Neurocomputing 521 (2023) 11–26

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Objects matter: Learning object relation graph for robust absolute pose regression

Chengyu Qiao [a], Zhiyu Xiang [b,*], Xinglu Wang [a], Shuya Chen [a], Yuangang Fan [a], Xijun Zhao [c]

[a] College of Information and Electronic Engineering, Zhejiang University, Hangzhou 310027, China
[b] Zhejiang Province Key Laboratory of Information Processing, Communication and Networking, Zhejiang University, Hangzhou310027, China
[c] China North Artificial Intelligence and Innovation Research Institute, China North Vehicle Research Institute, Beijing 100072, China

ARTICLE INFO

Article history:
Received 9 July 2022
Revised 5 October 2022
Accepted 27 November 2022
Available online 5 December 2022
Communicated by Zidong Wang

Keywords:
Absolute pose regression
Object relation graph
Graph neural networks

ABSTRACT

Visual relocalization aims to estimate the pose of a camera from one or more images. In recent years deep learning-based absolute pose regression (APR) methods have attracted many attentions. They feature predicting the absolute poses without relying on any prior built maps or stored images, making the relocalization very efficient. However, robust relocalization under environments with complex appearance changes and real dynamics remains very challenging. In this paper, we propose to enhance the distinctiveness of the image features by extracting the deep relationship among objects. In particular, we extract objects in the image and construct a deep object relation graph (ORG) to incorporate the semantic connections and relative spatial clues of the objects. We integrate our ORG module into several popular APR models. Extensive experiments on various public indoor and outdoor datasets demonstrate that our ORG module greatly enhances the robustness of image representation to environmental changes and improves the pose regression performance. The code is available at https://github.com/qcyay/ORGMapNet.

1. Introduction

Visual relocalization refers to the task of recovering the absolute pose of a camera from a single or multiple images, which is one of the fundamental tasks in lots of fields such as robotics, autonomous driving, and virtual/augmented reality. Due to the existence of large perspective and appearance changes, robust relocalization under real environment remains a challenging task.

Lots of effects have been made to deal with this problem. Structure-based algorithms [38,45,35,34,21] build a prior 3D map for the environment and relocalize the camera by establishing 2D-3D correspondences. They can achieve state-of-the-art performance, but require large memory space to store the map. Image retrieval based methods [1,36,47,18,3,10] search for the most similar images in the stored database and use the pose of the matched image as the recovered position. They can further be optimized by estimating the relative pose between the query and the matched images. Besides storing an image database, image retrieval based methods tend to fail when the database does not cover the novel pose of the query image.

In recent years, a series of absolute pose regression (APR) approaches [17,14,15,25,6,5,52,55] are proposed to directly predict the camera pose from an image, without relying on any prior built maps or stored image database. The pose of the camera can also be further improved by integrating constraints from multiple neighboring images with the help of visual odometry (VO) [4,48,56,32,7], Long-Short Term Memory Networks (LSTMs) [57,7,49] or Graph Neural Network (GNNs) [57]. Compared to the structure based or retrieval based counterparts, APR approaches learn the entire localization pipeline and are very efficient during runtime. They are an order of magnitude faster and require only a small memory footprint. However, current APR methods are also an order of magnitude less accurate than the structure-based approaches.

In this paper, we focus on improving the robustness of APR methods. No matter one or more images are used, current APR works mainly rely on convolutional network backbones similar to the ones used in image classification task to extract features of the image. However, different from the image classification problem where the pose and scale invariant features are desired, the ideal feature for the relocalization task should be pose and scale awareness while keeping invariance to the disturbances like illumination changes and dynamic objects.

We propose a novel object relation graph (ORG) to extract object relation feature and strengthen the inner representation of an image for relocalization. The objects in the image and the relationships among them can be important clues for the recovery of

* Corresponding author.
E-mail address: xiangzy@zju.edu.cn (Z. Xiang).

https://doi.org/10.1016/j.neucom.2022.11.090
0925-2312/

the camera pose in complex scenes, in that they can provide semantic stability and pose/scale awareness, as shown in Fig. 1. In most of the scenes, objects and the co-occurrence relationships among them are relatively stable. They do not change with the imaging projection, and can provide robust global localization of the camera on semantic space. Meanwhile, with the change of local position and view angle, the size of objects and the image distances among them change accordingly, which can facilitate local localization. When dealing with dynamic environments, certain dynamic object categories, e.g., vehicles and pedestrians, can be safely removed from the object relation graph, providing enhanced robustness to such scenes. We extract object features by employing Graph Neural Networks (GNNs). Specifically, we first detect objects in the image and obtain the primitive object features that contain spatial and semantic information. Then, the features are fed into a deep GNN where each node aggregates information from neighboring nodes with dynamic edge connections. To make the network more adaptive to the visual relocalization task, we redesign the nodes, edges, and the embedded functions of GNNs. Finally, we integrate our object relation graph model into PoseNet (with logq) [4] and MapNet [4], which are the popular single-frame and multi-frame based APR frameworks respectively. Both of the resulting frameworks are differentiable and can be trained end-to-end. The experimental results on multiple datasets demonstrate our success.

Our contributions can be summarized as:

- We explore the role of object relationship for the task of absolute pose regression. The object features with stable semantic co-occurrence and pose/scale awareness can provide vital clues to the task. To the best of our knowledge, we are the first to tackle this task by enhancing the visual representation with explicitly learned object relation features.
- We propose a novel object relation graph for object feature extraction, which utilizes deep multi-layer GNNs with dynamic edges to exploit semantic and spatial position relationships among objects.
- We integrate our ORG module into popular APR networks and carry out extensive experiments on various public datasets. The results show that our method has a significant improvement upon the baselines in terms of accuracy and robustness, and can achieve the state-of-the-art performance among APR methods.

2. Related work

2.1. Absolute pose regression

Absolute pose regression has become a popular method for relocalization in recent years. It does not require any prior map or stored image database during runtime. PoseNet [16] first introduce deep neural network to regress camera pose with a single image as input. Subsequent works [15,15,25,49,6,52,55,5,39] further extend this approach to improve the accuracy of pose regression. Hourglass-Pose [25] applies an encoder-decoder (hourglass) architecture containing a series of convolutional and up-convolutional layers to extract more representative features. [26] proposes a separate CNN-based architecture for camera pose estimation and employs data augmentation in 3D space to improve accuracy. LSTM-Pose [49] employs LSTM in image space to achieve more effective dimensionality reduction for features output from the CNN. [39] also utilizes LSTM along with increased field-of-view and data augmentation to improve the pose estimation accuracy of PoseNet.

The aforementioned methods only use the information from a single frame and are tend to suffer from visual ambiguity. In order to take advantage of sequential images, VidLoc [7] and ContextualNet [30] employ LSTMs to learn the temporal relationship between consecutive frames. MapNet [4] further applies the pose constraints between neighboring images and integrates visual odometry to improve localization accuracy. ViPR [28] incorporates an additional relative pose regression module into the APR process to improve absolute pose estimation accuracy. [38] summarizes the state-of-the-art APR methods and points out the limitation that APR methods tend to do interpolation between seen camera poses in the training set. Recently, LSG [56] and GRNet [57] use LSTMs and GNNs respectively to model the smoothness of motion within the image sequences. NeuralR-Pose [58] proposes to learn neural representations for camera poses and compute the pose loss in the learned neural space. MS-Trans [41] uses Transformers to

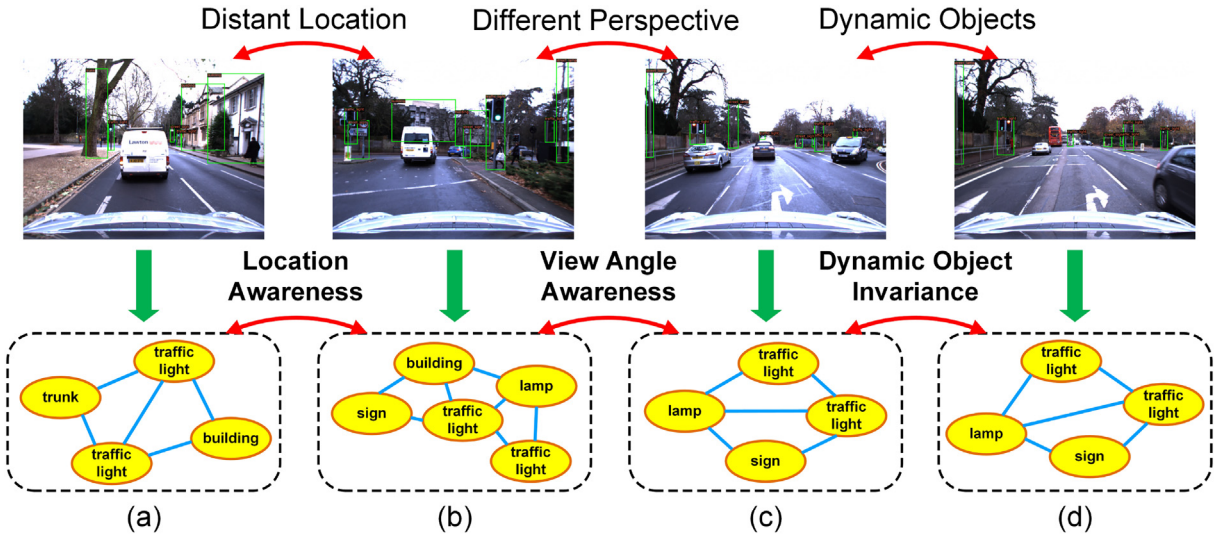

Fig. 1. Illustration of our proposed ORG features in different situations. Each image corresponds to a specific ORG that contains semantic connections and relative spatial cues for objects. For image pairs from distant locations shown in (a) and (b), the large difference of the ORG features makes it easy to distinguish their poses. For (b) and (c), the camera heads towards the same intersection from different routes, resulting in large perspective changes between images. In this situation, the co-occurrence relationship and the scale of objects in ORG can help to clarify the current camera pose. For (c) and (d), the relation features extracted from objects with certain static categories are immune to the dynamic objects, leading to more robustness to dynamic environments.

regress camera translation and rotation separately in multiple scenes.

Most of these APR methods acquire the representations of an image by commonly used convolutional backbone networks such as VGG [43], GoogLeNet [44] or ResNet [13], without paying attention to the objects and their relationship in the image. Instead, we explore object-level semantics through the object relation graph and extract a more discriminative representation from the image for the relocalization task.

2.2. Object related localization methods

There exists some approaches that utilize objects for localization. As an additional constraint, [34,20] incorporate the detected objects into the traditional Simultaneously Localization and Mapping (SLAM) pipeline to improve the robustness in the case of viewpoint changes. [23] extracts the object-level semantics to build a global semantic map, and estimates the pose of the query image within the global map by aligning the associated objects. [55,2] find 2D-2D dense matches between the detected objects and the objects in the database to localize the query image. Recently, some works use object detections as 2D projections of objects in the scene to estimate camera poses. QuadricSLAM [27] represents object-based landmarks as dual quadrics, and jointly estimates camera pose and quadric parameters in SLAM through 2D object detections. Gaudilliere et al. [12,13] propose to model objects of interests and object detections in the scene with a set of virtual ellipsoids and ellipses respectively, and recover the camera pose from the correspondences between 3D ellipsoids and 2D projections. These previous object based works either use geometrical object information in SLAM [34,20] or match the object to a prebuilt map [20] or databases [24,54]. In other words, they explicitly extract low level object features and need a map for matching to realize the localization. The APR task in our hand is much different from them in that APR directly output the absolute localization for the input image and does not built any map for matching. This makes the previous object based methods infeasible in APR. Different from previous methods that model objects with geometric representations like quadrics or ellipsoids, we directly encode object detections into learnable graph-based embeddings without any hand-engineering effort. Moreover, instead of matching objects separately, our method allows direct information exchange among objects via GNNs, which can effectively exploit the semantic cues within object relationships.

Object relationship has been employed in a few computer vision tasks such as scene graph generation [8]. It encodes the interplay between the object instances to achieve high-level image understanding to facilitate applications such as image captioning [53] and manipulation [9]. However, such kind of high-level relationship is too general and lacks of discrimination for the low-level camera relocalization task at hand.

3. Method

3.1. Acquiring primitive object feature

In order to take full advantage of the information of objects and capture the semantic and spatial relationships among objects, we first need to detect the objects of interest in the image. We leverage the Fully Convolutional One-Stage (FCOS) [46] which is a real-time object detection network to conduct object detection. Given an RGB image as input, FCOS detects the desired objects in the image and output their bounding box positions and labels. For n detected objects o_i where $i \in \{1, 2, \ldots, n\}$, we parameterize the detection as a tuple of $o = \{x, y, w, h, C\}$, where (x, y) denotes the pixel coordinates of the object center, (w, h) denotes the object size with its width and height, and C is the object category ID. Note that the objects of interest are different for indoor and outdoor scenes due to the huge difference between the two scenes. Objects that frequently appear in indoor scenes such as TVs or tables are usually static. However, objects like vehicles and people that are usually more interesting for outdoor object detection methods are dynamic and may have an adverse impact on the relocalization task. Thus, we train different object detectors for indoor and outdoor scenes and only detect those static objects such as tree trunks, buildings and traffic lights for outdoor scenes. The detection result o_i is fed into an MLP (Multi-Layer Perceptron) to obtain the primitive object feature x_i, which contains the object-level information and will be used as the node feature in the graph later, as shown in Fig. 2.

3.2. Graph definition

With the detected objects and the corresponding features at hand, we integrate them with the ORG. We first define the graph as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with $\mathcal{V} = \{v_1, \ldots, v_n\}$ and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ representing the nodes and edges respectively. Each detected object is regarded as a node v_i which contains the corresponding object feature x_i. Edge $e_{ij} = (v_i, v_j)$ denotes the link among different objects.

As each edge determines whether two objects are connected and allows them to exchange information, we need to define the edge connection in the initial stage. One direct way is to densely connect each pair of nodes, which will result in a fully-connected graph. However, it can lead to redundancy and expensive computational cost, especially when there are many objects detected in the image. In addition, full connection is not beneficial to capture the relationships among those objects that are really related. Therefore, we determine the existence of edges based on the distance between node features. Specifically, each node is connected to k nearest neighbors who have the smallest Euclidean distances to the current node in feature space, as shown in Fig. 2.

3.3. Learning the object relation graph

Given the nodes and edges of a graph, our goal is to learn a better object relationship through transferring information between nodes. For this purpose, we introduce the graph neural network to learn our object relation graph. Specifically, we employ four sequential node feature aggregation layers to propagate information among all object features, as shown in Fig. 3. Next, we introduce the feature update mechanism in the node feature aggregation layer.

3.3.1. Neighboring differential feature generation

In order to update the features of nodes, we first need to capture the relationship between nodes. For node v_i, we first calculate the feature distances between node v_i and the rest nodes v_j in graph $\mathcal{G}$. The distance d_{ij} is defined as:

$$d_{ij} = \left\| x_i - x_j \right\|_2 \tag{1}$$

Then we take k nodes $v_{j_{i1}}, \ldots, v_{j_{ik}}$ with the smallest distance to v_i, and establish k directed edges for node v_i in the form of $(i, j_{i1}), \ldots, (i, j_{ik})$. In this way, we construct $\mathcal{G}$ as a directed k-nearest neighbor (k-NN) graph. Given the neighboring node v_j, the neighboring differential feature n_{ij} for node v_i are computed as $n_{ij} = x_j - x_i$, which will be used in feature aggregation.

3.3.2. Node feature aggregation

Inspired by [31] who uses a symmetric function to aggregate point information, the node feature aggregation function is defined as:

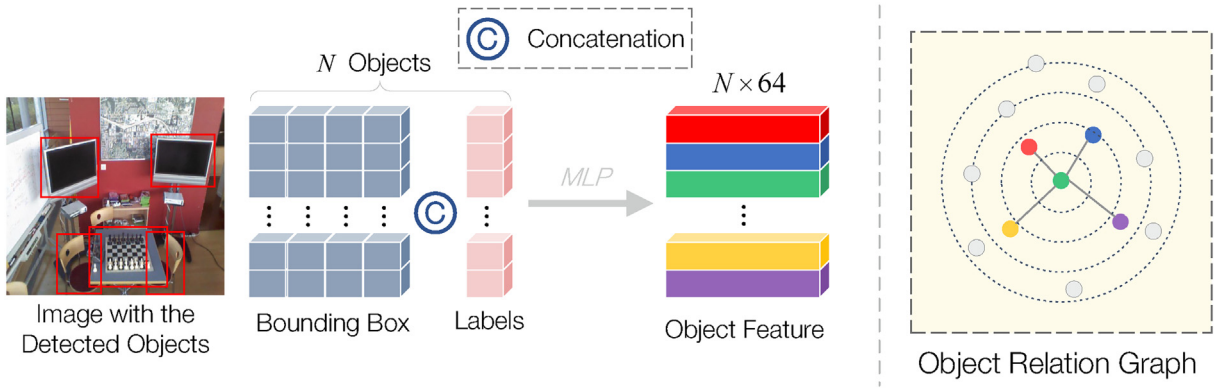

Fig. 2. Illustration of object relation graph. (left) Object detections containing bounding boxes and labels are extracted from the image and fed into a MLP to obtain primitive object features. (right) Object features are further modeled as the object relation graph, where the K nearest neighbors in feature space are connected to a node to construct the graph.

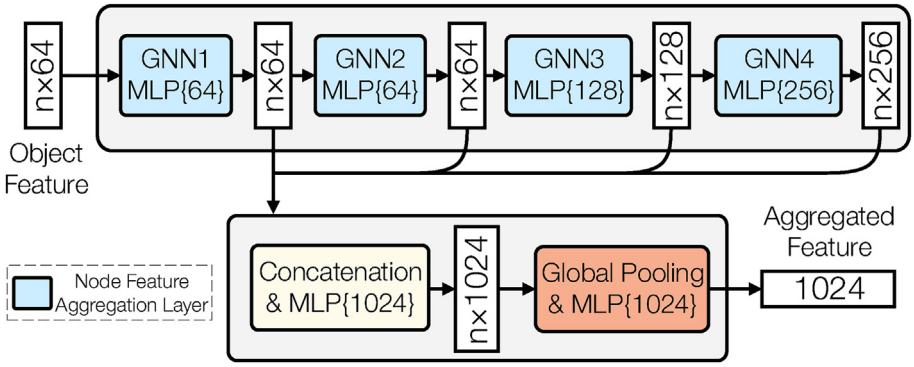

Fig. 3. The architecture of the model to learn our object relation graph. The detection features of n objects are fed to four consecutive node feature aggregation layers for information exchange. Fused features from the four layers are concatenated along the channel dimension, and the final aggregated features are obtained through two MLPs and global pooling. "MLP" stands for multi-layer perceptron, and numbers in brackets are layer sizes.

$$x_{i\prime} = \mathrm{MAX}_{j:(i,j)\in\mathscr{E}}\{f_u(x_i, n_{ij})\} \tag{2}$$

where f_u is a MLP followed by a BatchNorm layer and a ReLU non-linearity, and MAX is a vector max operator which takes k features as input and returns a new element-wise maximum vector. $x_{i\prime} \in \mathbb{R}^{d\prime}$ is the aggregated feature of node v_i, and $d\prime$ is the dimension of the resulting feature. x_i and n_{ij} are concatenated along the channel dimension before feeding into f_u. The resulting feature fuses object attributes x_i (contributed by the object positions and categories) and its local neighbor information n_{ij} to obtain a higher level of object representation.

3.4. Dynamic graph update

After passing through one node feature aggregation layer, the features of nodes in the graph are updated. Similar to applying the convolutional layer multiple times to obtain a larger receptive field, we apply the GNN layer multiple times so that nodes can obtain information from more other nodes. Different from the traditional GNNs in which edges are pre-defined and fixed, our ORG network dynamically defines edges at each layer according to the updated node features, resulting in a dynamic graph update. Let $X^l = \{x_1^l, \ldots, x_n^l\} \subseteq \mathbb{R}^{d_l}$ be the output of the l-th layer, the updated node feature of the $(l+1)$-st layer is

$$x_i^{l+1} = \mathrm{MAX}_{j:(i,j)\in\mathscr{E}^l}\left\{f_u^{l+1}(x_i^l, n_{ij}^l)\right\} \tag{3}$$

where $f_u^{l+1} : \mathbb{R}^{2d_l} \to \mathbb{R}^{d_{l+1}}$ is the corresponding MLP layer similar to Eq. 2.

As shown in Fig. 3, a four-layer GNN structure is designed in this paper to extract multi-level object features. At each GNN layer, k edges are dynamically constructed according to the updated features. The multi-level object features from four GNN layers are then concatenated and fed into a fully connected layer to obtain a fused feature x^a with dimension d_a. Finally a symmetry function is used to aggregate the information from all objects, as shown in Eq. 4.

$$x_g = f_g\left(\mathrm{MAX}_{i=1,\ldots,n}\{x_i^a\}, \mathrm{AVG}_{j=1,\ldots,n}\left\{x_j^a\right\}\right) \tag{4}$$

where $x_g \subseteq \mathbb{R}^{d_g}$ is the final object relation feature for the image, AVG is a vector average operator, which takes k features as input and returns a new element-wise averaged vector, and $f_g : \mathbb{R}^{2d_a} \to R^{d_g}$ is a MLP layer. In this step, the fused features are passed through a GMP (global max pooling) layer and a GAP (global average pooling) layer respectively for downsampling and then concatenated along the channel dimension, and finally fed into a fully connected layer to obtain a vector representing the object relation feature for the entire image.

3.5. Relocalization framework

Our object relation feature can serve as a complementary feature to the existing convolution based features in the relocalization task. Therefore, we embed our ORG module into the existing APR method. The resulting relocalization framework is shown in Fig. 4. We select two APR methods respectively as our baselines to construct the framework. The first baseline is PoseNet+ [4], a

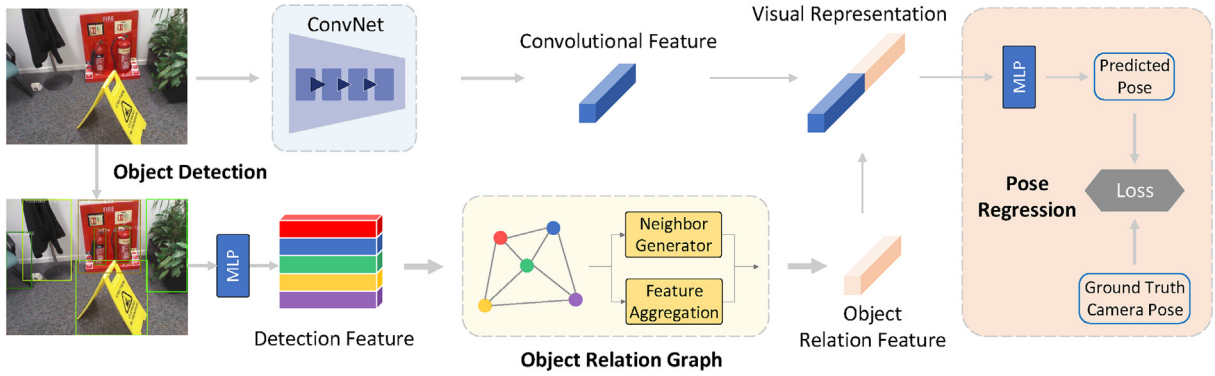

Fig. 4. Overview of the implemented APR framework. The upper branch is the baseline network and the lower branch is the pipeline which includes the proposed ORG module. Given the input image, we obtain object features based on object detections and model the features as an object relation graph. Consecutive node feature aggregation layers containing neighbor generator and feature aggregation are employed to learn the object relation graph. The convolutional feature and the object relation feature are concatenated before feeding into the pose regression head.

variant of PoseNet17 [15]. As an image-based APR network, it is based on PoseNet17 with ResNet34 as the backbone network and rotation parameterized as the logarithm of a unit quaternion to further improve the performance. For the methods using sequential images as input, MapNet [4] is a good choice. It follows the network structure of PoseNet+ and adds an additional loss term during training to further embed the multi-frame constraints. The resulting frameworks are called ORGPoseNet and ORGMapNet respectively. As shown in Fig. 4, the object relation feature from the ORG module and the convolutional feature from the baseline network are concatenated together before feeding into the pose regression head.

The pose regression head of the framework contains nothing but a fully connected layer followed by a dropout layer and two fully connected layers to predict the translation $\mathbf{t} \in \mathbb{R}^3$and rotation $\mathbf{r} \in SO(3)$of the image respectively. The predicted rotation term follows the form of MapNet, which is the logarithm of a unit quaternion.

3.6. Loss function

As in [16,4], the loss for ORGPoseNet is defined as:

$$L_{single} = \sum_{i=1}^{N} d(\mathbf{p}_i, \mathbf{p}_i^*) \tag{5}$$

where $\mathbf{p}_i = (\mathbf{t}_i, \mathbf{r}_i)$ and $\mathbf{p}_i^* = (\mathbf{t}_i^*, \mathbf{r}_i^*)$ are the predicted and the ground truth absolute poses, respectively. $d(\cdot)$ is a distance function defined as:

$$d(\mathbf{p}_i, \mathbf{p}_i^*) = \left\|\mathbf{t}_i - \mathbf{t}_i^*\right\|_1 e^{-\beta} + \beta + \left\|\mathbf{r}_i - \mathbf{r}_i^*\right\|_1 e^{-\gamma} + \gamma \tag{6}$$

where β and γ are learnable parameters used to balance the translation and rotation loss, and they are optimized jointly with other parameters in the neural network during the training process.

For ORGMapNet, the loss function also takes the relative pose loss into account. It is defined as in [4]:

$$L_{frame} = \sum_{i=1}^{N} d(\mathbf{p}_i, \mathbf{p}_i^*) + \sum_{i,j=1, i\neq j}^{N} d(\mathbf{v}_{ij}, \mathbf{v}_{ij}^*) \tag{7}$$

where $\mathbf{v}_{ij} = (\mathbf{t}_i - \mathbf{t}_j, \mathbf{r}_i - \mathbf{r}_j)$ is the relative pose between the predicted pose $\mathbf{p}_i$ for the image I_i and the predicted pose $\mathbf{p}_j$ for the image I_j. (I_i, I_j) is an image pair in each tuple of s images, and these images are sampled at intervals of k_f frames in the training set.

4. Experiments

4.1. Implementation details

For the backbone branch, we follow MapNet [4] to extract convolution features with 1024-dimension. In the ORG branch, the graph is constructed using $k = 5$ nearest neighbors. The dimension of the final output object relationship feature x_g is also 1024. For pose regression head, we use a fully connected layer with hidden feature dimension of 2048 $\Rightarrow$ 512, followed by two fully connected layers with dimension 512 $\Rightarrow$ 3 for translation and rotation respectively.

We use PyTorch [29] to implement our model on a single GTX 2080Ti GPU. As in [16,4], all input RGB images are resized with the height to 256 and normalized by pixel mean subtraction and standard deviation division. We set β and γ to 0.0 and -3.0 for initialization, respectively. For ORGMapNet, the number of input images s and sampling interval k_f of the image sequence are set to 3 and 10, respectively, as in [4]. We use the Adam optimizer [17] with a weight decay of 5×10^{-4}. The learning rate of the backbone network is set to 1×10^{-4} throughout the training process, while the learning rate of other parts is 1×10^{-3} at the beginning and divided by 10 at the 300-th epoch. All models are trained for 600 epochs with batch size of 64 and 20 for ORGPoseNet and ORGMapNet, respectively. The resulting network can run at a rate of 51.6 FPS with a single GPU during inference.

4.2. Settings

4.2.1. Datasets

We evaluate our algorithm on three well-known public datasets with different environmental scales, including 7-Scenes [42], RIO10 [50] and Oxford RobotCar [24].

The 7-Scenes dataset [42] is an indoor dataset containing seven different scenes recorded by a Kinect RGB-D sensor. The ground-truth camera poses are obtained by applying the KinectFusion system. Each scene contains a specific room in an office building, in which different trajectories are recorded to obtain multiple sequences. The scenes contain many textureless regions, which is not easy for visual relocalization.

The RIO10 dataset [50] contains ten RGB-D image sequences of indoor scenes captured by a mobile phone. Each scene is visited multiple times over a period of up to one year. The groundtruth camera poses are obtained by an offline bundle adjustment frame-

work. In addition to RGB and depth images, the dataset also provides semantic maps for reference. Dynamic objects, changes in long-term spans, blurring and illumination changes captured by the mobile phone make this dataset very challenging for relocalization task.

The Oxford RobotCar dataset [24] contains over 100 repetitions of a continuous route through Oxford over a year. The images are collected from a camera mounted on a vehicle. Compared with the indoor datasets, this dataset contains much longer trajectories and larger areas, which brings greater challenges to the relocalization methods due to the complex dynamic outdoor environment. In addition, the dataset images are collected under different lighting and weather conditions, which makes it an ideal choice for evaluating the method in real outdoor scenes.

4.2.2. Evaluation metrics

Following [16], we use the median absolute pose errors as the evaluation metric for the 7-Scenes and RIO10 datasets. For the Oxford RobotCar dataset, we employ both median and mean absolute pose errors as metrics. In addition, in the RIO10 dataset, we further adopt their proposed new metrics such as Dense Correspondence Re-Projection Error (DCRE), Outlier and Score. DCRE is calculated as the average magnitude of the 2D correspondence displacement, normalized by the image diagonal. The displacement is calculated according to the 2D projection of dense 3D points rendered from an underlying 3D model using ground truth and the predicted camera poses. DCRE(0.05) and DCRE(0.15) represent the proportion of frames with DCRE lower than 0.05 and 0.15 respectively, while Outlier(0.5) is the proportion of frames whose DCRE is above 0.5. The Score metric is defined as the sum of 1 and the difference between DCRE(0.15) and Outlier(0.5) to represent the general quality of the result.

4.3. Experiments on the 7-scenes dataset

To illustrate the effect of our ORG module on pose regression, we compare our implemented models respectively with the two baseline, as well as some other state-of-the-art methods on 7-Scenes dataset. For the image-based baseline PoseNet+ [4], the results of its variation versions are also listed. The quantitative results in Table 1 show that our ORGPoseNet outperforms the baseline PoseNet+ in both translation and rotation prediction, especially in repetitive and weak texture scenes, e.g., Stairs and Heads. This is because by constructing the object relation graph, our method learns a more robust image representation for changes in illumination and viewing angles, thus contributing to pose estimation. When introducing temporal information, our ORGMapNet further improves the performance upon the sequence-based method MapNet. Both accuracy gains over image-based and sequence-based baselines demonstrate the effectiveness of our proposed ORG module. More detailed results of this experiment can be found in the B.

4.4. Experiments on the RIO10 dataset

Containing the indoor images recorded over a year, the RIO10 dataset [50] is suitable for testing the method on long-term relocalization task. We compare our method with the baselines as well as some representative image retrieval methods [1,47]. As shown in Table 2, our ORGMapNet outperforms other methods on all of the metrics. In particular, Outlier(0.5) error is brought down by 17.05% and Score is raised by about 16% for our ORGMapNet upon the baseline. This should thank to the object relation graph which captures the location and semantic relationships of objects in the scene. The enhanced object relation features improve the robustness to appearance changes caused by long-term environmental variation. It illustrates the effectiveness of our proposed ORG module in dynamic indoor environments.

4.5. Experiments on the RobotCar dataset

We evaluate our method on the Oxford RobotCar dataset which consists of challenging 9562 m long FULL sequences covering an area of 1.2×10^6 m^2. All models are trained with two sequences captured on the FULL route under overcast weather, while the test sequences are captured in various weather conditions with long time spans. We report both median and mean pose errors on all of the test sequences. Median errors focus on reflecting the general accuracy of the model, while mean errors are more impacted by outliers, representing the overall robustness of the method.

Table 3 shows the quantitative results comparing with the baselines and other state-of-the-art methods. Firstly, our methods significantly improve the performance upon the baselines, in that the corresponding average median translation errors are both reduced by about 66%. Secondly, in FULL3-5 sequences with different weather and long time spans, we can observe the degraded performance for all models. It reveals the great difficulties of the relocalization task posed by these challenging conditions. In the meantime, compared with the baselines, our methods achieve much lower error variances and exhibit more robustness to the changing environment. We believe this is due to the explicit integration of object relation features to the common convolutional visual features. Thirdly, our ORGMapNet also outperforms multi-frame methods like LSG [56], AtLoc+ [51] and GRNet [57] on mean error in the available FULL1-2 sequences. It is worth noting that methods like LSG and GRNet predict camera pose by aggregating at least 7 images, while our ORGMapNet only takes 3 frames as input. On the other hand, our method is complementary to these multi-frame methods, as it aims to learn a more discriminative feature representation within an image while LSG and GRNet focus on fusing the information of multiple images by LSTM or GNN models.

Table 1
Comparison of the pose errors on the 7-Scenes dataset. The column PoseNet+ (*) are the results of running the code provided by [4]. Following the convention, the median prediction errors are reported here. The best results are highlighted.

Method	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs	Average
PoseNet17 [15]	0.14 m, 4.50°	0.27 m, 11.80°	0.18 m, 12.10°	0.20 m, 5.77°	0.25 m, 4.82°	0.24 m, 5.52°	0.37 m, 10.60°	0.24 m, 7.87°
GeoPoseNet [15]	0.13 m, 4.48°	0.27 m, 11.30°	0.17 m, 13.00°	0.19 m, 5.55°	0.26 m, 4.75°	0.23 m, 5.35°	0.35 m, 12.40°	0.23 m, 8.12°
PoseNet+ [4]	0.11 m, 4.29°	0.27 m, 12.13°	0.19 m, **12.15**°	0.19 m, 6.35°	0.22 m, 5.05°	0.25 m, 5.27°	0.30 m, 11.29°	0.22 m, 8.07°
PoseNet+ (*)	0.17 m, 4.96°	0.36 m, 11.22°	0.20 m, 13.35°	0.23 m, 7.05°	0.26 m, 5.87°	0.29 m, 6.10°	0.36 m, 10.18°	0.27 m, 8.39°
IRPNet [40]	0.13 m, 5.64°	**0.25 m**, 9.67°	**0.15 m**, 13.10°	0.24 m, 6.33°	0.22 m, 5.78°	0.30 m, 7.29°	0.34 m, 11.60°	0.23 m, 8.49°
NeuralR-Pose [58]	0.12 m, 4.83°	0.27 m, **8.91**°	0.16 m, 12.84°	0.19 m, 6.64°	0.22 m, 5.45°	0.24 m, 6.10°	0.29 m, 10.70°	0.21 m, 7.92°
ORGPoseNet (Ours)	0.10 m, 3.29°	0.33 m, 11.02°	**0.15 m**, 13.34°	0.19 m, 5.91°	0.20 m, 5.42°	0.24 m, 5.71°	0.27 m, 10.63°	0.21 m, 7.90°
MapNet [4]	**0.08 m**, **3.25**°	0.27 m, 11.69°	0.18 m, 13.25°	**0.17 m**, 5.15°	0.22 m, **4.02**°	0.23 m, **4.93**°	0.30 m, 12.08°	0.21 m, 7.77°
ORGMapNet (Ours)	0.09 m, 3.60°	0.26 m, 9.49°	**0.15 m**, 12.81°	0.20 m, **4.96**°	**0.18 m**, 5.04°	**0.22 m**, 5.68°	**0.27 m**, **9.54**°	**0.20 m**, **7.30**°

Table 2
Comparison on the validation split of the RIO10 dataset w.r.t. the average score, DCRE errors, median pose errors and outlier ratio. The best results are highlighted.

Method	Score ↑	DCRE(0.05) ↑	DCRE(0.15) ↑	$(\Delta t, \Delta\theta)$ ↓	Outlier(0.5) ↓
NetVLAD [1]	0.673	0.006	0.125	(0.93 m, 31.44°)	0.452
DenseVLAD [47]	0.604	0.008	0.124	(0.98 m, 32.26°)	0.520
PoseNet+ [4]	0.624	0.012	0.149	(0.89 m, 41.89°)	0.525
ORGPoseNet (Ours)	0.662	0.010	0.153	(0.86 m, 37.23°)	0.491
MapNet [4]	0.682	0.030	0.192	(0.84 m, 37.43°)	0.510
ORGMapNet (Ours)	**0.791**	**0.064**	**0.214**	(**0.80 m**, **30.83**°)	**0.423**

Table 3
The pose errors on the Oxford RobotCar dataset. "N" indicates the number of images input to the network at one time. We report the median/ mean translation and rotation errors. The best results are highlighted.

Method	N	Scene					
		FULL1 overcast	FULL2 overcast	FULL3 rain	FULL4 sun	FULL5 overcast, 1 year	Average
PoseNet+ [15]	1	107.62 m/125.6 m, 22.49°/27.1°	101.94 m/131.06 m, 20.00°/26.05°	81.31 m/151.87 m, 15.71°/35.17°	161.92 m/213.62 m, 29.67°/42.07°	115.47 m/165.23 m, 27.16°/34.95°	113.65 m/157.48 m, 23.01°/33.07°
ORGPoseNet (Ours)	1	26.58 m/53.08 m, 4.65°/10.18°	28.33 m/72.36 m, 4.59°/16.15°	38.45 m/160.58 m, 6.25°/33.72°	43.74 m/127.34 m, 7.14°/24.72°	40.71 m/129.75 m, 7.98°/24.51°	35.56 m/108.62 m, 6.12°/21.86°
MapNet [4]	3	18.21 m/41.40 m, 6.66°/12.50°	21.16 m/59.30 m, 6.38°/14.81°	45.51 m/167.75 m, 10.00°/35.56°	76.51 m/208.11 m, 13.62°/32.15°	32.44 m/107.54 m, 9.22°/28.98°	38.77 m/116.82 m, 9.18°/24.82°
MapNet + PGO [4]	3	–/29.5 m, –/7.8°	–	–	–	–	–
LSG [56]	7	–/31.65 m, –/4.51°	–/53.45 m, –/8.60°	–	–	–	–
AtLoc+ [51]	3	**6.40 m**/21.0 m, 1.50°/6.15°	**7.00 m**/42.6 m, 1.48°/9.95°	–	–	–	–
GRNet [57]	8	–/17.35 m, –/**3.47**°	–/37.81 m, –/**7.55**°	–	–	–	–
ORGMapNet (Ours)	3	8.84 m/**14.41 m**, **1.13**°/3.73°	8.01 m/**36.61 m**, **1.26**°/9.49°	**15.15 m**/**103.42 m**, **2.58**°/**20.99**°	**14.86 m**/**75.46 m**, **1.79**°/**12.30**°	**16.85 m**/**95.03 m**, **2.09**°/**16.85**°	**12.74 m**/**64.99 m**, **1.77**°/**12.67**°

It will be interesting to explore the combination of these ideas in future.

The trajectories obtained by the different models are further illustrated in Fig. 5. With a single image as input, PoseNet+ obtains inaccurate estimations with lots of outliers due to scene similarities and over-exposure. In contrast, ORGPoseNet reduces outliers and achieves higher accuracy which is comparable to the sequence based method MapNet. Introducing relative poses between consecutive frames as constraints, MapNet has relatively good pose predictions on the trajectory. However, due to the complexity of the environment, it still produces a large number of unstable predictions. In contrast, our proposed ORGMapNet significantly reduces the number of outliers and outperforms the baseline MapNet. This is because we construct the object relation graph on the static objects and enhance the discrimination of image representation by object relation features.

To further illustrate the effectiveness of our introduced ORG module, we depict the feature changes of sequential images under different weather conditions. Specifically, we select some subsequences where the camera remains static and calculate the feature distances (L2) between the first and the subsequent images. The ideal change of the features should be very small. As we can see in Fig. 6 (left), PoseNet+ is very sensitive to moving vehicles, resulting in very large variations in feature distances. On the contrary, ORGPoseNet obtains more stable features and exhibits less sensitivity to dynamic objects. At the right side of Fig. 6, rain weather (unlike overcast for the training sequences) brings blur and strong glare, making relocalization more challenging. In this case, the features extracted from PoseNet+ are perturbed with the presence of cyclists and produce drastic changes under strong glare. Subsequent dynamic vehicles continue to cause variations in feature space even they occupy only a small region of the image. In contrast, by introducing object-level features as a complement using the ORG module, ORGPoseNet obtains much less sensitive features to these perturbations. Feature distances remain surprisingly low in challenging scenarios, demonstrating the effectiveness of our ORG module.

Some examples of the object nodes as well as the edges in different scenes are illustrated in Fig. 7. Benefiting from the constructed dynamic graph, our ORG can capture the relationships among objects and generate a discriminative representation for the relocalization task.

4.6. Ablation Study

We conduct ablation studies on RobotCar dataset to verify the configurations of our design. As shown in Table 4, compared to the baseline MapNet [4], pose errors start to decrease progressively with the increase of GNN layers. A drastic improvement happens when 3rd layers of GNN is added. Increasing the number of layers from 4 to 6 can further reduce part of the error but with more computing cost. Adding to 8 layers does not reduce the error and causes a decrease in performance. This is because too deep layers may suffer from the vanishing gradient problem [19]. Therefore, we choose the layer number of GNN as 4. If we reduce the number of nearest neighbor from 5 to 3, or change the aggregate mode of each layer from Max to Sum, the pose error will both increase accordingly. Increasing the kNN neighbors to 9 also leads to a drop in performance. This may due to the actual limited number of objects in an image, and using too many neighbors in ORG can lead to over-smoothing of the ORG feature. We also test the effectiveness of our dynamic graph update strategy by replacing it with the fixed edge connection from the first layer, the degraded performance verify the necessity of this strategy. Finally, to verify the robustness of the learned object feature with respect to the degraded detecting capability, we manually drop out a number of the detected objects with a drop rate and only use the rest as input to the

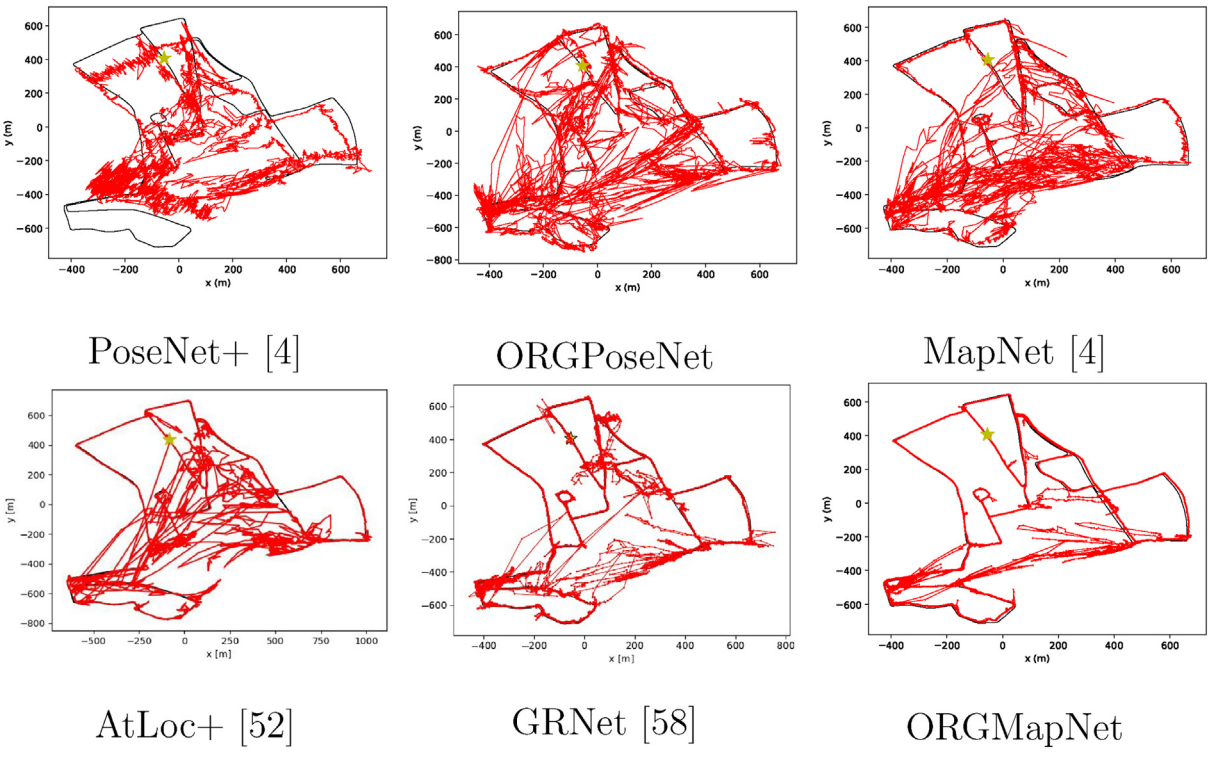

Fig. 5. Comparison of trajectories on the FULL1 scene of Oxford RobotCar dataset [24]. The red and black lines represent the predicted and groundtruth camera poses, respectively.

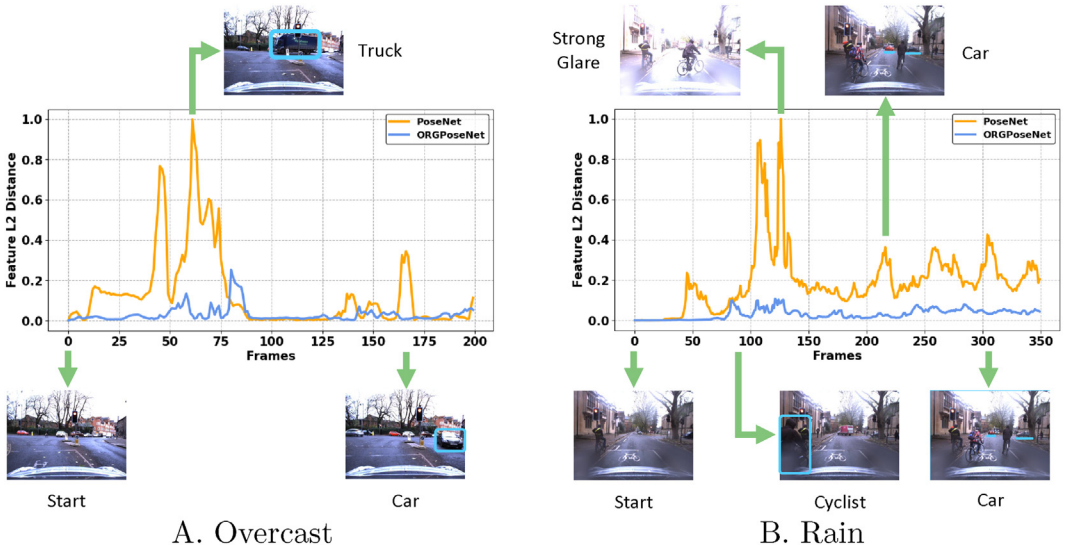

Fig. 6. Comparisons on feature stability under various weather conditions. (a) Overcast and (b) Rain. Features from ORGPoseNet demonstrate robustness to perturbations including dynamic objects, challenging weather, and illumination changes, while PoseNet+ suffers in all cases.

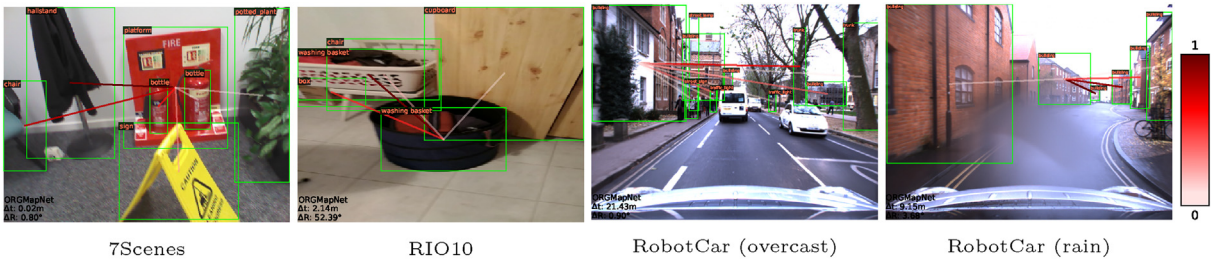

Fig. 7. Examples of object nodes and the edges in different dataset. For clarity, only the edges for one object are shown, with the deeper color encodes the smaller distance of objects.

Table 4
Ablation study on RobotCar dataset. The median pose errors are reported with the best results highlighted.

	Configuration	Scene	
		FULL1	FULL2
GNN Layers	0 (MapNet)	18.21 m, 6.66°	21.16 m, 6.38°
	1	17.14 m, 3.26°	17.11 m, 2.80°
	2	15.17 m, 2.36°	17.14 m, 2.66°
	3	9.92 m, 1.25°	9.28 m, 2.36°
	4 **(ORGMapNet)**	8.84 m, **1.13**°	**8.01 m**, 1.26°
	6	**8.59 m**, 1.41°	8.20 m, **1.25**°
	8	9.23 m, 1.51°	8.63 m, 1.35°
kNN Neighbors	$k = 3$	10.44 m, 1.35°	12.50 m, 1.41°
	$k = 5$ **(ORGMapNet)**	**8.84 m**, **1.13**°	**8.01 m**, **1.26**°
	$k = 9$	9.01 m, 1.36°	8.44 m, 1.53°
Aggregate Mode	max→sum	11.76 m, 1.55°	12.50 m, 1.61°
Update Mode	dynamic→static	9.93 m, 1.54°	10.22 m, 1.80°
Objects Detected	drop rate = 0.8	14.89 m, 4.59°	17.76 m, 4.65°
	drop rate = 0.4	9.92 m, 2.08°	8.66 m, 2.05°
	drop rate = 0.2	9.05 m, 1.46°	8.29 m, 1.62°
	drop rate = 0 **(ORGMapNet)**	**8.84 m**, **1.13**°	**8.01 m**, **1.26**°

GNN. The performance increases with the lower drop rate. Surprisingly, dropping 40% of the detected objects can achieve close performance with using all of the objects, demonstrating the robustness of our method to the object detector.

5. Conclusion and future work

In this paper, we propose to improve the absolute pose regression performance by fusing the object relation feature. We construct an ORG module for the image, and capture the object relation feature by dynamic GNNs. Comparing with the common convolutional features, our object relation feature leverages the semantic stability and the pose awareness which are highly desirable for the relocalization task. We integrate our ORG module to two popular baseline models to verify the effectiveness of idea. Extensive experiments on multiple public datasets are conducted and the results demonstrate that our method greatly enhances the robustness of the feature under various disturbances and is able to achieve very competitive pose regression performance.

Our future plan is to combine our method with the recent work of 3D object detection, which can provide more advanced structural information about objects. Our model can inherently adapt to this object representation and potentially generate more robust image features for absolute pose regression.

CRediT authorship contribution statement

Chengyu Qiao: Conceptualization, Methodology, Software, Writing – original draft. **Zhiyu Xiang:** Conceptualization, Resources, Writing – review & editing, Supervision, Funding acquisition. **Xinglu Wang:** Data curation, Methodology, Software. **Shuya Chen:** Resources, Validation. **Yuangang Fan:** Conceptualization, Data curation, Software. **Xijun Zhao:** Conceptualization.

Data availability

Data will be made available on request.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the NSFC-Zhejiang Joint Fund for the Integration of Industrialization and Informatization under Grant U1709214, in part by the Key Research & Development Plan of Zhejiang Province underGrant 2021C01196, and in part by the Fund of Science and Technology on Aerospace Flight Dynamics Laboratory under Grant 2022-JYAPAF-F1027.

Appendix A. Implementation

A.1. Object detection

Since the accuracy of object detection results is directly related to the performance of our model, it is important to train a robust object detector in complex environments. We use FCOS [46] under different training settings to detect objects in indoor and outdoor scenes. Specifically, we use FCOS pretrained on the COCO dataset [22] to detect objects of interest on the 7-Scenes dataset [42]. For the RIO10 dataset [50], as it provides groundtruth semantic maps, we obtain the groundtruth boxes from the semantic images and retrain FCOS. We cannot directly utlize the FCOS detection results on the Oxford RobotCar dataset [24], as the detected objects such as vehicles and people are not beneficial for the relocalization task. Thus, we select 300 images in the training split of the FULL scene that contains 70,275 images and annotate objects with seven labels: buildings, street lights, traffic lights, street signs, benches, postboxes, and tree trunks. Then we use the labeled images to train FCOS.

A.2. Training sequence

We use the officially released training sequences to train our networks on the 7-Scenes [42] and RIO10 datasets [50]. For the Oxford RobotCar dataset [24], we follow the training/test split of the FULL scene used in MapNet [4] and GRNet [57]. In addition, we add three sequences with various weather conditions and long time spans to the test split. The specific training and testing sequences with their descriptions are illustrated in Table A.5. By this way, we compare our method with two baseline methods, PoseNet+ [4] and MapNet [4], as well as other state-of-the-art methods such as LSG [56] and GRNet [57] on this dataset.

Table A.5
Sequences and corresponding descriptions on the Oxford RobotCar dataset [24]. We adopt the same training/test split as PoseNet+ [4] and MapNet [4] and add additional test sequences to illustrate the effectiveness of our method in complex scenes.

	Sequence	Tag	Train	Test
–	2014-11-28-12-07-13	overcast	✓	
–	2014-12-02-15-30-08	overcast	✓	
FULL1	2014-12-09-13-21-02	overcast		✓
FULL2	2014-12-12-10-45-15	overcast		✓
FULL3	2014-11-25-09-18-32	rain		✓
FULL4	2014-12-16-09-14-09	sun		✓
FULL5	2015-10-30-13-52-14	overcast		✓

Appendix B. Experimental results

B.1. Experiments on the 7-scenes dataset

Fig. B.8, Fig. B.9 and Fig. B.10 show the recovered trajectories of PoseNet+ [4], MapNet [4] and our method for the testing sequences on the 7-Scenes dataset. Samples from the testing sequences are also plotted to illustrate the challenging conditions on this dataset. Following [4], we also calculate a series of statistics on all the testing sequences, as shown in Table B.6. The Avg Median (Scene), Avg Median (Seq), and Avg Median (Image) denote the average values of the median error over scenes, sequences, and images, respectively. From the figures and the table, we can see that our method obtains more accurate pose estimation results than the baseline methods due to the utilize of object features.

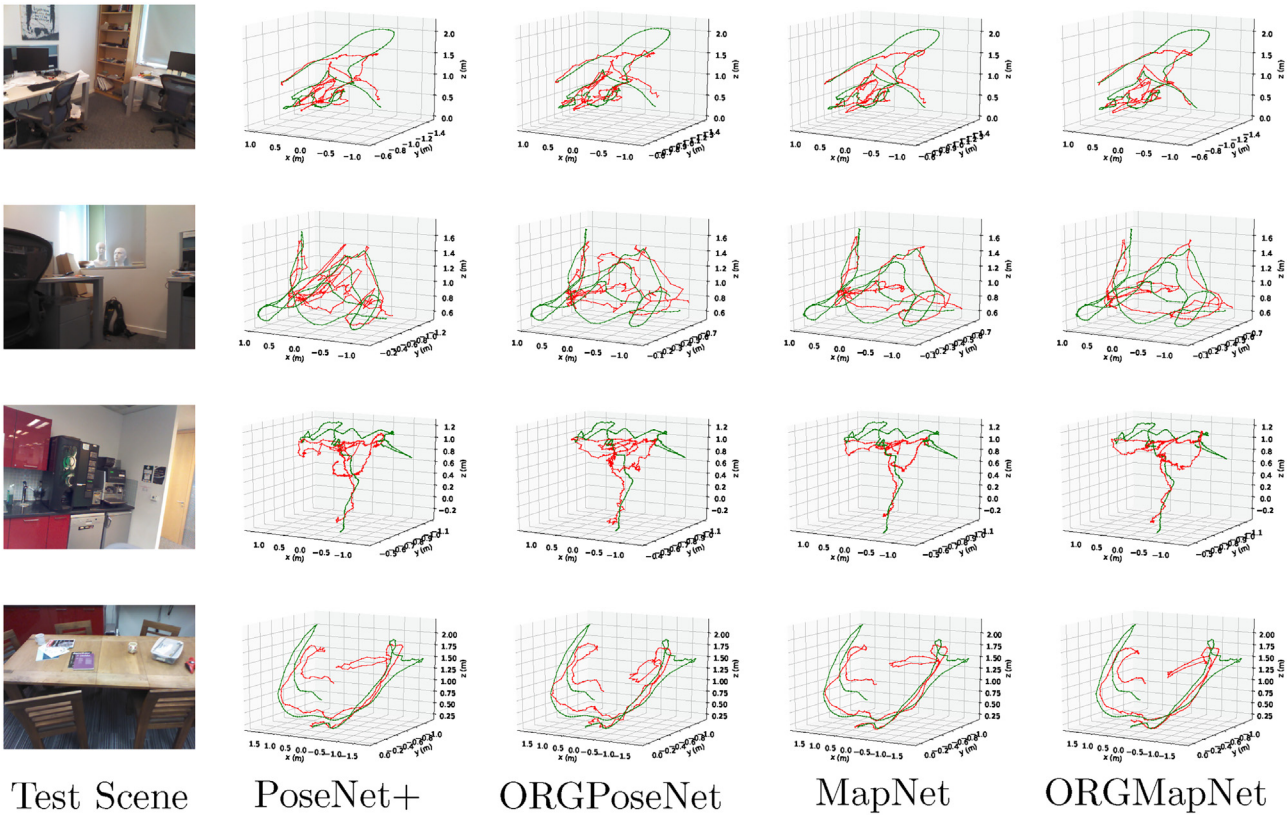

Fig. B.8. Raw images and the recovered trajectory of PoseNet, MapNet, and our method on the 7-Scenes dataset. The green and red lines are the groundtruth and predicted trajectories, respectively. The testing sequences (from top to bottom) are office-07, office-09, pumpkin-01 and redkitchen-12.

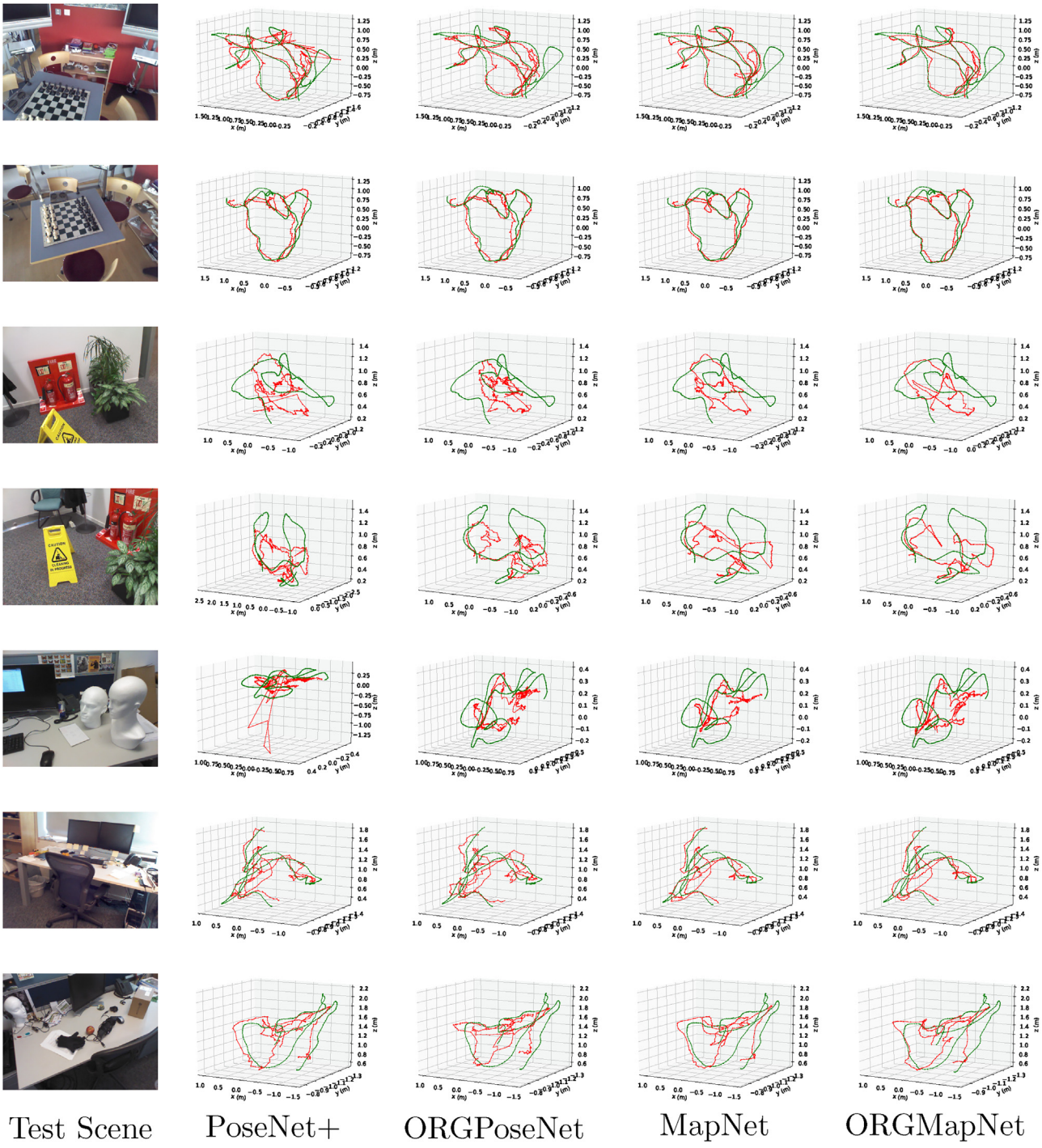

Fig. B.9. Raw images and the recovered trajectory of PoseNet+, MapNet, and our method on the 7-Scenes dataset. The green and red lines are the groundtruth and predicted trajectories, respectively. The testing sequences (from top to bottom) are chess-03, chess-05, fire-03, fire-04, heads-01, office-02 and office-06.

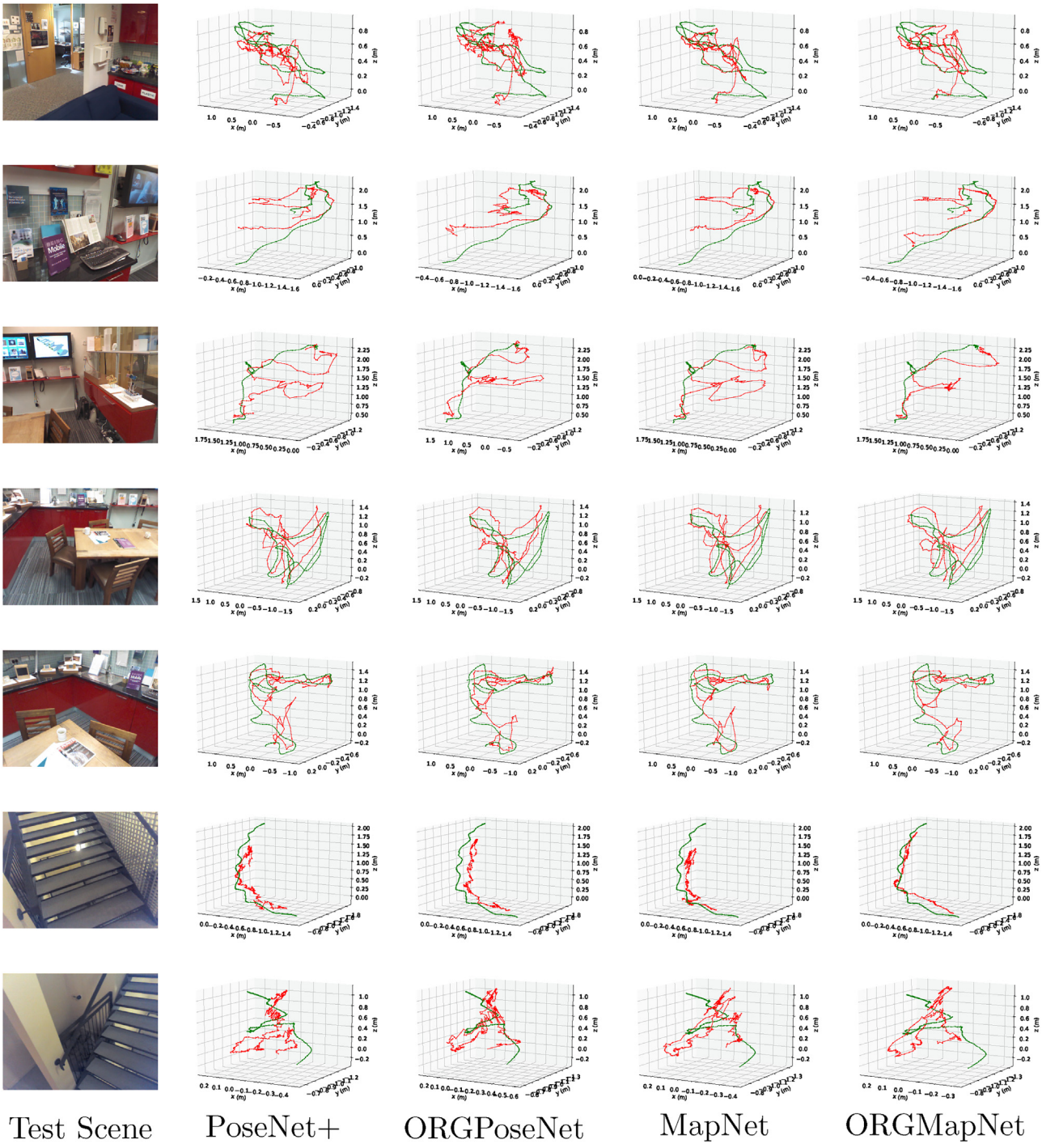

Fig. B.10. Raw images and the recovered trajectory of PoseNet+, MapNet, and our method on the 7-Scenes dataset. The green and red lines are the groundtruth and predicted trajectories, respectively. The testing sequences (from top to bottom) are pumpkin-07, redkitchen-03, redkitchen-04, redkitchen-06, redkitchen-14, stairs-01, and stairs-04.

Table B.6
Statistics of the baselines and our method on the 7-Scenes dataset. The best results are highlighted.

Scene	PoseNet+ [4]	ORGPoseNet (Ours)	MapNet [4]	ORGMapNet (Ours)
Avg Median (Scene)	0.22 m, 8.07°	0.21 m, 7.90°	0.21 m, 7.77°	**0.20 m**, **7.30°**
Avg Median (Seq)	0.24 m, 7.40°	0.22 m, 7.11°	0.22 m, 6.88°	**0.21 m**, **6.51°**
Avg Median (Image)	0.21 m, 6.54°	0.21 m, 6.32°	**0.20 m**, **5.80°**	**0.20 m**, 5.87°
Avg Mean (Scene)	0.28 m, 10.43°	0.26 m, 10.19°	0.27 m, 10.08°	**0.24 m**, **8.85°**
Avg Mean (Seq)	0.30 m, 9.84°	0.28 m, 9.70°	0.28 m, 9.12°	**0.26 m**, **8.04°**
Avg Mean (Image)	0.30 m, 9.98°	0.28 m, 9.64°	0.28 m, 8.87°	**0.26 m**, **7.89°**

B.2. Experiments on the RIO10 dataset

Table B.7 shows more details of the evaluation on the RIO10 dataset, which reports the median pose error on each scene.

B.3. Experiments on the RobotCar dataset

Fig. B.11 shows the trajectories of PoseNet+, MapNet, and our method on the FULL scene of the Oxford RobotCar dataset. For better visualization, we plot the estimated poses of frames with the

Table B.7
Pose errors of the baselines and our method on the RIO10 dataset. We report the median prediction errors here. The best results are highlighted.

Scene	PoseNet+ [4]	ORGPoseNet (Ours)	MapNet [4]	ORGMapNet (Ours)
S01	1.06 m, 36.61°	0.92 m, 34.66°	0.90 m, 38.12°	**0.87 m**, **32.85**°
S02	**0.44 m**, 19.59°	0.52 m, 16.77°	0.49 m, 18.80°	0.57 m, **15.16**°
S03	0.77 m, 31.34°	0.65 m, 21.37°	0.61 m, 21.60°	**0.57 m**, **16.27**°
S04	0.84 m, 81.91°	0.82 m, 85.60°	0.81 m, 73.49°	**0.79 m**, **63.10**°
S05	0.47 m, 46.39°	0.47 m, 36.76°	**0.44 m**, 33.63°	0.50 m, **28.21**°
S06	0.65 m, 21.15°	0.65 m, 18.12°	0.53 m, 12.38°	**0.47 m**, **10.46**°
S07	0.70 m, 35.23°	0.74 m, 30.52°	**0.68 m**, 38.56°	0.70 m, **27.76**°
S08	1.42 m, 48.68°	1.22 m, 61.08°	1.16 m, 56.21°	**1.05 m**, **44.06**°
S09	1.48 m, 77.73°	1.52 m, 83.00°	1.36 m, 60.89°	**1.21 m**, **56.57**°
S10	1.02 m, 36.26°	0.89 m, 32.62°	1.02 m, 35.28°	**0.88 m**, **30.24**°
Average	0.89 m, 43.49°	0.84 m, 42.05°	0.80 m, 38.90°	**0.76 m**, **32.46**°

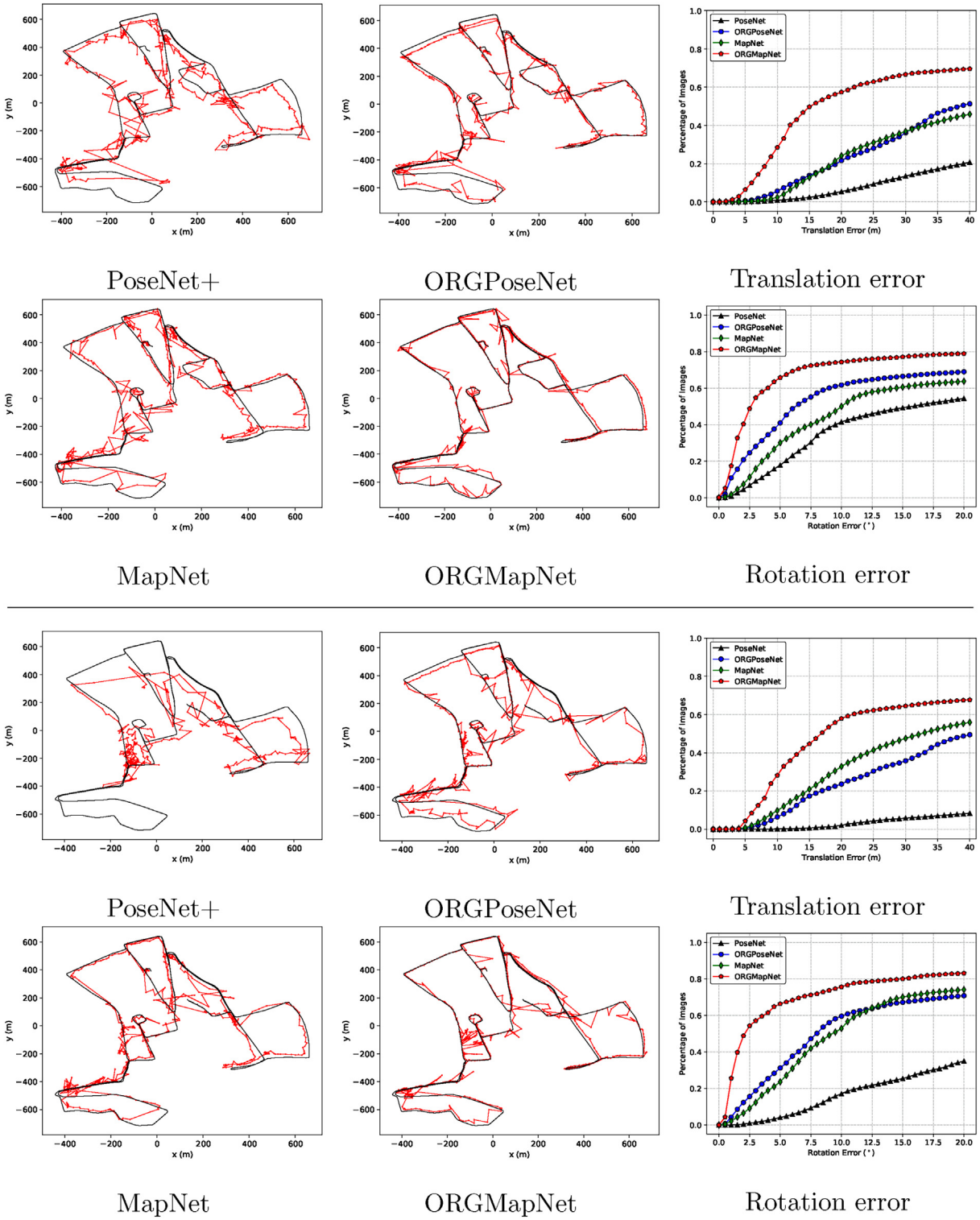

Fig. B.11. Trajectories of PoseNet+, MapNet, and our method on the FULL3 (top) and FULL5 (bottom) sequences of the Oxford RobotCar dataset. The red and black lines represent the predicted and the groundtruth camera poses, respectively. For better visualization, we select points with translation errors within 100 m. The cumulative distributions of translation and rotation errors are also shown on the right column.

translation errors within 100 m. Cumulative distributions of the translation and the rotation errors for the models are also illustrated. As can be seen, compared with the baselines, our method reduces the number of outliers and obtains more accurate pose estimation results under various complex weather conditions.

Appendix C. Supplementary data

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.neucom.2022.11.090.

References

[1] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, J. Sivic, Netvlad: Cnn architecture for weakly supervised place recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 5297–5307.

[2] S. Ardeshir, A.R. Zamir, A. Torroella, M. Shah, Gis-assisted object detection and geospatial localization, European Conference on Computer Vision, Springer. (2014) 602–617.

[3] V. Balntas, S. Li, V. Prisacariu, Relocnet: Continuous metric learning relocalisation using neural nets, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 751–767.

[4] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, J. Kautz, Mapnet: Geometry-aware learning of maps for camera localization, 2017.

[5] M. Bui, C. Baur, N. Navab, S. Ilic, S. Albarqouni, Adversarial networks for camera pose regression and refinement, in: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, 2019, pp. 0–0.

[6] M. Cai, C. Shen, I. Reid, hybrid probabilistic model for camera relocalization, 2019.

[7] R. Clark, S. Wang, A. Markham, N. Trigoni, H. Wen, Vidloc: A deep spatio-temporal model for 6-dof video-clip relocalization, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6856–6864.

[8] B. Dai, Y. Zhang, D. Lin, Detecting visual relationships with deep relational networks, in: Proceedings of the IEEE conference on computer vision and Pattern recognition, 2017, pp. 3076–3086.

[9] H. Dhamo, A. Farshad, I. Laina, N. Navab, G.D. Hager, F. Tombari, C. Rupprecht, Semantic image manipulation using scene graphs, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 5213–5222.

[10] M. Ding, Z. Wang, J. Sun, J. Shi, P. Luo, Camnet: Coarse-to-fine retrieval for camera re-localization, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 2871–2880.

[11] V. Gaudillière, G. Simon, M.O. Berger, Camera pose estimation with semantic 3d model, in: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2019, pp. 4569–4576.

[12] V. Gaudillière, G. Simon, M.O. Berger, Perspective-2-ellipsoid: Bridging the gap between object detections and 6-dof camera pose, IEEE Robotics and Automation Letters 5 (2020) 5189–5196.

[13] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[14] A. Kendall, R. Cipolla, Modelling uncertainty in deep learning for camera relocalization, in: 2016 IEEE international conference on Robotics and Automation (ICRA), IEEE, 2016, pp. 4762–4769.

[15] A. Kendall, R. Cipolla, Geometric loss functions for camera pose regression with deep learning, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 5974–5983.

[16] A. Kendall, M. Grimes, R. Cipolla, Posenet: A convolutional network for real-time 6-dof camera relocalization, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 2938–2946.

[17] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014. arXiv preprint arXiv:1412.6980.

[18] Z. Laskar, I. Melekhov, S. Kalia, 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, pp. 929–938.

[19] G. Li, M. Muller, A. Thabet, B. Ghanem, Deepgcns: Can gcns go as deep as cnns?, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9267–9276.

[20] J. Li, D. Meger, G. Dudek, Semantic mapping for view-invariant relocalization, in: 2019 International Conference on Robotics and Automation (ICRA), IEEE, 2019, pp. 7108–7115.

[21] X. Li, S. Wang, Y. Zhao, J. Verbeek, J. Kannala, Hierarchical scene coordinate classification and regression for visual localization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11983–11992.

[22] T.Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: Common objects in context, European conference on computer vision, Springer. (2014) 740–755.

[23] Y. Liu, Y. Petillot, D. Lane, S. Wang, Global localization with object-level semantics and topology, in: 2019 International Conference on Robotics and Automation (ICRA), IEEE, 2019, pp. 4909–4915.

[24] W. Maddern, G. Pascoe, C. Linegar, P. Newman, 1 year, 1000 km: The oxford robotcar dataset, The International Journal of Robotics Research 36 (2017) 3–15.

[25] I. Melekhov, J. Ylioinas, J. Kannala, E. Rahtu, Image-based localization using hourglass networks, in: Proceedings of the IEEE international conference on computer vision workshops, 2017, pp. 879–886.

[26] T. Naseer, W. Burgard, Deep regression for monocular camera-based 6-dof global localization in outdoor environments, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2017, pp. 1525–1530.

[27] L. Nicholson, M. Milford, N. Sünderhauf, Quadricslam: Dual quadrics from object detections as landmarks in object-oriented slam, IEEE Robot. Autom. Lett. 4 (2018) 1–8.

[28] F. Ott, T. Feigl, C. Loffler, C. Mutschler, Vipr: visual-odometry-aided pose regression for 6dof camera localization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 42–43.

[29] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, Advances in neural information processing systems 32 (2019) 8026–8037.

[30] M. Patel, B. Emery, Y.Y. Chen, Contextualnet: Exploiting contextual information using lstms to improve image-based localization, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2018, pp. 5890–5896.

[31] C.R. Qi, H. Su, K. Mo, L.J. Guibas, Pointnet: Deep learning on point sets for 3d classification and segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 652–660.

[32] N. Radwan, A. Valada, W. Burgard, Vlocnet++: Deep multitask learning for semantic visual localization and odometry, IEEE Robotics and Automation Letters 3 (2018) 4407–4414.

[33] R.F. Salas-Moreno, R.A. Newcombe, H. Strasdat, P.H. Kelly, A.J. Davison, Slam++: Simultaneous localisation and mapping at the level of objects, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2013, pp. 1352–1359.

[34] P.E. Sarlin, C. Cadena, R. Siegwart, M. Dymczyk, From coarse to fine: Robust hierarchical localization at large scale, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 12716–12725.

[35] P.E. Sarlin, A. Unagar, M. Larsson, H. Germain, C. Toft, V. Larsson, M. Pollefeys, V. Lepetit, L. Hammarstrand, F. Kahl, et al., Back to the feature: Learning robust camera localization from pixels to pose, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 3247–3257.

[36] T. Sattler, M. Havlena, K. Schindler, M. Pollefeys, Large-scale location recognition and the geometric burstiness problem, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 1582–1590.

[37] T. Sattler, B. Leibe, L. Kobbelt, Efficient & effective prioritized matching for large-scale image-based localization, IEEE Trans. Pattern Anal. Mach. Intell. 39 (2016) 1744–1756.

[38] T. Sattler, Q. Zhou, M. Pollefeys, L. Leal-Taixe, Understanding the limitations of cnn-based absolute camera pose regression, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 3302–3312.

[39] S. Seifi, T. Tuytelaars, How to improve cnn-based 6-dof camera pose estimation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, 2019, pp. 0–0.

[40] Y. Shavit, R. Ferens, Do we really need scene-specific pose encoders?, in: 2020 25th International Conference on Pattern Recognition (ICPR), IEEE, 2021, pp. 3186–3192.

[41] Y. Shavit, R. Ferens, Y. Keller, Learning multi-scene absolute pose regression with transformers, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 2733–2742.

[42] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, 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, 2013, pp. 2930–2937.

[43] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014. arXiv preprint arXiv:1409.1556.

[44] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.

[45] H. Taira, M. Okutomi, T. Sattler, M. Cimpoi, M. Pollefeys, J. Sivic, T. Pajdla, A. Torii, Inloc: Indoor visual localization with dense matching and view synthesis, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7199–7209.

[46] Z. Tian, C. Shen, H. Chen, T. He, Fcos: Fully convolutional one-stage object detection, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 9627–9636.

[47] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, T. Pajdla, 24/7 place recognition by view synthesis, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1808–1817.

[48] A. Valada, N. Radwan, W. Burgard, Deep auxiliary learning for visual localization and odometry, in: 2018 IEEE international conference on robotics and automation (ICRA), IEEE, 2018, pp. 6939–6946.

[49] F. Walch, C. Hazirbas, L. Leal-Taixe, T. Sattler, S. Hilsenbeck, D. Cremers, Image-based localization using lstms for structured feature correlation, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 627–637.

[50] J. Wald, T. Sattler, S. Golodetz, T. Cavallari, F. Tombari, Beyond controlled environments: 3d camera re-localization in changing indoor scenes, European Conference on Computer Vision, Springer (2020) 467–487.

[51] B. Wang, C. Chen, C.X. Lu, P. Zhao, N. Trigoni, A. Markham, Atloc: Attention guided camera localization, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2020, pp. 10393–10401.

[52] X. Wang, X. Wang, C. Wang, X. Bai, J. Wu, E.R. Hancock, Discriminative features matter: Multi-layer bilinear pooling for camera localization, in: British Machine Vision Conference, York, 2019.

[53] Y. Wang, W. Zhang, Q. Liu, Z. Zhang, X. Gao, X. Sun, Improving intra-and inter-modality visual relation for image captioning, in: Proceedings of the 28th ACM International Conference on Multimedia, 2020, pp. 4190–4198.
[54] P. Weinzaepfel, G. Csurka, Y. Cabon, M. Humenberger, Visual localization by learning objects-of-interest dense match regression, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5634–5643.
[55] J. Wu, L. Ma, X. Hu, Delving deeper into convolutional neural networks for camera relocalization, in: 2017 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2017, pp. 5644–5651.
[56] F. Xue, X. Wang, Z. Yan, Q. Wang, J. Wang, H. Zha, Local supports global: Deep camera relocalization with sequence enhancement, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 2841–2850.
[57] F. Xue, X. Wu, S. Cai, J. Wang, Learning multi-view camera relocalization with graph neural networks, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2020, pp. 11372–11381.
[58] Y. Zhu, R. Gao, S. Huang, S.C. Zhu, Y.N. Wu, Learning neural representation of camera pose with matrix representation of pose shift via view synthesis, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 9959–9968.

Chengyu Qiao is a Ph.D. candidate in the College of Information Science and Electronic Engineering at Zhejiang University, Hangzhou, China. He received his B.E. degree in Communication Engineering in 2018 from College of Information Science and Electronic Engineering, Zhejiang University. His research interests include autonomous driving technology, computer vision and deep learning.

Zhiyu Xiang received his B.S. and M.S degrees in mechanical engineering and automation and Ph.D. degree in information and telecommunication engineering from Zhejiang University, Hangzhou, China, in 1996, 1999, and 2002, respectively. He is currently a Professor with the College of Information Science and Electronic Engineering, Zhejiang University. His research interests include 2D and 3D computer vision, vision-based localization and mapping, and intelligent transportation systems.

Xinglu Wang is a Ph.D. student in the Computing Science of Simon Fraser University, British Columbia, Canada. He received his B.E. degree of Information Engineering in 2018 and Master's degree of Information and Communication Engineering in 2021 from the College of Information Science and Electrical Engineering, Zhejiang University. His research interests include data science and machine learning.

Shuya Chen received her B.E. degree in information science and electronic engineering from Zhejiang University, Hangzhou, Zhejiang, China, in 2017. She is currently pursuing the Ph.D. degree with the College of Information Science and Electronic Engineering. Her research interests include 2D and 3D computer vision and deep learning.

Yuangang Fan is a Master candidate in the College of Information Science and Electronic Engineering at Zhejiang University, Hangzhou, China. He received his B.E. degree in Communication Engineering in 2020 from College of Information Science and Electronic Engineering, Zhejiang University. His research interests include autonomous driving technology, computer vision and deep learning.

Xijun Zhao received the B.S degrees in vehicular engineering from Beijing Institute of Technology, Beijing, China, in 2007 and the Ph.D. degree in mechanical engineering from Beijing Institute of Technology, Beijing, China, in 2011. He is now working with China North Artificial Intelligence & Innovation Research Institute, China North Vehicle Research Institute. His research interests include perception, localization, motion planning and control of Unmanned Ground Vehicles.