°	5.03°	5.52°	5.80°	5.04°	8.29°	6.65°	**3.47°**

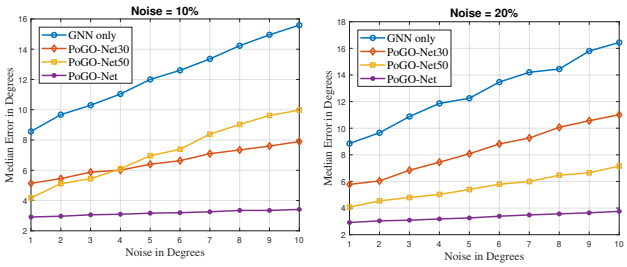
Note that [5] and [72] both have utilized the ResNet [30] feature extractor which is more robust compared with the conventional approach VisualSfM we adopt for the initial view-graph generation during the image retrieval phase.

Cambridge. In the experiments on the Cambridge dataset, we demonstrate the transferability of PoGO-Net by training on distinct datasets. Results are given in Table 2. Specifically, we record the comparable testing results on the Cambridge dataset with PoGO-Net trained solely on the 7Scenes [55], ScanNet [15] and the Photo Tourism [65] datasets separately. We finally report the performance with training and testing both on the Cambridge dataset and our PoGO-Net presents significant outperformances on most of the scenes, further proving the network robustness in large-scale outdoor scenes. Note that data on *Trinity Great Court* and *Street* are not provided for PoseNet15 [33] and PoseNet+LSTM [62], the average errors for the two approaches are based on the results on the left four scenes.

Table 3: Experiment results on the ScanNet Dataset [15]. Results are based on 5 runs of conventional approaches. The average runtime is evaluated on CPU.

	mean angle err.	median angle err.	runtime
IRLS [11]	14.07°	10.65°	2.08s
Robust-IRLS [12]	13.23°	8.17°	2.33s
Weiszfeld [28]	19.74°	15.32°	85.21s
Arrigoni [4]	27.16°	20.43°	37.83s
Wang [63]	16.30°	10.04°	13.2s
NeuRoRA [49]	11.02°	6.92°	0.92s
PoGO-Net	**8.22°**	**3.04°**	**0.37s**

ScanNet. We then test the performance of PoGO-Net against the conventional state-of-the-art approaches. Specifically, we record the angular errors and the runtime to demonstrate the accuracy and efficiency of PoGO-Net compared with traditional MRA-PGO methods. We also include the results reported by NeuRoRA [49], which is a GNN-based MRA framework with two sub-networks. Note that NeuRoRA is pre-trained with synthetic datasets which are captured by the authors, and the CleanNet and Fine-tuning network are trained separately while PoGO-Net is trained end-to-end without pre-tuned parameters. We cite the results reported in [49] for NeuRoRA and we execute the conventional approaches and report the 5-run averages, the results are given in Table 3. It can be seen that PoGO-Net outperforms the previous methods by a wide margin in both accuracy and speed.



(a) Noise = 10% (b) Noise = 20%

Figure 4: Study of different de-noise layers settings on the noise distributed to a) 10% b) 20% of the view-graph edges.

Tourism. Similar with the experiments on ScanNet, the angular errors and runtime of experiments on the Photo Tourism Dataset [65] are reported in Table 4. We cite the results partly from [4, 12, 49]. It can be observed that PoGO-Net has achieved the best results on most of the scenes. On the datasets with large-scale view-graphs (*e.g.* Piccadilly), PoGO-Net demonstrates its efficiency by outperforming conventional approaches by up to 400x faster and is almost 2x faster compared with learning-based NeuRoRA. Full result and more analysis of the experiments are provided in the *supplementary materials*.

5.3. Ablation Study

To study the effects of the de-noise layers, we conduct the ablation study on the 7Scenes dataset with several variations of PoGO-Net. In detail, we re-train the network with 0%, 30%, 50% amounts of the de-noise layers in the original PoGO-Net and test them on the testing sets with additional noise (from 1° to 10 °) on the randomly selected edges in the viewgraph. The accuracy plots are given in Fig. 4. With the setting of 0% of the de-noise layers, it is very difficult to initialize the nodes in the view-graph with the spanning tree scheme as the edge errors are severely propagated over the graph. Therefore in the experiments with the *GNN-only* variation, we first manually filter out outlier edges in randomly selected cycles in the view-graph by enforcing the cycle identity [49]. It can be seen that though the network with fewer de-noise layers can work, it yields a much lower accuracy compared with the original

Table 4: Experiment results on the Tourism Dataset [65]. We report the angular errors (°) and runtime (s) on CPU. The best results are highlighted. Full result is given in the *supplementary materials*.

Scene	# Nodes	# Edges		IRLS [11]	Robust-IRLS [12]	Weiszfeld [28]	Arrigoni [4]	Wang [63]	DISCO [14]	CEMP [39]	MPLS [54]	NeuRoRA [49]	PoGO-Net
Alamo	627	97206	mean	3.64	3.67	4.9	6.2	5.3	-	4.05	3.44	4.9	**2.96**
			median	1.30	1.32	1.4	1.2	1.1	7.86	1.62	1.16	1.2	**0.85**
			runtime	14.2s	15.1s	84.0s	2.7s	20.6s	3917s	10.38s	20.6s	2.2s	**1.74s**
M.N.D	474	52424	mean	1.25	1.22	2.1	4.8	2.0	-	1.33	1.04	1.2	**0.82**
			median	0.58	0.57	0.7	0.9	0.8	6.81	0.79	0.51	0.6	**0.37**
			runtime	8.5s	7.3s	41.5s	2.9s	10.1s	1608s	7.3s	9.3s	1.0s	**0.53s**
N.Dame	715	64678	mean	2.63	2.26	4.7	3.9	3.5	-	2.35	2.06	1.6	**1.17**
			median	0.78	0.71	0.8	1.0	0.9	7.48	0.94	0.67	0.6	**0.35**
			runtime	17.2s	22.5s	80.8s	4.2s	19.5s	4070s	13.2s	31.5s	2.0s	**1.24s**
Picca.	2508	319257	mean	5.12	5.19	26.4	22.0	10.1	36.0	4.66	**3.93**	4.7	4.93
			median	2.02	2.34	7.5	9.7	3.9	-	1.98	1.81	1.9	**1.75**
			runtime	353.5s	370.2s	1342.6s	43.7s	118.1s	15604s	45.8s	191.9s	5.9s	**3.19s**
R.Frm.	1134	70187	mean	2.66	2.69	4.8	13.2	4.6	-	2.80	2.62	2.30	**1.55**
			median	1.58	1.57	1.8	8.2	3.5	35.36	1.45	1.37	1.3	**0.69**
			runtime	18.6	21.4	115.0s	16.8s	19.6s	1559s	6.1s	8.8s	1.3s	**1.26s**
T.o.L.	508	24863	mean	3.42	3.41	4.7	4.6	2.9	-	2.84	3.16	2.6	**1.77**
			median	2.52	2.50	2.9	1.8	1.5	10.38	1.57	2.20	1.4	**0.43**
			runtime	2.6s	2.4s	17.4s	3.9s	3.6s	479s	2.2s	2.7s	**0.3s**	0.38s
U.Sq.	930	25561	mean	6.77	6.77	40.9	9.2	6.8	-	7.47	6.54	5.9	**3.3**
			median	3.66	3.85	10.3	4.4	3.2	26.27	3.64	3.48	2.0	**1.25**
			runtime	9.0s	8.6s	42.8s	12.1s	4.1s	466s	2.5s	5.7s	0.6s	**0.29s**
Yorkm.	458	27729	mean	2.6	2.45	5.7	4.5	3.5	-	2.49	2.47	2.5	**2.03**
			median	1.59	1.53	2.0	1.6	1.3	26.17	1.37	1.45	0.9	**0.72**
			runtime	3.4s	4.3s	32.0s	2.5s	4.9s	641s	2.8s	3.9s	0.4s	**0.12s**
San.F.	7866	101512	mean	4.3	**3.6**	18.8	66.8	89.2	-	-	-	17.6	6.82
			median	3.9	3.4	16.4	43.9	75.5	54.38	-	-	12.6	**3.16**
			runtime	18.9s	15.2s	1462.7s	354.7s	27.2s	1413s	-	-	2.6s	**1.54s**
Vien.C.	918	103550	mean	9.1	8.2	11.7	19.3	10.1	6.91	7.21	-	**3.9**	4.26
			median	3.9	1.2	1.9	2.39	1.8	22.35	2.63	2.83	1.5	**1.44**
			runtime	56.9s	48.1s	158.3s	6.0s	25.7s	4085s	13.1s	42.6s	2.1s	**1.53s**

PoGO-Net. Moreover, it is noteworthy that the accuracy of PoGO-Net holds stable in spite of the increasing noise level, further demonstrating the robustness of the network. The full study on the de-noise layer effects are provided in the *supplementary materials*.

5.4. Discussions and Future Work

To further demonstrate the capability of generalization of PoGO-Net, we test it on the KITTI Odometry [23] and integrate it with the state-of-the-art SLAM pipeline ORB-SLAM [47]. Evaluations and analysis are given in the *supplementary materials*. Observing that PoGO-Net achieves real-time performances with high accuracy further validates the potential of PoGO-Net as to be extended to a full SfM/SLAM system. While accurate MRA, especially combined with the graph-based formulation, is compact and lightweight to address PGO efficiently, expanding PoGO-Net for $\mathbb{SE}(3)$ regression is neither immediate nor trivial. We nonetheless believe that the adoption of feature subnets endows the full pose regression, such that rotations and translations can be jointly optimized within the graph form.

6. Conclusion

In this work, we propose a novel PGO scheme fueled by GNNs, namely PoGO-Net, to conduct the absolute camera pose regression leveraging MRA. PoGO-Net takes noisy view-graphs as inputs where the nodes and edges are designed to encode the pair-wise geometric constraints and aggregated with the local graph consistency. To address the outlier edge removal toward a robust MRA-GNN approach, we design the de-noise layers by exploiting an edge-dropping scheme on the noisy or corrupted edges, which are effectively filtered out with parameterized networks. Our joint loss function embeds MRA formulation, enabling end-to-end training so that the parameters of the de-noise layers and GNN layers optimized concurrently. Extensive experiments on multiple benchmarks demonstrate the accuracy, efficiency and robustness of PoGO-Net.

Acknowledgment. This work is supported in part by National Science Foundation Grants 2006665 and 1814745.

References

[1] S. Agarwal, K. Mierle, and Others. Ceres solver. http://ceres-solver.org.

[2] A. M. Andrew. Multiple view geometry in computer vision. *Kybernetes*, 2001.

[3] F. Arrigoni and A. Fusiello. Synchronization problems in computer vision with closed-form solutions. *International Journal of Computer Vision (IJCV)*, 128(1):26–52, 2020.

[4] F. Arrigoni, B. Rossi, P. Fragneto, and A. Fusiello. Robust synchronization in so (3) and se (3) via low-rank and sparse matrix decomposition. *Computer Vision and Image Understanding*, 174:95–113, 2018.

[5] V. Balntas, S. Li, and V. Prisacariu. Relocnet: Continuous metric learning relocalisation using neural nets. In *European Conference on Computer Vision (ECCV)*, 2018.

[6] U. Bhattacharya and V. M. Govindu. Efficient and robust registration on the 3d special euclidean group. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

[7] T. Birdal, M. Arbel, U. Simsekli, and L. J. Guibas. Synchronizing probability measures on rotations via optimal transport. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[8] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz. Geometry-aware learning of maps for camera localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[9] A. P. Bustos, T.-J. Chin, A. Eriksson, and I. Reid. Visual slam: Why bundle adjust? In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[10] 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 *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[11] A. Chatterjee and V. M. Govindu. Efficient and robust large-scale rotation averaging. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013.

[12] A. Chatterjee and V. M. Govindu. Robust relative rotation averaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 40(4), 2017.

[13] R. Clark, S. Wang, A. Markham, N. Trigoni, and H. Wen. VidLoc: A deep spatio-temporal model for 6-dof video-clip relocalization. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[14] D. Crandall, A. Owens, N. Snavely, and D. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[15] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. ScanNet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[16] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (ToG)*, 36(4):1, 2017.

[17] F. Dellaert, D. M. Rosen, J. Wu, R. Mahony, and L. Carlone. Shonan rotation averaging: Global optimality by surfing $SO(p)^n$. In *European Conference on Computer Vision (ECCV)*. Springer, 2020.

[18] M. Ding, Z. Wang, J. Sun, J. Shi, and P. Luo. Camnet: Coarse-to-fine retrieval for camera re-localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

[19] J. Engel, T. Schöps, and D. Cremers. Lsd-slam: Large-scale direct monocular slam. In *European conference on computer vision (ECCV)*. Springer, 2014.

[20] A. Eriksson, C. Olsson, F. Kahl, and T.-J. Chin. Rotation averaging with the chordal distance: Global minimizers and strong duality. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 43(1):256–268, 2019.

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

[22] R. Garg, V. K. Bg, G. Carneiro, and I. Reid. Unsupervised CNN for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision (ECCV)*. Springer, 2016.

[23] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[24] S. Gidaris and N. Komodakis. Generating classification weights with GNN denoising autoencoders for few-shot learning. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[25] Z. Gojcic, C. Zhou, J. D. Wegner, L. J. Guibas, and T. Birdal. Learning multiview 3d point cloud registration. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[26] V. M. Govindu. Combining two-view constraints for motion estimation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[27] G. Grisetti, R. Kümmerle, H. Strasdat, and K. Konolige. g2o: A general framework for (hyper) graph optimization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[28] R. Hartley, K. Aftab, and J. Trumpf. L1 rotation averaging using the Weiszfeld algorithm. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[29] R. Hartley, J. Trumpf, Y. Dai, and H. Li. Rotation averaging. *International Journal of Computer Vision (IJCV)*, 103 (3), 2013.

[30] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[31] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. *International Conference on Learning Representations (ICLR)*, 2017.

[32] A. Kendall and R. Cipolla. Geometric loss functions for camera pose regression with deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[33] A. Kendall, M. Grimes, and R. Cipolla. PoseNet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[34] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations (ICLR)*, 2017.

[35] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, 2007.

[36] M. Klodt and A. Vedaldi. Supervising the new with the old: learning sfm from sfm. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[37] Z. Laskar, I. Melekhov, S. Kalia, and J. Kannala. Camera relocalization by computing pairwise relative poses using convolutional neural network. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017.

[38] S. H. Lee and J. Civera. Robust single rotation averaging. *Computing Research Repository (CoRR)*, 2020.

[39] G. Lerman and Y. Shi. Robust group synchronization via cycle-edge message passing. *arXiv preprint arXiv:1912.11347*, 2019.

[40] X. Li and H. Ling. Hybrid camera pose estimation with online partitioning for SLAM. *IEEE Robotics and Automation Letters (RA-L)*, 5(2):1453–1460, 2020.

[41] C. Louizos, M. Welling, and D. P. Kingma. Learning sparse neural networks through l_0 regularization. *International Conference on Learning Representations (ICLR)*, 2018.

[42] M. I. Lourakis and A. A. Argyros. Sba: A software package for generic sparse bundle adjustment. *ACM Transactions on Mathematical Software (TOMS)*, 36(1):1–30, 2009.

[43] D. Luo, W. Cheng, W. Yu, B. Zong, J. Ni, H. Chen, and X. Zhang. Learning to drop: Robust graph neural network via topological denoising. *Web Search and Data Mining (WSDM)*, 2021.

[44] D. Martinec and T. Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[45] J. J. Moré. The Levenberg-Marquardt algorithm: implementation and theory. *Numerical analysis*, pages 105–116, 1978.

[46] P. Moulon, P. Monasse, and R. Marlet. Global fusion of relative motions for robust, accurate and scalable structure from motion. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013.

[47] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics (T-RO)*, 31(5):1147–1163, 2015.

[48] M. J. Powell. A new algorithm for unconstrained optimization. In *Nonlinear programming*, pages 31–65. Elsevier, 1970.

[49] P. Purkait, T.-J. Chin, and I. Reid. Neurora: Neural robust rotation averaging. In *European Conference on Computer Vision (ECCV)*. Springer, 2020.

[50] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J.-M. Frahm. Usac: a universal framework for random sample consensus. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 35(8):2022–2038, 2012.

[51] Y. Rong, W. Huang, T. Xu, and J. Huang. Dropedge: Towards deep graph convolutional networks on node classification. *International Conference on Learning Representations (ICLR)*, 2020.

[52] T. Sattler, Q. Zhou, M. Pollefeys, and L. Leal-Taixe. Understanding the limitations of cnn-based absolute camera pose regression. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[53] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.

[54] Y. Shi and G. Lerman. Message passing least squares framework and its application to rotation synchronization. *Proceedings of the International Conference on Machine Learning (ICML)*, 2020.

[55] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[56] N. Snavely, S. M. Seitz, and R. Szeliski. Skeletal graphs for efficient structure from motion. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[57] C. Tang and P. Tan. BA-Net: Dense bundle adjustment networks. In *International Conference on Learning Representations (ICLR)*, 2018.

[58] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment—a modern synthesis. In *International Workshop on Vision Algorithms*, pages 298–372. Springer, 1999.

[59] A. Valada, N. Radwan, and W. Burgard. Deep auxiliary learning for visual localization and odometry. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[60] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. Graph attention networks. *International Conference on Learning Representations (ICLR)*, 2018.

[61] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki. Sfm-net: Learning of structure and motion from video. *arXiv preprint arXiv:1704.07804*, 2017.

[62] F. Walch, C. Hazirbas, L. Leal-Taixe, T. Sattler, S. Hilsenbeck, and D. Cremers. Image-based localization using lstms for structured feature correlation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[63] L. Wang and A. Singer. Exact and stable recovery of rotations for robust synchronization. *Information and Inference: A Journal of the IMA*, 2(2):145–193, 2013.

[64] R. W. Wedderburn. Quasi-likelihood functions, generalized linear models, and the gauss—newton method. *Biometrika*, 61(3):439–447, 1974.

[65] K. Wilson and N. Snavely. Robust global translations with 1dsfm. In *European Conference on Computer Vision (ECCV)*. Springer, 2014.

[66] K. Wilson, D. Bindel, and N. Snavely. When is rotations averaging hard? In *European Conference on Computer Vision (ECCV)*. Springer, 2016.

[67] C. Wu. Towards linear-time incremental structure from motion. In *International Conference on 3D Vision (3DV)*. IEEE, 2013.

[68] C. Wu, S. Agarwal, B. Curless, and S. M. Seitz. Multicore bundle adjustment. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[69] C. Wu et al. Visualsfm: A visual structure from motion system. 2011.

[70] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? *International Conference on Learning Representations (ICLR)*, 2019.

[71] F. Xue, X. Wang, Z. Yan, Q. Wang, J. Wang, and H. Zha. Local supports global: Deep camera relocalization with sequence enhancement. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

[72] F. Xue, X. Wu, S. Cai, and J. Wang. Learning multi-view camera relocalization with graph neural networks. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[73] L. Zhao and L. Akoglu. PairNorm: Tackling oversmoothing in gnns. In *International Conference on Learning Representations (ICLR)*, 2019.