GTCaR: Graph Transformer for Camera Re-localization

Xinyi Li[1] and Haibin Ling[2*]

[1] Magic Leap, Sunnyvale, CA, USA xinli@magicleap.com
[2] Stony Brook University, Stony Brook, NY, USA hling@cs.stonybrook.edu

Abstract. Camera re-localization or absolute pose regression is the centerpiece in numerous computer vision tasks such as visual odometry, structure from motion (SfM) and SLAM. In this paper we propose a neural network approach with a graph Transformer backbone, namely **GTCaR** (**G**raph **T**ransformer for **Ca**mera **R**e-localization), to address the multi-view camera re-localization problem. In contrast with prior work where the pose regression is mainly guided by photometric consistency, GTCaR effectively fuses the image features, camera pose information and inter-frame relative camera motions into encoded graph attributes. Moreover, GTCaR is trained towards the graph consistency and pose accuracy combined instead, yielding significantly higher computational efficiency. By leveraging graph Transformer layers with edge features and enabling the adjacency tensor, GTCaR dynamically captures the global attention and thus endows the pose graph with evolving structures to achieve improved robustness and accuracy. In addition, optional temporal Transformer layers actively enhance the spatiotemporal inter-frame relation for sequential inputs. Evaluation of the proposed network on various public benchmarks demonstrates that GTCaR outperforms state-of-the-art approaches.

1 Introduction

The past decade has witnessed surging research interest in developing camera pose regression methods, benefiting various computer vision applications including robot navigation, autonomous driving and AR/VR technologies. *Camera re-localization* is an absolute pose regression (APR) process to localize query images against a known 3D environment. Conventional approaches solving the camera pose estimation problem involve extensive implementations of Perspective-n-Point (PnP) [19] followed by optimization steps of bundle adjustment (BA) [37], which is the iterative process of joint optimization of the 3D scene points and the 6-DoF camera pose parameters, aided by numerical solvers. The formulation yields a non-linear high-dimensional system and is thus computationally challenging to solve [36, 44].

With the prevalence of deep neural networks, many recent studies have steered research attentions towards leveraging deep learning techniques to re-formulate

[*] Corresponding author.

the camera pose estimation problem as a pose regression network, *i.e.*, the network is trained with training images and the ground-truth camera poses such that it can learn to regress the camera pose(s) given single or multiple images. Among these studies, PoseNet [22] pioneers in incorporating neural networks into camera pose regression frameworks, where the CNN-based network is trained to directly estimate the camera pose from individual images without explicit feature processing. As multi-view APR methods can preserve more inter-frame information (*e.g.*, temporal/global pose consistency) beyond those achieved solely from single image retrieval, they yield higher accuracy and robustness [26,28,29,31,48]. Later work adopts sophisticated networks to address the task, *e.g.*, in VidLoc [6] a CNN-RNN joint model is presented to leverage the temporal consistency of the sequential images. Recently, GNNs have been exploited in camera pose regression [48], where the message passing scheme embraces the inter-frame dependency.

Lately, the development of Transformers [39] has empowered massively successful applications in natural language processing (NLP), computer vision [3,12] and many other fields. Specifically, the adoption of the self-attention mechanisms enables Transformers to effectively capture the global spatiotemporal consistency of sequential information. Additionally, while graph-based networks such as GNNs have been widely proven to be efficient in modeling arbitrarily structured inputs, it is generally computationally challenging to have the networks update the graph structure dynamically [40,46,50,51], limiting its performance on downstream tasks where high amounts of noise or missing information are present.

Inspired by the aforementioned observations, in this work we propose a neural network fused with a graph Transformer backbone, namely *GTCaR*, to tackle the camera re-localization problem. In GTCaR, the view graph is constructed by a novel graph embedding mechanism, where the nodes are encoded with image features and 6-DoF absolute camera pose of the image frame, while the edge attributes consist of the relative inter-frame camera motions. Moreover, our proposed network introduces an *adjacency tensor* that stores the correlation on both the feature level and the frame level. In particular, the feature correspondences between the frames are encoded into the elements in the adjacency matrix, where the element value is based on the normalized feature correspondence score and thus falls into the range of $[0,1]$. The adjacency matrix is updated through the graph Transformer layers to reflect the evolving graph structure, *e.g.*, redundant/noisy edge pruning, newly-added edges according to high correlations between a new image and some previous image, *etc*. GTCaR is trained end-to-end, guided by the loss function that integrates the graph consistency [1] such that to localize multiple query images simultaneously. Additionally, the temporal Transformer layers are utilized to obtain the temporal graph attention for consecutive images.

The architecture overview of GTCaR is given in Fig. 1. The design of the proposed network is favorable for camera re-localization tasks in three aspects. First, it is efficient to exploit the intra- and inter-frame structure information and correlation with the utilization of graphs; Second, the self-attention mechanism can effectively capture the spatiotemporal consistency in arbitrarily long-term periods, achieving high global pose accuracy; Third, with the adjacency ma-

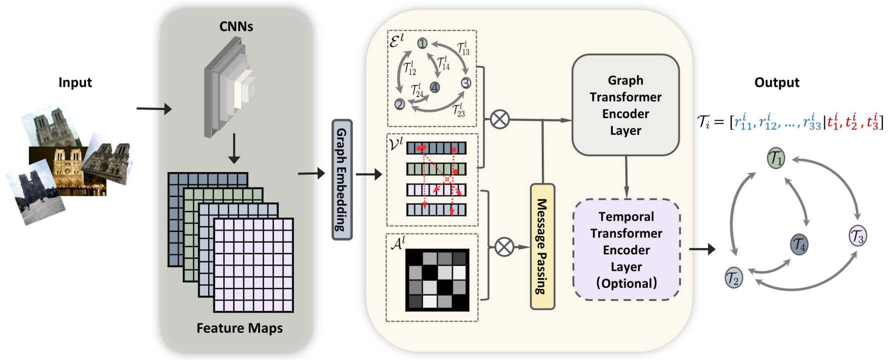

Fig. 1: Overview of the proposed GTCaR architecture for camera re-localization. The network takes query images as input and then models the corresponding camera poses, image features and the pair-wise relative camera motions into a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$. Then, the adjacency tensor $\mathcal{A}$ and nodes are fed into the message passing layers, before passing through the graph Transformer encoder layers ("l" indicates the l-th layer). For consecutive image sequences, the graph will be passed through additional temporal Transformer encoder layers. The global camera poses are embedded into the node information in the final output.

trix being dynamically updated, the network can quickly adjust according to the changing graph structure, further reducing the negative effects caused by erroneous feature matching.

To the best of our knowledge, our proposed network is the first to exploit graph Transformer for camera re-localization. Our contributions can be summarized as:

– We propose a novel framework with the Transformer backbone for the multiple camera re-localization task. By encoding the image features, intra- and inter-frame relative camera poses into a graph, the proposed network is trained efficiently towards both the pose accuracy and the graph consistency.
– We design an adjacency tensor to dynamically capture the global attention, so as to endow the pose-graph with an evolving structure to achieve boosted robustness and accuracy.
– We exploit optional temporal Transformer layers to obtain the temporal graph attention for consecutive images, such that the proposed model can work with both unordered and sequential data.

2 Related Work

Graph Transformers. By virtue of its powerful yet agile data representation, GNNs [23,32,40] have achieved exceptional performances on numerous computer vision tasks. In [10], Graph-BERT enables pre-training on the original graphs and adopts a subgraph batching scheme for parallelized learning. However, Graph-BERT assumes that the subgraphs are linkless, thus not suitable for tasks where global connectivity is important. Recently with the success of Transformers [39], several studies [5,13,43,50] have attempted to develop graph Transformers which can leverage the powerful message passing scheme on graphs while utilizing the multi-head self attention mechanism in Transformers. Among which the approach proposed in [43] is capable of transforming the heterogeneous graphs into homogeneous graphs such that the Transformer can be exploited. GTNs proposed in [50] also addresses the heterogeneous graphs, where the proposed network is capable of generating new graph structures by defining meta-paths with arbitrary edge types. In [13], a generalized graph form of Transformers is proposed with the edge features addressed. Despite their successes, straightforward adoptions of GNNs in modeling camera re-localization task is not applicable due to GNN's vulnerability against noisy graphs [15,30,38,46,52].

Camera Pose Regression Networks. It was not until recently that research interests began to focus on incorporating deep neural networks into SfM pipelines and camera pose regression tasks [2, 11, 14, 22, 24, 36, 41, 47]. As one of the earliest work adopting neural networks for camera pose regression, the deep convolutional neural network pose regressor proposed in [22] is trained according to a loss function embedding the absolute camera pose prediction error. While [22] 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 accuracy, as improved later in [6,31,48]. Other work exploits the algebraic or geometric relations among the given images and train the networks to predict to locate the images [4,6,38,41], among which [6] leverages temporal consistency of the sequential images by equipping bi-directional LSTMs [18] with a CNN-RNN model such that temporal regularity can provide more pose information in the regression. The approach in [4] trains DNNs model with the pair-wise geometric constraints between frames, by leveraging additional sensor measurements.

Recent work [48] is the first study to leverage GNNs in a full absolute camera pose regression framework, where the authors model the view-graph with CNN-feature nodes. Later study [26] proposes a pose-graph optimization framework with GNNs, guided by the multiple rotation averaging scheme. In [33], a multi-scene absolute camera pose regression framework with Transformers is proposed. While GNNs are capable of effectively capturing the topological neighborhood information of each individual node (*i.e.*, the featured frame in such task), they are rather prone to noise; Moreover, co-visibility graphs in real-world camera relocalization tasks are often quite dense, causing either noise removal or 'edge-dropping' further entangled [13,26,44,51]. Leveraging graph Transformers in relocalization tasks facilitates the noise handling by virtue of the attention mechanism in (original) Transformers. Our work differs from [33] on: 1) we model the pose regression with a graph structure; 2) we train one end-to-end graph Transformer network while in [33] two separate Transformers are adopted for rotation and translation regression respectively; 3) we leverage rotation averaging addressing both the graph consistency and pose accuracy to guide the training, whereas only camera pose loss is exploited in the training of [33].

3 Problem Formulation

Given a set of 2D image frames and a known 3D scene, *camera re-localization* seeks a consistent set of optimized camera rigid motions, aiming to recover the locations and orientations of the camera aligned with the scene coordinate. Formally, let $\mathbf{R}_i \in \mathbb{SO}(3)$ and $\mathbf{t}_i \in \mathbb{R}^3$ denote the camera orientation and the camera translation for the i^{th} image frame respectively, the absolute camera pose is denoted by $\mathcal{T}_i = [\mathbf{R}_i|\mathbf{t}_i]$. Then the camera re-localization task can be formulated into the following pose regression objective

$$\arg\min_{\mathcal{T}_i} \sum_i \rho\big(d(\mathcal{T}_i, \overline{\mathcal{T}_i})\big), \tag{1}$$

where $\rho(\cdot)$ is a robust cost function, $d(\cdot,\cdot)$ is a distance metric and $\overline{\mathcal{T}_i} = [\overline{\mathbf{R}_i}|\overline{\mathbf{t}_i}]$ denotes the groundtruth camera poses. Accordingly, let $\mathcal{T}_{ij} = [\mathbf{R}_{ij}|\mathbf{t}_{ij}]$ denote the relative camera motion between the i^{th} and j^{th} image frames. In our formulation, we leverage multiple rotation averaging [25, 26, 29, 34, 49] and introduce the *graph-level consistency* term into the objective, that is

$$\arg\min_{\mathbf{R}_i,\mathbf{R}_j} \sum_{(i,j)} \rho\big(d(\mathbf{R}_{ij}, \mathbf{R}_j\mathbf{R}_i^{-1})\big). \tag{2}$$

In detail, given the camera relative orientations $\{\mathbf{R}_{ij}\}$, the optimization process involves minimizing a cost function that penalizes the discrepancy between the camera relative orientations achieved from image retrieval and those inferred from the solved absolute camera poses. We argue that low costs in Eq. 2 indicate high global consistency of the solution set, and thus fuse the cost into the loss function as the *global consistency loss*. Therefore, given the ground truth camera poses, the objective function is assembled as

$$\begin{aligned}\arg\min_{\mathbf{R}_i,\mathbf{R}_j} &\sum_{(i,j)} \rho\big(d_{\mathbf{R}}(\mathbf{R}_{ij}, \mathbf{R}_j\mathbf{R}_i^{-1})\big) \\ &+ \arg\min_{\mathbf{R}_i} \sum_i \rho'\big(d_{\mathbf{R}}(\mathbf{R}_i, \overline{\mathbf{R}_i})\big) + \arg\min_{\mathbf{t}_i} \sum_i \rho''\big(d_{\mathbf{t}}(\mathbf{t}_i, \overline{\mathbf{t}_i})\big),\end{aligned} \tag{3}$$

where ρ' and ρ'' are robust cost functions, $d_{\mathbf{R}} : \mathbb{SO}(3) \times \mathbb{SO}(3) \to \mathbb{R}_+$ and $d_{\mathbf{t}} : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}_+$ are the distance metrics for rotations and translations respectively. Specifically, the first term measures the global consistency, *i.e.*, it should be zero if the relative transformations on the edges align perfectly with the absolute transformations on the nodes for the whole graph. The other two terms depict the rotation and translation prediction errors respectively, echoing Eq. 1. Details on the loss function formulation are given in §4.4.

In the design of our proposed network, we model the multi-view camera re-localization problem as graphs and embed the 2D image features and the camera absolute pose $\mathcal{T}_i$ as the corresponding latent node information, whereas the inter-frame camera relative motions $\mathcal{T}_{ij}$ are encoded as the edge attributes, as introduced in §4.2.

4 GTCaR Architecture

In this section we detail the network architecture of the proposed GTCaR. First we provide the architecture overview in §4.1, followed by the elaboration of feature embedding and graph embedding in §4.2. We then emphasize the structure of the spatiotemporal graph Transformer layers in §4.3, followed by the graph update and the proposed graph loss function illustrated in §4.4.

4.1 Architecture Overview

As shown in Fig. 1, the proposed network takes query RGB images as input. The images are first fed into a pre-trained CNN-type [17] feature network, then the output feature maps are embedded in an initial view-graph such that the nodes encode the visual information of the images, and the edges encode inter-frame correlations. Additionally, the local feature matching information and the aggregated image matching score are combined and arranged into a tensorized adjacency matrix, namely the adjacency tensor.

After assembling the images into a graph, the adjacency tensor and the hidden node features are first passed into MPNN [16] layers such that, for each node, the neighboring node features are aggregated efficiently with the implicit attention information embedded in the adjacency tensor. Then the aggregated node features are fed into graph Transformer encoder layers, where the self attention mechanism are equipped with edge features such that the camera relative transformations encoded on the edges can be exploited to generate the attention weights. Additionally, the temporal Transformer encoder layers capture the self-attention for the sequential input. The global camera poses, as node attributes, are updated through the network and are embedded in the final output as the localized camera poses.

4.2 Graph Embedding

We propose to model the input query images, the corresponding camera poses and the pair-wise camera transformations into a graph based on the construction of conventional pose graph, *i.e.*, each node represents an image frame and the edges connecting two nodes represent the inter-frame image relations. In detail, consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{v_i\}$ denotes the set of the images and $\mathcal{E} = \{(i,j)|v_i, v_j \in \mathcal{V}\}$ represents the pair-wise feature-base connectivity between frames. Additionally, let $\mathcal{A}_\mathcal{G}$ denote the adjacency matrix of $\mathcal{G}$ such that $\mathcal{A}_\mathcal{G}(i,j) = 0$ if $(i,j) \notin \mathcal{E}$ and vice versa. For simplicity of notation, we will use $\mathcal{A}$ for $\mathcal{A}_\mathcal{G}$ in the following discussion.
Node Attributes. Consider an image $\mathbf{I}_i$, let $\mathbf{x}_i$ denote the feature vector as the output of the CNN-type feature sub-network, and denote $\mathbf{p}_i \in \mathbb{R}^7$ as the camera absolute pose vector, where $\mathbf{p}_i$ consists of the 4-dimensional quaternion ω_i representing the camera orientation and the 3-dimensional t_i representing the camera translation. That is, the vector embedding of each node v_i contains the information part which encodes the image latent feature and the learning part

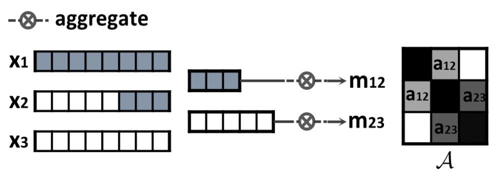

Fig. 2: Each element a_{ij} of the adjacency tensor $\mathcal{A}$ embeds the feature correspondences and the normalized aggregated value. $m_{ij} = 0$ if there exists none co-visible feature between image i and image j. Note that $\mathcal{A}$ is symmetric.

which embeds the camera pose. It is noteworthy to mention that, in contrast with NLP tasks where the word positions or text orders are crucial, the camera absolute poses are invariant to node positions as we leverage the graph structure to model the problem. We believe that topological position (vertex degree, local neighborhood structure, global connectivity, etc.) plays a significant role in the proposed graph-based framework, therefore we skip the positional encoding in the original Transformer model [39] and embed the 'relative position' or 'relative distance' as the image matching vector into the adjacency tensor instead.

Adjacency Tensor. Let a_{ij} be the element at (i, j) of the adjacency matrix with self-connections $\mathcal{A}$, by convention $a_{ij} = 1$ if there exists an edge connecting v_i and v_j and $a_{ij} = 0$ otherwise. To capture and maintain the pair-wise relation, we introduce the adjacency tensor where a_{ij} represents the vector feature correspondence index between the i^{th} and j^{th} image frames.

Specifically, consider $a_{ij} \in \mathcal{A}$ and let $\mathbf{x}_i$ and $\mathbf{x}_j$ be the corresponding feature vectors, and assume that there exists some feature correspondences between image i and image j. Then a_{ij}^k, *i.e.*, the k^{th} element of a_{ij} portraying the k^{th} feature correspondence, is a tuple with the feature index in $\mathbf{x}_i$ and $\mathbf{x}_j$ respectively. That is, $\mathbf{x}_i(a_{ij}^k(1)) \sim \mathbf{x}_j(a_{ij}^k(2))$. Additionally, each vector a_{ij} is aggregated into an initial meta-feature $\mathbf{m}_{ij}$ as the normalized feature correspondence score with range $[0, 1]$, which measures the edge credibility evaluation and the image matching result between the two connected nodes. The adjacency tensor encodes pixel-wise and image-wise correspondence, depicting the edge weights and is updated through the network while interacting spatiotemporally with the whole evolving graph. Illustration of the adjacency tensor is given in Fig. 2.

Edge Attributes. Similar with the 7-dimensional pose feature embedded on the nodes, the camera relative transformation is encoded on the edge connecting v_i and v_j as $\mathbf{p}_{ij} = \langle \omega_{ij}, t_{ij} \rangle$. During the graph embedding, only nodes with matched features are connected with edges with initialized edge feature (unit quaternion translation and zero vector translation). In our modeling of the graph we consider the edge features as node-symmetric according to the nature of pose graph construction. As we aim to keep the graph lightweight, the edges do not contain any low-level correspondence information between the connected nodes. Instead, the inter-node dependency is implicitly arranged into the adjacency tensor $\mathcal{A}$.

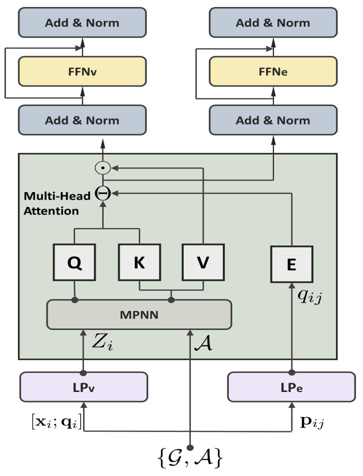

Fig. 3: The graph Transformer encoder layer structure. Q, K and V are compliant with the original Transformer, E represents the edge attention module.

4.3 Graph Transformer Layer

Now we have constructed the graph embedding the node and edge features as the input into the graph Transformer layer. Our proposed network adopts the encoder layer structure in the original Transformer [39] and transforms the initial source graph to the target graph with evolved structural edge information and derived pose values on the nodes. Specifically, the graph Transformer layer exploits the multi-head attention mechanism to generate the sptiotemporal relation between nodes, such that 1) the edges connecting two nodes where high amounts of common features (pair-wise co-visible visual features) are equipped with high attention weights and 2) the edges carrying abundant or noisy image matching yield low attention weights or get removed from the graph. The emerging adjacency tensor progressively interacts with the whole graph and propagates the update over the nodes and the edges.

Message Passing. Before passing the graph into the graph Transformer encoder layer, the neighboring node features are aggregated along with the adjacency tensor for each node. Specifically, consider the graph at the l^{th} layer and let Z^l denote the hidden feature tensor of the nodes, let $\mathcal{A}^l$ denote the adjacency tensor. Then after the message passing layers the node tensor is thus

$$\hat{Z}^l = Z^l \mathbin{+\!\!+} [\phi(\mathcal{A}^l, Z^l) \otimes Z^l], \tag{4}$$

where $\mathbin{+\!\!+}$ denotes the concatenation operation, $\phi(,)$ denotes the message aggregation, $\otimes$ denotes the tensor product. We adopt the mean function as the aggregation operation in this work. Precisely, Z^l embeds the node information while the latter term embeds the edge information over the neighborhood. The adjacency tensor is exploited here instead of the edges as $\mathcal{A}$ has collected the

local attention information such that the message passing is more efficient.
Graph Transformer Encoder Layer. We leverage the multi-head self attention mechanism in the graph Transformer encoder layer with edge features. Borrowing notations from the original Transformer network, let $Q_k^l, K_k^l, V_k^l \in \mathbb{R}^{d_k \times d}$, where $k = 1$ to N is the number of the attention heads, d_k denotes the query dimension. Consider the attention weight for the k^{th} head on the edge connecting the source node i and the target node j, that is

$$w_{ij} = \text{softmax}_j(Q_k^l \hat{Z}_i^l \odot K_k^l \hat{Z}_j^l), \tag{5}$$

where $\odot$ denotes the Hadamard product. Following [13], we add the edge features into generating the attention. Let E_k^l be in the same dimension space with Q_k^l, K_k^l, V_k^l and let q_{ij} denote the hidden edge features, then the attention weight with edge feature is thus

$$w_{ij}^e = \text{softmax}_j \Theta(Q_k^l \hat{Z}_i^l, K_k^l \hat{Z}_j^l, E_k^l q_{ij}^l), \tag{6}$$

where Θ denotes the consecutive dot product operation. Then the update function for nodes and edges are thus

$$Z_i^{l+1} = \mathbin{+\!\!+}_k \ (w_{ij}^e V_k^l \hat{Z}_j^l) \otimes O_Z^l, \tag{7}$$

$$q_{ij}^{l+1} = \mathbin{+\!\!+}_k \ (w_{ij}^e) \otimes O_e^l, \tag{8}$$

where $O_Z^l, O_e^l \in \mathbb{R}^{d \times d}$, d is the dimension of the hidden space of nodes and edges, $\mathbin{+\!\!+}_k$ denotes multihead (k heads) concatenation. Illustration is given in Fig. 3.
Temporal Transformer Encoder Layer. The temporal inter-frame relation contains high amounts of useful information especially when the input is sequential images or video clips. In the proposed network we address the temporal dependencies for consecutive camera re-localization tasks by equipping the network with an optional temporal Transformer encoder layer. The temporal Transformer encoder layer exploits the standard Transformer network structure, takes the graph embedding as input and generates intra-graph temporal dependencies between nodes by constructing temporal attention.

4.4 Graph Loss and Update

GTCaR is trained end-to-end, guided by the joint loss function representing both the graph consistency and the accuracy of the predicted camera poses. Recalling the objective function Eq. 3, the loss function is thus assembled as follows

$$\begin{aligned}\mathcal{L} = \alpha \sum_{i,j} \rho(d_{\mathbf{R}}(\omega_{ij}, \omega_j \omega_i^{-1})) + \alpha' \sum_{i,j} \rho'(d_{\mathbf{t}}(t_{ij}, d_{\mathbf{t}}(t_i, t_j))) \\ + \beta \sum_i \rho(d_{\mathbf{R}}(\omega_i, \overline{\omega_i})) + \beta' \sum_i \rho'(d_{\mathbf{t}}(t_i, \overline{t_i})),\end{aligned} \tag{9}$$

where $\alpha, \alpha', \beta, \beta' \in \mathbb{R}$ are the loss parameters, $\overline{\omega_i}, \overline{t_i}$ are the ground truth camera orientations and translations. The graph loss function can be seen as a joint optimization regarding both the graph consistency and the prediction accuracy.

Specifically, during the training the nodes are updated according to a) edge updates which reflect both relative transformation updates and graph connectivity updates (first two terms in Eq. 9), and b) node updates according to the absolute pose loss (last two terms in Eq. 9). Therefore the graph evolves in terms of 1) message passing aggregates attention with the pose information embedded into the nodes and the local connectivity embedded by the adjacency tensor, then 2) attention mechanism assists to update attention weights on the edges, followed by 3) node and edge features (absolute and relative poses) are updated according to the attention, represented in Eq. 7 and Eq. 8. The graph is therefore evolving with nodes, edges, and adjacency updated.

5 Experimental Results

The proposed network is evaluated on three public benchmarks: 7-Scenes [35], the Cambridge dataset [22] and the Oxford Robotcar dataset [27]. We first elaborate the datasets, metrics, baselines and implementation details we conduct the experiments with (§5.1), followed by the evaluation results (§5.2), we then conduct the ablation study on the spatiotemporal mechanism of the proposed network (§5.3) and discuss the limitations (§5.4).

5.1 Experiment Setting

Implementation Details. The proposed network is implemented in PyTorch on a machine with Intel(R) i7-7700 3.6GHz processors with 8 threads and 64GB memory and a single Nvidia GeForce 3060Ti GPU with 8GB memory. For training we adopt standard SGD optimizer with no dropout, the learning rate is annealed geometrically starting at 1e-3 and decreases to 1e-5.

We adopt ResNet [17] pretrained on ImageNet [9] for the feature handling. The input RGB images are scaled to 341× 256 pixels, normalized by the subtraction of mean pixel values. The proposed network is pre-trained end-to-end on ScanNet [7], an RGB-D video sequence dataset which contains 2.5million views in over 1500 indoor scans, we only use the RGB monocular images and the ground truth camera pose values are given by [8]. The node poses (absolute pose) and edge poses (relative pose) are initialized as unit orientations and zero translations. We fix the input query size to be 32 though we have observed that the proposed network is capable of taking large input size up to 128. In all the experiments, the image frames are fed sequentially from the test set, analogous to existing work [6, 47, 48] for a fair comparison.

Datasets and Metrics. We conduct extensive experiments on datasets with different scales and report the median errors of camera orientation (°) and translation (m). The *7-Scenes dataset* [35] consists of RGB-D video sequences covering seven small indoor scenes, captured by hand-held Kinect camera. In some of the scenes, many texture-less surfaces and repetitive patterns are present, thus making the dataset challenging in spite of its relatively small size containing less than 10K images. The *Cambridge dataset* is a large-scale dataset containing

Table 1: Experiment results on the 7—Scenes Dataset [35]. Results are cited directly, the best results are **highlighted**.

Scene Scene scale	**Chess** 3 x $2m^2$	**Fire** 2.5 x $1m^2$	**Heads** 2 x $0.5m^2$	**Office** 2.5 x $2m^2$	**Pumpkin** 2.5 x $2m^2$	**Kitchen** 4 x $3m^2$	**Stairs** 2.5 x $2m^2$	**Avg.**
RelocNet [2]	0.12m, 4.14°	0.26m, 10.4°	0.14m, 10.5°	0.18m, 5.32°	0.26m, 4.17°	0.23m, 5.08°	0.28m, 7.53°	0.21m, 6.73°
LsG [47]	0.09m, 3.28°	0.26m, 10.92°	0.17m, 12.70°	0.18m, 5.45°	0.20m, 3.69°	0.23m, 4.92°	0.23m, 11.3°	0.19m, 7.47°
MapNet [4]	0.08m, 3.25°	0.27m, 11.69°	0.18m, 13.25°	0.17m, 5.15°	0.22m, 4.02°	0.23m, 4.93°	0.30m, 12.08°	0.21m, 7.77°
MapNet+ [4]	0.10m, 3.17°	**0.20m**, 9.04°	0.13m, 11.13°	0.18m, 5.38°	0.19m, 3.92°	0.20m, 5.01°	0.30m, 13.37°	0.19m, 7.29°
MapNet(pgo) [4]	0.09m, 3.24°	**0.20m**, 9.29°	**0.12m**, **8.45**°	0.19m, 5.42°	0.19m, 3.96°	0.20m, 4.94°	0.27m, 10.57°	0.18m, 6.55°
PoseNet15 [22]	0.32m, 8.12°	0.47m, 14.4°	0.29m, 12.0°	0.48m, 7.68°	0.47m, 8.42°	0.59m, 8.64°	0.47m, 13.8°	0.44m, 10.4°
PoseNet16 [20]	0.37m, 7.24°	0.43m, 13.7°	0.31m, 12.0°	0.48m, 8.04°	0.61m, 7.08°	0.58m, 7.54°	0.48m, 13.1°	0.47m, 9.81°
PoseNet17 [21]	0.14m, 4.50°	0.27m, 11.80°	0.18m, 12.10°	0.20m, 5.77°	0.25m, 4.82°	0.24m, 5.52°	0.37m, 10.60°	0.24m, 7.87°
PoseNet17+ [21]	0.13m, 4.48°	0.27m, 11.30°	0.17m, 13.00°	0.19m, 5.55°	0.26m, 4.75°	0.23m, 5.35°	0.35m, 12.40°	0.23m, 8.12°
LSTM+Pose [42]	0.24m, 5.77°	0.34m, 11.9°	0.21m, 13.7°	0.30m, 8.08°	0.33m, 7.00°	0.37m, 8.83°	0.40m, 13.7°	0.31m, 9.85°
Hourglass [28]	0.15m, 6.17°	0.27m, 10.84°	0.19m, 11.63°	0.21m, 8.48°	0.25m, 7.01°	0.27m, 10.15°	0.29m, 12.46°	0.23m, 9.53°
BranchNet [45]	0.18m, 5.17°	0.34m, 8.99°	0.20m, 14.15°	0.30m, 7.05°	0.27m, 5.10°	0.33m, 7.40°	0.38m, 10.26°	0.29m, 8.30°
VidLoc [6]	0.18m, —	0.26m, —	0.14m, —	0.26m, —	0.36m, —	0.31m, —	0.26m, —	0.25m, —
CNN+GNN [48]	**0.08m**, 2.82°	0.26m, 8.94°	0.17m, 11.41°	0.18m, 5.08°	**0.15m**, 2.77°	0.25m, 4.48°	**0.23m**, **8.78**°	0.19m, 6.33°
MS-Trans. [33]	0.11m, 4.66°	0.24m, 9.6°	0.14m, 12.19°	0.17m, 5.66°	0.18m, 4.44°	**0.17m**, 5.94°	0.26m, 8.45°	0.18m, 7.28°
GTCaR (ours)	0.09m, **1.94**°	0.27m, **8.45**°	**0.12m**, 9.34°	**0.12m, 2.41**°	**0.15m, 2.13**°	0.21m, **2.73**°	0.26m, 8.92°	**0.18m, 5.13**°

six outdoor scene scans outside the Cambridge University, the dataset consists of around 12K images and the corresponding camera pose ground truth.

The *Oxford RobotCar dataset* contains image sequences taken through driving in Oxford with different weathers, traffic conditions and lighting, the total trajectory is over 10km and is very challenging for camera re-localization. Following [4, 47, 48], we conduct experiments on the LOOP route (1120m) and FULL route (9562m) to evaluate the performance of the proposed network on long consecutive sequences. In all the experiments, we comply with the train/test split provided in the original 7-Scenes and Cambridge benchmarks, and that given in MapNet [4] for fair comparisons.

Baselines. The proposed network is evaluated against recent state-of-the-art camera re-localization networks, including single image-based absolute camera pose regression network PoseNet and its variants [20–22, 42] among which, LSTM+Pose [42] along with MapNet and its variants [4], LsG [47] and VidLoc [6] have utilized temporal inter-frame relations in the network. CNN+GNN [48] models the multi-view camera pose regression with a graph and leverages GNNs on the task. Other approaches include RelocNet [2], Hourglass [28] and BranchNet [45].

5.2 Performance Evaluation

7-Scenes. We first evaluate GTCaR on the 7-Scenes dataset against recent state-of-the-art approaches, the experiment results are given in Table. 1. It can be observed that our proposed network overperforms the other approaches on most of the scenes. Among the approaches, LsG [47], MapNet [4] and VidLoc [6] rely heavily on the temporal information of the input, *i.e.*, the approaches can handle consecutive sequences more efficiently but tend to lose the spatial inter-frame correlation especially for large-scale datasets or over long camera trajectories. Additionally, PoseNet [22] and its variants conduct absolute pose regression

Table 2: Experiment results on the Cambridge Dataset [22]. Evaluation with MapNet [4] is cited from [31], other results are cited directly. The average is taken on the first four datasets. The best results are **highlighted**.

Scene Scene scale	**College** 5.6x10^3 m^2	**Shop** 8.8x10^3 m^2	**Church** 4.8x10^3 m^2	**Hospital** 2.0x10^3 m^2	**Court** 8.0x10^3 m^2	**Street** 5.0x10^3 m^2	**Avg.**
MapNet [4]	1.07m, 1.89°	1.49m, 4.22°	2.00m, 4.53°	1.94m, 3.91°	7.85m, 3.76°	22.23m, 27.55°	1.63m, 3.64°
PoseNet15 [22]	1.66m, 4.86°	1.41m, 7.18°	2.45m, 7.96°	2.62m, 4.90°	-	-	2.04m, 6.23°
PoseNet16 [20]	1.74m, 4.06°	1.25m, 7.54°	2.11m, 8.38°	2.57m, 5.14°	-	-	1.92m, 6.28°
LSTM+Pose [42]	0.99m, 3.65°	1.18m, 7.44°	**1.52m**, 6.68°	1.51m, 4.29°	-	-	1.30m, 5.52°
PoseNet17 [21]	0.99m, 1.06°	1.05m, 3.97°	1.49m, 3.43°	2.17m, 2.94°	7.00m, 3.65°	20.70m, 25.70°	1.43m, 2.85°
PoseNet17+ [21]	0.88m, 1.04°	0.88m, 3.78°	1.57m, 3.32°	3.20m, 3.29°	6.83m, 3.47°	20.30m, 25.50°	1.63m, 2.86°
CNN+GNN [48]	0.59m 0.65°	0.50m, 2.87°	1.90m, 3.29°	1.88m, 2.78°	6.67m, 2.79°	14.72m, 22.44°	1.12m, 2.40°
MS-Trans. [33]	0.83m 1.47°	0.86m, 3.07°	1.62, 3.99°	1.81m, 2.39°	-	-	1.28m, 2.73°
GTCaR (ours)	**0.42m, 0.52°**	**0.64m, 1.56°**	1.55m, **2.56°**	**1.32m, 1.97°**	**5.62m, 2.17°**	**10.27m, 19.88°**	**0.98m, 1.65°**

from single images, such that the networks perform poorly on the scene where repetitive patterns or texture-less surfaces are present.

Similar to our proposed network, CNN+GNN [48] leverages graphs to model the multi-view camera re-localization with message passing among the image-embedded nodes. However, the network does not exploit temporal information in sequential images, and enforces a maximum value of neighbors of each node. As a result, it tends to miss the temporal correlation for consecutive frames or discard useful inter-frame spatial correlation. It is also noteworthy that the proposed approach achieves real-time performance for all the experiments, as we have observed the average runtime ranging from 12ms to 23ms per frame with the batch size set to be 32, while [48] records 8-batch performance with unknown runtime efficiency.

Cambridge. We demonstrate the capability to handle large-scale dataset of GTCaR by evaluating the network on the Cambridge dataset, where the proposed network outperforms the baselines on most of the scenes. Among the scenes, 'Court' and 'Street' are the largest datasets in size and cover long complex trajectories and huge outdoor areas, as challenging to handle with single image-based regression networks like PoseNet15, PoseNet16 and even LSTM+Pose with additional LSTM units, the aforementioned networks have not reported the results on these two datasets. It can be observed that GTCaR demonstrates great improvements over approaches solely relying on temporal relation or spatial relation on datasets with long camera trajectories.

RobotCar. The RobotCar dataset is especially challenging for the presence of weather variations, dynamic objects/pedestrians, occlusions, *etc*. Following [4, 22, 47, 48], we conduct experiments on the two subsets from the dataset. The LOOP route covers 1120m and the FULL route has a total length of 9562m. As PoseNet [22] conducts camera pose regression with heavy reliance on the visual information from singe images, large amounts of outliers are produced with insufficient inter-frame correlations, thus yielding low accuracy. MapNet [4] utilizes inputs from other sensors like GPS and IMU and fuses the measurements to aid the camera re-localization. Specifically, MapNet(pgo) acquires the relative

Table 3: Experiment results on the Oxford Robotcar Dataset [27]. Evaluation with PoseNet [22] is cited from [4], other results are cited directly, the best results are **highlighted**.

Scene Scene scale	**LOOP** 1120m	**FULL** 9562m
MapNet [4]	9.84m, 3.96°	41.4m, 12.5°
MapNet+ [4]	8.17m, 2.62°	30.3m, 7.8°
MapNet(pgo) [4]	6.73m, 2.23°	29.5m, 7.8°
PoseNet [22]	25.29m, 17.45°	125.6m, 27.1°
LsG [47]	9.19m, 3.52°	31.65m, 4.51°
CNN+GNN [48]	8.15m, 2.57°	17.35m, **3.47°**
GTCaR (ours)	**5.46m, 1.98°**	**14.37m**, 3.68°

camera pose from VO and acts in a sliding-window manner to predict the absolute poses. Compared with GNN-based approach [48], the proposed network shows major improvement as it efficiently models the spatiotemporal relation for sequential images, whereas the former network mainly relies on the spatial inter-frame dependencies.

Additionally, we report the cumulative distributions of the translation and rotation prediction errors on the two datasets against prior work in the *supplementary*. The baselines include PoseNet [22], MapNet [4], LsG [47] and CNN+GNN [48]. It can be observed that the proposed network outperforms the baselines on all the datasets.

5.3 Ablation Study

We conduct ablation study to investigate the significance of different modules of the proposed network. We show the ablation results on 'Pumpkin' scene from 7-Scenes dataset, 'Court' from the Cambridge and LOOP from the RobotCar dataset, to cover scenes of different scales and lengths. The results are given in Table. 4. The comprehensive ablation experiments are given in the *supplementary*.

We first evaluate GTCaR without the MPNN layers, such that the graph is directly fed into the graph Transformer layers without the node information aggregation aided by the adjacency tensor. It can be observed that the performance of the network is significantly worse on the 'Court' dataset. The reason is that the simple linear projection of the node features cannot preserve much information, compared with the message aggregated node features in the original network, where the neighboring node information is efficiently preserved. For the 'Pumpkin' scene, high amounts of repetitive patterns are present such that the graph is densely connected; For the LOOP route, the images are highly consecutive such that temporal Transformer can capture the neighboring node information along the temporal dimension. We then study the effects of the individual Transformer modules, *i.e.*, the experiments are conducted with GTCaR without graph Transformer layers (GTCaR[temporal]) and without temporal Transformer layers (GTCaR[graph]). It can be observed that the accuracy of

Table 4: Ablations on Pumpkin, Court and LOOP.

Configuration	Pumpkin	Court	LOOP
GTCaR	**0.15m, 2.13°**	**5.62m, 2.17°**	**5.46m, 1.98°**
GTCaR [temporal]	0.31m, 3.26°	6.95m, 2.95°	8.25m, 3.48°
GTCaR [graph]	0.17m, 2.28°	7.34m, 4.84°	9.03m, 3.55°
GTCaR [MPNN]	0.20m, **2.13°**	5.98m, 2.38°	7.95m, 2.64°

GTCaR[temporal] decreses harshly on 'Court' and 'Pumpkin' without the spatial correlation. Indeed, GTCaR can be seen as a GNN+RNN type of camera re-localization network, which can only preserve inter-frame dependencies over short period of time but tend to yield a overly sparse graph. On the other hand, the performance of GTCaR[graph] is slightly worse than the original network on all the datasets without significant decreased accuracy.

5.4 Discussions and Limitations

Generalizability. By virtue of utilizing the underlying geometric constraints implicitly, the proposed network can deliver higher accuracy and better robustness compared to its single-view APR counterparts. Nonetheless we have observed that the network generalizability to vastly different scenes is still limited, *i.e.*, the best performance is achieved by training the network on sets of similar scenes regarding indoor/outdoor, scale and lighting, *etc.*
Computational Cost. From the experiments and the ablation study, we have observed that the output graphs are mostly dense according to the spatiotemporal dependencies. The high density brings in high amounts of unnecessary computations, especially in the case where the scene scale is small and the camera motion is slow. Equipping more GNN layers after the Transformer layers can remove the unnecessary edges but tends to introduce over-fitting and graph memory overhead to the network.

6 Conclusion

In this paper we propose a neural network approach with a graph Transformer backbone, namely GTCaR, to address the multi-view camera re-localization problem. We model the multi-view camera pose regression problem with graph embedding, where the image features, camera poses and pair-wise camera transformations are fused into graph attributes. With the introduction of a novel adjacency tensor, the proposed network can effectively capture the local node connection information. By leveraging graph Transformer layers with edge features and enabling temporal Transformer to generate the spatiotemporal dependencies between the frames, GTCaR can actively gain the graph attention and achieves state-of-the-art robustness, accuracy and efficiency.

Acknowledgement. We thank all reviewers for valuable comments and suggestions. Ling was supported in part by US National Science Foundation Grants 2006665, 2128350 and 1814745.

References

1. Arrigoni, F., Fusiello, A., Ricci, E., Pajdla, T.: Viewing graph solvability via cycle consistency. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021) 2
2. Balntas, V., Li, S., Prisacariu, V.: Relocnet: Continuous metric learning relocalisation using neural nets. In: European Conference on Computer Vision (ECCV) (2018) 4, 11
3. Bertasius, G., Wang, H., Torresani, L.: Is space-time attention all you need for video understanding? Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 2
4. Brahmbhatt, S., Gu, J., Kim, K., Hays, J., Kautz, J.: Geometry-aware learning of maps for camera localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 4, 11, 12, 13
5. Chu, P., Wang, J., You, Q., Ling, H., Liu, Z.: Transmot: Spatial-temporal graph transformer for multiple object tracking. arXiv **abs/2104.00194** (2021), https://arxiv.org/abs/2104.00194 4
6. Clark, R., Wang, S., Markham, A., Trigoni, N., Wen, H.: 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) 2, 4, 10, 11
7. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: ScanNet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 10
8. Dai, A., Nießner, M., Zollhöfer, M., Izadi, S., Theobalt, C.: Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. ACM Transactions on Graphics (ToG) **36**(4), 1 (2017) 10
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: IEEE conference on computer vision and pattern recognition (CVPR). Ieee (2009) 10
10. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. NAACL (2019) 3
11. Ding, M., Wang, Z., Sun, J., Shi, J., Luo, P.: Camnet: Coarse-to-fine retrieval for camera re-localization. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2019) 4
12. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. International Conference on Learning Representations (ICLR) (2021) 2
13. Dwivedi, V.P., Bresson, X.: A generalization of transformer networks to graphs. DLG-AAAI (2021) 4, 9
14. Garg, R., Bg, V.K., Carneiro, G., Reid, I.: Unsupervised CNN for single view depth estimation: Geometry to the rescue. In: European Conference on Computer Vision (ECCV). Springer (2016) 4
15. Gidaris, S., Komodakis, N.: 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) 4
16. Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. In: International Conference on Machine Learning (ICML) (2017) 6

17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 6, 10
18. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation (1997) 4
19. Horn, B.K.: Closed-form solution of absolute orientation using unit quaternions. Josa a (1987) 1
20. Kendall, A., Cipolla, R.: Modelling uncertainty in deep learning for camera relocalization. In: IEEE international conference on Robotics and Automation (ICRA). IEEE (2016) 11, 12
21. Kendall, A., Cipolla, R.: Geometric loss functions for camera pose regression with deep learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 11, 12
22. Kendall, A., Grimes, M., Cipolla, R.: PoseNet: A convolutional network for real-time 6-dof camera relocalization. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2015) 2, 4, 10, 11, 12, 13
23. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. International Conference on Learning Representations (ICLR) (2017) 3
24. Klodt, M., Vedaldi, A.: Supervising the new with the old: learning sfm from sfm. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018) 4
25. Lerman, G., Shi, Y.: Robust group synchronization via cycle-edge message passing. Foundations of Computational Mathematics (2021) 5
26. Li, X., Ling, H.: Pogo-net: Pose graph optimization with graph neural networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2021) 2, 4, 5
27. Maddern, W., Pascoe, G., Linegar, C., Newman, P.: 1 Year, 1000km: The Oxford RobotCar Dataset. The International Journal of Robotics Research (IJRR) **36**(1), 3–15 (2017) 10, 13
28. Melekhov, I., Ylioinas, J., Kannala, J., Rahtu, E.: Image-based localization using hourglass networks. In: Proceedings of the IEEE international conference on computer vision workshops (ICCV Workshops) (2017) 2, 11
29. Purkait, P., Chin, T.J., Reid, I.: Neurora: Neural robust rotation averaging. In: European Conference on Computer Vision (ECCV). Springer (2020) 2, 5
30. Rong, Y., Huang, W., Xu, T., Huang, J.: Dropedge: Towards deep graph convolutional networks on node classification. International Conference on Learning Representations (ICLR) (2020) 4
31. Sattler, T., Zhou, Q., Pollefeys, M., Leal-Taixe, L.: 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) 2, 4, 12
32. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE Transactions on Neural Networks **20**(1), 61–80 (2008) 3
33. Shavit, Y., Ferens, R., Keller, Y.: Learning multi-scene absolute pose regression with transformers. Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2021) 4, 11, 12
34. Shi, Y., Lerman, G.: Message passing least squares framework and its application to rotation synchronization. In: International Conference on Machine Learning (ICML) (2020) 5

35. Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., Fitzgibbon, A.: 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) 10, 11
36. Tang, C., Tan, P.: BA-Net: Dense bundle adjustment networks. In: International Conference on Learning Representations (ICLR) (2018) 1, 4
37. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment—a modern synthesis. In: International Workshop on Vision Algorithms. Springer (1999) 1
38. Valada, A., Radwan, N., Burgard, W.: Deep auxiliary learning for visual localization and odometry. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) (2018) 4
39. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. Conference on Neural Information Processing Systems (NeurIPS) (2017) 2, 4, 7, 8
40. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. International Conference on Learning Representations (ICLR) (2018) 2, 3
41. Vijayanarasimhan, S., Ricco, S., Schmid, C., Sukthankar, R., Fragkiadaki, K.: Sfm-net: Learning of structure and motion from video. arXiv preprint arXiv:1704.07804 (2017) 4
42. Walch, F., Hazirbas, C., Leal-Taixe, L., Sattler, T., Hilsenbeck, S., Cremers, D.: Image-based localization using lstms for structured feature correlation. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017) 11, 12
43. Wang, X., Ji, H., Shi, C., Wang, B., Ye, Y., Cui, P., Yu, P.S.: Heterogeneous graph attention network. In: The World Wide Web Conference (WWW) (2019) 4
44. Wu, C., et al.: Visualsfm: A visual structure from motion system (2011) 1, 4
45. Wu, J., Ma, L., Hu, X.: Delving deeper into convolutional neural networks for camera relocalization. In: IEEE International Conference on Robotics and Automation (ICRA). IEEE (2017) 11
46. Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? International Conference on Learning Representations (ICLR) (2019) 2, 4
47. Xue, F., Wang, X., Yan, Z., Wang, Q., Wang, J., Zha, H.: Local supports global: Deep camera relocalization with sequence enhancement. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2019) 4, 10, 11, 12, 13
48. Xue, F., Wu, X., Cai, S., Wang, J.: 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) 2, 4, 10, 11, 12, 13
49. Yang, L., Li, H., Rahim, J.A., Cui, Z., Tan, P.: End-to-end rotation averaging with multi-source propagation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 5
50. Yun, S., Jeong, M., Kim, R., Kang, J., Kim, H.J.: Graph transformer networks. Conference on Neural Information Processing Systems (NeurIPS) (2019) 2, 4
51. Zhang, J., Zhang, H., Xia, C., Sun, L.: Graph-bert: Only attention is needed for learning graph representations. arXiv preprint arXiv:2001.05140 (2020) 2, 4
52. Zhao, L., Akoglu, L.: PairNorm: Tackling oversmoothing in gnns. In: International Conference on Learning Representations (ICLR) (2019) 4