Computer Vision and Image Understanding 259 (2025) 104440

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

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

A method for absolute pose regression based on cascaded attention modules

Xiaogang Song [a,b,*], Junjie Tang [a], Kaixuan Yang [a], Weixuan Guo [c], Xiaofeng Lu [a,b], Xinhong Hei [a,b]

[a] Xi'an University of Technology, School of Computer Science and Engineering, Xi'an, 710048, China
[b] Engineering Research Center of Human-machine integration intelligent robot, Universities of Shaanxi Province, Xi'an, 710048, China
[c] Xi'an Microelectronics Technology Institute, China

ARTICLE INFO

Communicated by Juergen Gall

Keywords:
Absolute pose regression
Attention mechanism
Feature fusion

ABSTRACT

The absolute camera pose regression estimates the position and orientation of the camera solely based on captured RGB images. However, current single-image techniques often lack robustness, resulting in significant outliers. To address the issues of pose regressors in repetitive textures and dynamic blur scenarios, this paper proposes an absolute pose regression method based on cascaded attention modules. This network integrates global and local information through cascaded attention modules and then employs a dual-stream attention module to reduce the impact of dynamic objects and lighting changes on localization performance by constructing dual-channel dependencies. Specifically, the cascaded attention modules guide the model to focus on the relationships between global and local features and establish long-range channel dependencies, enabling the network to learn richer multi-scale feature representations. Additionally, a dual-stream attention module is introduced to further enhance feature representation by closely associating spatial and channel dimensions. This method is evaluated and analyzed on various indoor and outdoor datasets, with our method reducing the median position error and orientation error to 0.19 m/7.44° on 7-Scenes and 7.09 m/1.45° on RobotCar, demonstrating that the proposed method can significantly improve localization performance. Ablation studies on multiple categories further verify the effectiveness of the proposed modules.

1. Introduction

Camera relocalization determines the position and orientation of a camera in three-dimensional space by analyzing image data, providing crucial technical support for the autonomous navigation of unmanned devices. By computing feature points, structures, or depth information in images and the correlations between adjacent frames, it finds extensive applications in augmented reality, virtual reality, mapping, and autonomous driving.

Traditional structure-based methods (Lowe, 2004; Mur-Artal et al., 2015) primarily rely on principles of geometry and triangulation. These methods match and measure feature points, lines, or planes in images and use local descriptors to establish high-quality correspondences (Mera-Trujillo et al., 2023; Revaud et al., 2019) between 2D pixel locations and 3D points for a given query image. The camera's position and orientation are then recovered using the PnP algorithm. While these methods perform well in static, small-scale scenes, they encounter challenges in large-scale, complex indoor and outdoor environments. The complexity of scene structures and the increase in the number of feature points often lead to matching confusion and increased computational complexity.

With the rise of deep learning technologies, learning-based camera pose estimation methods have demonstrated exceptional performance in handling complex environments due to their powerful ability to capture intricate nonlinear patterns from large datasets and establish corresponding mapping relationships. Currently, hierarchical structure-based localization pipelines (Taira et al., 2018; Sarlin et al., 2019) have achieved superior localization accuracy. These methods extract feature representations of scene images and use local feature matching to identify the image nodes most similar to the query image. Subsequently, they obtain the 2D-3D mapping through these matches and compute the camera pose using the PnP (Kneip et al., 2011) and RANSAC (Fischler and Bolles, 1981) algorithms. However, the multi-layered design of these methods results in high computational complexity, requiring more computational resources and longer processing times, which impacts their real-time performance to some extent.

The absolute pose regression method predicts the camera pose of a query image through a single forward pass of a trained network model, offering an efficient alternative to traditional image retrieval-based localization. It simultaneously learns to predict the camera's spatial position and orientation using a single feature vector from

* Corresponding author at: Xi'an University of Technology, School of Computer Science and Engineering, Xi'an, 710048, China.
E-mail addresses: songxg@xaut.edu.cn (X. Song), 2211221119@stu.xaut.edu.cn (J. Tang), 2241221135@stu.xaut.edu.cn (K. Yang), wxguo0818@hotmail.com (W. Guo), luxiaofeng@xaut.edu.cn (X. Lu), heixinhong@xaut.edu.cn (X. Hei).

https://doi.org/10.1016/j.cviu.2025.104440
Received 30 August 2024; Received in revised form 12 June 2025; Accepted 30 June 2025
Available online 5 July 2025

a convolutional network. PoseNet (Kendall et al., 2015) pioneered this approach, using a CNN for end-to-end training on a single RGB image, eliminating the need for keypoint extraction and matching, thus reducing computational burden and enhancing robustness. However, it still lags in localization accuracy compared to traditional structure-based methods, limiting its practical performance. Furthermore, recent theoretical and empirical studies (Sattler et al., 2019) have highlighted the fundamental limitations of CNN-based absolute pose regression, demonstrating that these models often fail to generalize beyond training scenes due to their limited capacity to model 3D geometry. To better handle uncertainty and improve the robustness of PoseNet in challenging scenarios such as those with obstacles, Kendall and Cipolla (2016) introduced Bayesian CNN and optimized the setting of weight hyperparameters to enhance the localization accuracy of the network model. The subsequent PoseNet2 (Kendall and Cipolla, 2017) optimized the loss function, further improving network performance. Later improvements based on PoseNet (Melekhov et al., 2017; Wu et al., 2017) focused on the model encoder and localizer, employing different structures and strategies. Esfahani et al. (2019) to enhance the accuracy and robustness of camera pose estimation. Shavit et al. (2024) presents a Transformer Encoder-based camera pose regression method that is faster, more efficient, and achieves state-of-the-art accuracy. Compared to hierarchical structure-based methods, the simpler structure and smaller size of APR methods offer superior portability. Typically, they can process a query image within 10 ms and do not rely on depth maps or SFM reconstruction, requiring only RGB images as input. Although these methods achieve end-to-end estimation from images to camera poses, they still face challenges in scenarios with sparse or repetitive textures, viewpoint changes, occlusion, and lighting variations. These challenges lead to deficiencies in the robustness and generalization capability of the networks.

In this work, we propose a camera absolute pose regression framework designed to learn rich multi-scale feature representations to address these challenges. The network model introduces attention mechanisms in the feature extractor part to guide the model to focus on the relationships between global and local features and establish long-range channel dependencies, enabling the network to learn richer multi-scale feature representations. The design of dual-stream attention allows the network to fully leverage the close association between spatial and channel semantics, further enhancing feature representation and obtaining high-precision and robust camera localization results. We experimentally evaluate the proposed method on indoor and outdoor public datasets, demonstrating significant improvements in localization performance. Additionally, we conduct ablation experiments across multiple categories to validate the effectiveness of the proposed modules. Our main contributions are as follows:

- We propose a novel absolute pose regression network called CA-Net (Cascade Attention-guided Network). We validated our method on various public datasets, and experimental results demonstrate that our approach achieves state-of-the-art performance.
- We have designed a cascade attention bottleneck block that integrates convolution and attention mechanisms. By associating global and local information and fusing multi-scale features, it effectively improves the performance and generalization capability of the network.
- We introduced a dual-stream attention module to enhance the model's focus on key features and suppress interference and noise, thereby achieving better localization performance.

2. Related work

2.1. Camera relocalization

Camera pose estimation can be categorized into several distinct types based on the methods of image information processing and the implementation techniques used.

Structure-based Methods: Correspondences between 2D pixel locations and 3D scene coordinates by matching local feature descriptors (Lowe, 2004; Murillo et al., 2007). The camera pose is then solved using the RANSAC-based PnP algorithm. To meet real-time requirements, fast binary descriptor methods (Calonder et al., 2010; Mur-Artal et al., 2015) emerged, representing feature point image information with binary codes, enabling efficient matching. Subsequent methods further improved localization accuracy by optimizing the feature matching process (Revaud et al., 2019; DeTone et al., 2018).

Retrieval-based Methods: By learning global descriptors, retrieval-based methods (Arandjelovic et al., 2016) search for the most similar images in a pre-constructed scene database and utilize their position and orientation information to estimate the camera's position and orientation. To improve model performance, SuperGlue (Sarlin et al., 2020) utilizes an attention model to dynamically calculate the matching degree of keypoint pairs. The LoFTR (Sun et al., 2021) algorithm introduces the Transformer architecture to enhance local feature matching performance. The Patch2Pix (Zhou et al., 2021) algorithm improves pixel-level matching accuracy. MatchFormer (Wang et al., 2022) combines self-attention and cross-attention in a hierarchical architecture of multi-scale features, enhancing matching robustness. These methods demonstrate state-of-the-art performance in large-scale scenarios. Moreover, this approach can be used in hierarchical structure pose estimation, serving as an initialization phase when combined with structure-based methods (Taira et al., 2018; Sarlin et al., 2019) or relative pose regression (Ding et al., 2019).

Scene Coordinate Regression: SCoRe methods map 2D pixel coordinates in an image to 3D scene coordinates. Once the corresponding 2D-3D relationships are found, the camera pose is computed using a RANSAC-based pose optimization method. DSAC (Brachmann et al., 2017) introduced a differentiable RANSAC method to regress camera scene coordinates from a scene coordinate random forest. Subsequent improvements, DSAC++ (Brachmann and Rother, 2018) and DSAC* (Brachmann and Rother, 2021), addressed issues such as overfitting in the scoring module to further enhance network accuracy. More recent methods (Xie et al., 2022; Dai et al., 2023) have improved camera localization accuracy by using attention modules to fuse contextual information or by considering image patch similarity issues.

Relative Pose Regression: RPR methods (Ding et al., 2019; Balntas et al., 2018; Charco et al., 2021) calculate the relative motion information of the camera between different times or positions through steps such as feature extraction, matching, motion estimation, and optimization. These methods focus on inferring the camera's motion relative to previous frames or positions and typically rely on sequentially acquired images over time (Valada et al., 2018; Radwan et al., 2018). They are often combined with retrieval-based methods or other approaches.

Absolute Pose Regression: APR methods aim to directly predict the position and orientation of a camera in a global coordinate system using an end-to-end deep neural network model. PoseNet (Kendall et al., 2015), a pioneering work in this area, employs a CNN to train on a single RGB image end-to-end, eliminating the need for keypoint extraction and matching required by traditional methods, thus reducing computational burden and enhancing robustness. Subsequent improvements (Kendall and Cipolla, 2017; Melekhov et al., 2017; Wu et al., 2017) have optimized the loss function and employed different structures and strategies to achieve higher robustness and accuracy. To address the cumbersome retraining process required when new data is introduced and the accuracy limitations caused by the small number of training images available in individual scenes, Blanton et al. (2020) proposed a cross-scene generalization method. This method involves joint training, enabling the network to perform scene recognition alongside camera pose regression. Building on this, Shavit and Ferens (2021) introduced a Transformer to effectively capture camera pose variations in diverse scenes. AtLoc (Wang et al., 2020a) utilizes an attention mechanism to make the network more focused on geometrically meaningful regions, encouraging the model to learn stable features. Direct-PN (Chen et al., 2021) combines an APR network with a new view synthesis-based direct matching module, achieving superior accuracy without increasing regression time.

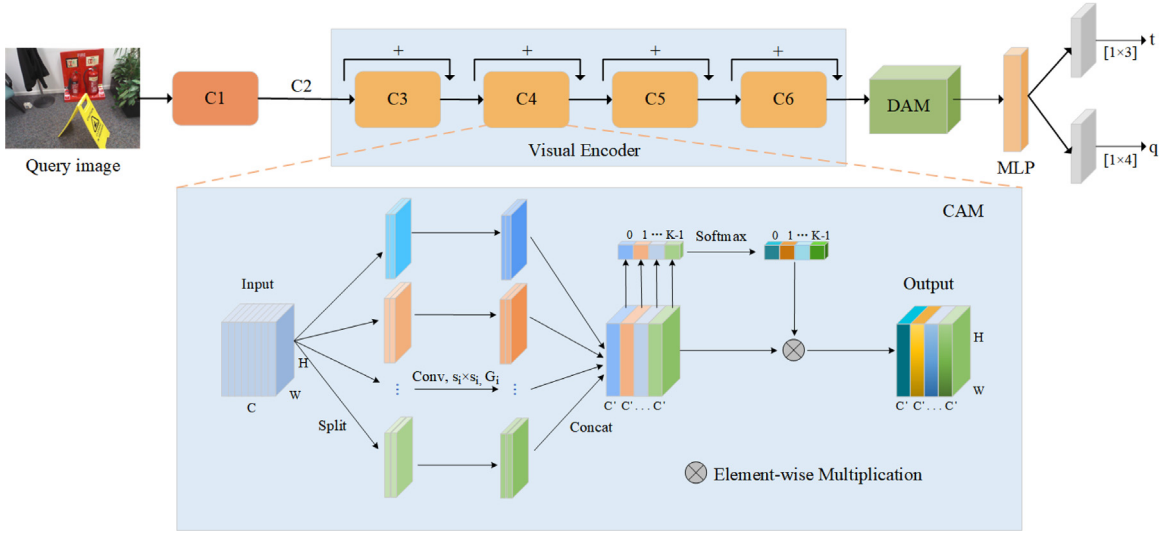


Fig. 1. Overview of the proposed CA-Net method. This method consists of three parts: a visual encoder, a dual-stream attention module, and a multi-layer perceptron. The network takes a single RGB image as input and outputs the position and orientation of the camera.

2.2. Attention mechanism

Attention mechanisms allow neural networks to selectively focus on key parts of input data, guiding the model to suppress redundant features and attend to the parts relevant for effective network processing of sequential data or images.

One of the earliest proposed attention models is the SE module (Hu et al., 2018), which adjusts the feature responses of each channel using global information, enabling the network to focus more on important features while suppressing irrelevant ones, thereby enhancing model performance. GSoP-Net (Gao et al., 2019) proposes modeling higher-order statistics using global second-order pooling, while also collecting global information to improve the squeeze module. EncNet (Zhang et al., 2018) utilizes a context encoding module that integrates semantic encoding loss to model the relationship between environmental background information and confidence in target classification, thereby achieving precise extraction and prediction of semantic information from images. ECA-Net (Wang et al., 2020b) replaces traditional dimensionality reduction operations with one-dimensional convolution operations to facilitate interaction between channels, reducing computational costs while maintaining performance. CBAM (Woo et al., 2018) enhances the focus on important features in neural networks by stacking channel and spatial attention modules, while SGE (Li et al.) enhances features through spatial grouping and normalization. In addition to the attention mechanisms discussed above, recent works have explored specialized designs to better capture multi-dimensional relationships in visual data. For example, Li et al. (2025) proposed an Enhanced Multiview Attention Network (EMAN) that uses static multi-view divisions with fixed weights to fuse visual features, demonstrating strong performance in few-shot surface defect detection tasks. Wang et al. (2025) introduced a dual-branch attention network that explicitly separates content and style features using style contrastive learning, thereby improving image enhancement under complex underwater conditions. Our work integrates global and local information via attention modules, facilitating the network in acquiring more comprehensive multi-scale feature representations. Additionally, it mitigates the influence of variables like dynamic objects and lighting alterations on localization accuracy by establishing dual-channel dependencies.

3. Method

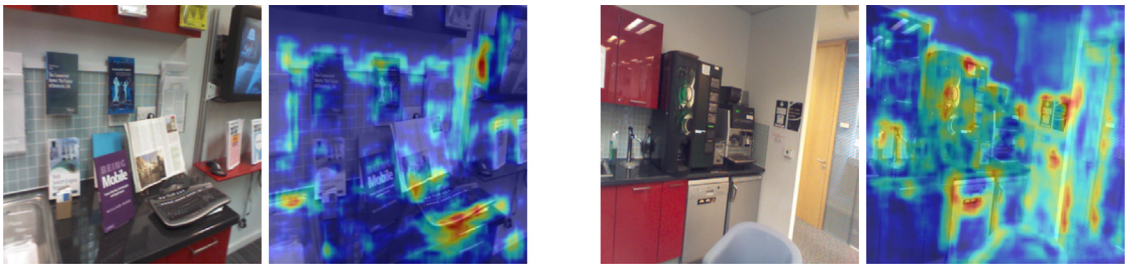
Our proposed absolute pose regression network mainly consists of three parts, as illustrated in Fig. 1. The first part is the visual encoder, which processes the input single RGB image to extract features required for the pose regression task, such as edges, textures, and shapes. In the subsequent steps, the attention module performs segmentation to generate channel-based multi-scale feature maps. Then, attention is extracted from each scale of the feature maps to form channel attention vectors. These vectors are then normalized using the Softmax function to generate corresponding weights and feature maps. Subsequently, the pose regressor maps the feature representation adjusted through feature extraction and attention mechanisms to the camera pose space. This process involves learning the nonlinear mapping relationship between features and camera poses, thereby achieving direct regression of the camera position and orientation.

Table 1
Comparison of structural parameters between ResNet50 and our Visual Encoder.

Stage	Output	ResNet50	Visual Encoder
C1	512 × 512	7 × 7, 64, stride:2	7 × 7, 64, stride:2
C2	256 × 256	3 × 3, max pool, stride:2	3 × 3, max pool, stride:2
C3	256 × 256	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ \text{CAM}, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$
C4	256 × 256	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ \text{CAM}, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$
C5	256 × 256	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ \text{CAM}, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$
C6	256 × 256	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ \text{CAM}, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$

3.1. Visual encoder

The visual encoder learns abstract representations of images through multiple layers of convolutional and pooling operations to capture feature information in the images. Classic visual encoders such as GoogLeNet, VGG, and ResNet have been widely used in the field of camera pose estimation due to their simple yet effective structures. Among them, ResNet addresses common issues in training deep neural networks, such as vanishing gradients and exploding gradients, by introducing residual connections, resulting in superior localization



(a) Kitchen (b) Pumpkin

Fig. 2. Examples of attention visualization. For each scenario, the left image is the original input and the right image is the attention heatmap.

results. Therefore, to balance the relationship between localization accuracy and network parameter count, ResNet50 is adopted as the base network for the visual encoder in the CA-Net model of this chapter, embedding the designed cascade attention modules. The specific structural parameters are shown in Table 1.

In order to effectively enhance the performance of the deep convolutional neural network, we made modifications to the basic backbone network. Specifically, motivated by the limitations of traditional attention modules in constructing long-range channel dependencies with complex designs, we replaced the 3×3 spatial convolution in the residual bottleneck block with the Cascade Attention Module (CAM). This replacement aims to strengthen the weighting of feature importance within the model, thereby enhancing the network's representational capacity and overall performance. In the visual encoder's final stage, we removed the Softmax layer typically used for classification tasks and modified the initial 1000-dimensional fully connected layer. Instead, we employed a 2048-dimensional fully connected layer to mitigate the impact of feature compression and capture more diverse and richer image features. For a given input image $I \in \mathbb{R}^{(H \times W \times C)}$, the features extracted by the visual encoder are represented as $X \in \mathbb{R}^{(8 \times 8 \times 2048)}$. The proposed CAM-based architecture demonstrates stronger generalization capabilities in complex environments, particularly under dynamic object changes and lighting variations, by learning fine-grained multiscale features through segmentation-fusion operations.

The implementation process of CAM specifically involves several key steps. Firstly, we utilize specific segmentation and concatenation techniques to construct channel-based multiscale feature maps. Subsequently, crucial attention information is extracted from these multiscale feature maps, forming channel attention vectors accordingly. Following this, the channel attention vectors are finely calibrated and adjusted using the Softmax function. Finally, we adopt an element-wise multiplication approach to fuse the calibrated weights with the corresponding feature maps, generating more refined feature maps, which serve as the final output of the module.

Given an input feature map X with a shape of (H, W, C), it is divided into K parts along the channel dimension, denoted as $[X_0, X_1, \ldots, X_{K-1}]$. For each individual part, the common number of channels is:

$$C' = \frac{C}{K} \tag{1}$$

where C is divisible by K. For the ith feature map, we denote it as $X_i \in R^{C' \times H \times W}$, where $i = 0, 1, \ldots, K-1$. This partitioning strategy we employ aims at concurrently processing input tensors of different scales, thereby generating feature maps with specific kernel types. Within each partition, they independently learn multi-scale spatial information and achieve cross-channel information exchange in a local manner.

In our implementation, the input feature map is evenly divided into K parts along the channel dimension to construct multi-scale feature maps. This even splitting strategy ensures that each group has sufficient representation capacity while simplifying the design of the group convolution structure. Although K is empirically set rather than analytically derived, it is selected based on preliminary experiments to ensure effective multi-scale feature extraction while maintaining computational efficiency. In practice, this choice provides a good balance between model complexity and performance. The proposed CAM module demonstrates significant gains under this design.

To address the issue of escalating computational cost due to the sharp increase in parameter count with the growth of kernel size, we employed a group convolution approach (Zhang et al., 2021), which enables parallel execution of multiple sets of convolution operations. Here, the association between group size G and kernel size S is defined as $G = 2^{\frac{S-1}{2}}$. Furthermore, for feature maps of different scales $F_i \in R^{C' \times H \times W}$, where C', H', and W' represent the number of channels, height, and width, respectively. Its expression is as follows:

$$F_i = Conv_{s_i \times s_i}(G_i, X_i) \tag{2}$$

where $s_i \times s_i$ denotes the size of the ith kernel.

The processing result of multi-scale feature maps, denoted as $F_{out} \in R^{C \times H \times W}$, can be obtained through $Concat(\cdot)$ operation, represented as:

$$F_{out} = \mathrm{Concat}([F_0, F_1, \ldots, F_{K-1}]) \tag{3}$$

where $Concat(\cdot)$ denotes the concatenation operation.

Through detailed analysis of multi-scale feature maps, weight information related to channel attention is extracted. Based on this information, attention weight vectors specific to each scale, denoted as $V_i \in R^{C' \times 1 \times 1}$, can be generated. These vectors accurately reflect the channel attention distribution at different scales. This process not only integrates multi-scale information but also enhances the optimization of feature maps. Subsequently, while preserving the integrity of the original channel attention vectors, attention information interaction is achieved, and effective fusion of cross-dimensional vectors is completed. The final attention vector is integrated through concatenation, as follows:

$$V = V_0 \oplus V_1 \oplus \cdots \oplus V_{K-1} \tag{4}$$

In cross-channel applications, soft attention is utilized to adaptively select and focus on different spatial scales. The formula can be expressed as:

$$Att_i = \mathrm{softmax}(V_i) = \frac{exp(V_i)}{\sum_{i=0}^{K-1} exp(V_i)} \tag{5}$$

where the weights Att_i are obtained by applying the $softmax(\cdot)$ function. These weights not only account for the feature information at spatial locations but also incorporate the attention distribution characteristics within the channels. This design successfully achieves an effective combination of local channel attention and global channel attention, thereby significantly enhancing the model's performance in feature processing.

Subsequently, Att_i is multiplied with the feature map F_i at the corresponding scale, which can be represented as:

$$Y_i = F_i \odot Att_i \tag{6}$$

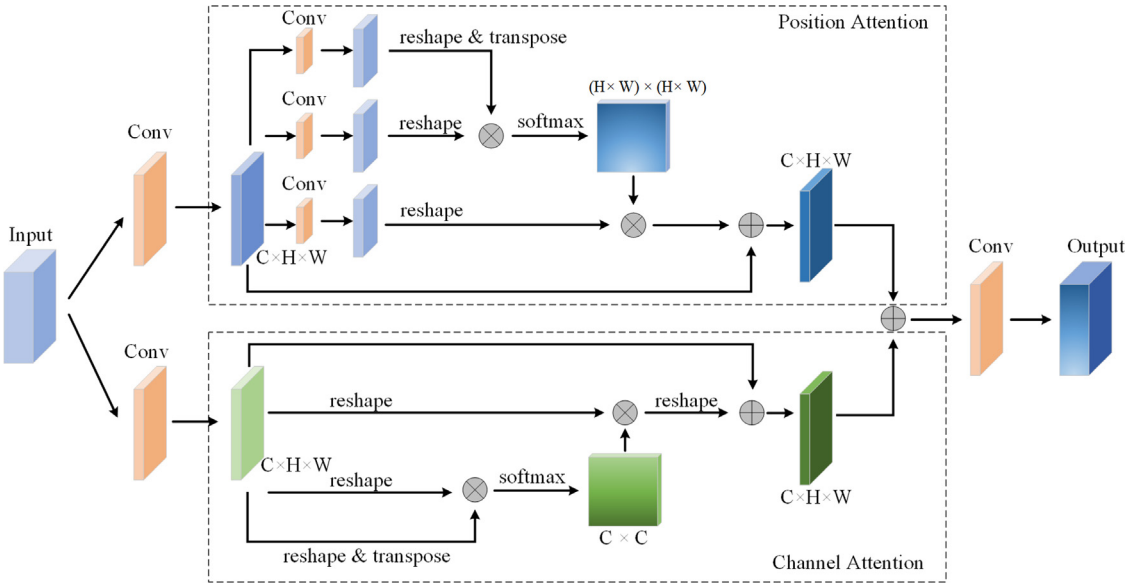


Fig. 3. Overview of the Dual-Stream Attention Module. This module consists of two sub-modules: positional attention and spatial attention.

where ⊙ denotes channel-wise multiplication, and Y_i is the resulting feature map.

Considering that the concatenation operator, as opposed to summation, can comprehensively maintain the integrity of feature representation without compromising the original feature map information, the final refined output Y_{out} can be expressed as:

$$Y_{out} = C\text{oncat}([Y_0, Y_1, \ldots, Y_{K-1}]) \tag{7}$$

The visual encoder part adopts a new block structure, CABo, by replacing the corresponding 3 × 3 convolution in the residual block with CAM. This allows the model to retain its original resistance to vanishing and exploding gradients while introducing finer-grained spatial information necessary for camera pose regression. By integrating multi-scale information and attention mechanisms into CABo, the network achieves strong multi-scale representation capabilities and can adaptively recalibrate cross-dimensional channel weights, thus obtaining robust feature representations more suitable for pose regression tasks. To better illustrate the effectiveness of the proposed attention modules, we visualize the attention maps from two representative scenes. As shown in Fig. 2, our model learns to focus on geometrically meaningful structures such as object edges, layout boundaries, and stable surfaces, which are crucial for accurate pose regression.

In this work, ResNet50 is selected as the visual encoder considering its proven effectiveness, architectural stability, and computational efficiency across a wide range of vision tasks. Although recent studies have demonstrated the capability of Transformer-based encoders in capturing long-range dependencies with global attention mechanisms, their higher computational cost and data demands may limit their practicality in certain real-time or resource-constrained settings. Nevertheless, incorporating lightweight vision Transformers could offer further improvements in representation learning, which merits investigation in future work.

3.2. Dual-stream attention module

In pose regression networks, features extracted by the visual encoder typically undergo global average pooling before being fed into the fully connected regressor. While the spatial invariance of global average pooling allows the model to correctly identify objects despite changes in their positions within the image, it can adversely affect the network's ability to handle noise and maintain spatial information. To address this issue, we introduce a Dual-Stream Attention Module (DAM). This module incorporates a spatial attention mechanism, which effectively captures spatial information within the image, aiding the network in better understanding the geometric structure and layout of the scene. Additionally, a channel attention mechanism is introduced to capture the relationships between different channels, enabling better expression of feature information through weighted processing. The outputs of these two modules are then fused to further enhance feature representation, allowing the model to better comprehend the semantic structure of the scene and providing more reliable environmental information for camera pose estimation. The structure of DAM is shown in Fig. 3.

For a given feature map $X \in R^{C\times H\times W}$, the Position Attention Module first performs convolution operations to obtain three derived feature maps f_1, f_2, and f_3, where $\{f_1, f_2, f_3\} \in R^{C\times H\times W}$. These are then reshaped into matrix form $R^{C\times N}$, where $N = H\times W$ represents the number of pixels. Subsequently, the matrices f_1 and f_2 are multiplied, followed by the application of a softmax layer to produce the desired spatial attention map $M \in R^{N\times N}$, expressed as:

$$m_{ji} = \frac{\exp(f_1^i \cdot f_2^j)}{\sum_{i=1}^{N} \exp(f_1^i \cdot f_2^j)}, \quad i, j \in \{1, 2, \ldots, N\} \tag{8}$$

Afterwards, f_3 is multiplied with the transpose of matrix M, and reshaped back to $R^{C\times H\times W}$. Then, the obtained matrix is multiplied by the scale parameter γ to adjust the weights, and element-wise addition is performed with the original feature X to obtain the final output Y_p:

$$Y_p = \gamma \sum_{i=1}^{N} (f_3^i \cdot m_{ji}) + X_j, \quad i, j \in \{1, 2, \ldots, N\} \tag{9}$$

where γ is initialized to 0 and is learned to allocate more weights (Zhang et al., 2019).

In the Channel Attention Module, each channel map is treated as a response signal for a specific category. These response signals influence each other at a semantic level, reflecting the correlations between different features. Here, the original feature map $X \in R^{C\times H\times W}$ is used as the computational basis to directly derive the channel attention map $Z \in R^{C\times C}$. To achieve this goal, first, reshape the feature map X into a matrix structure $R^{C\times N}$. Then, by performing matrix multiplication with its transpose, a new matrix is obtained. Finally, to obtain the channel attention map Z, apply a softmax layer to this new matrix, expressed as:

$$z_{ji} = \frac{\exp(X_i \cdot X_j)}{\sum_{i=1}^{C} \exp(X_i \cdot X_j)}, \quad i, j \in \{1, 2, \ldots, C\} \tag{10}$$

where z_{ji} is used to measure the influence of the ith position on the jth position.

Through matrix multiplication, the integrated features of the attention map Z and the original feature X are obtained, followed by size adjustment. By introducing a scaling parameter ε, it is multiplied with the aforementioned result, and finally, by element-wise addition, this result is combined with the original feature X to obtain the final output Y_c:

$$Y_c = \varepsilon \sum_{i=1}^{C} (z_{ji} \cdot X_i) + X_j \tag{11}$$

where ε is gradually learned from 0 to assign weights. After obtaining the outputs of the two modules, they are subjected to element-wise summation, followed by convolution to obtain the final prediction map Y_{res}, expressed as:

$$Y_{res} = Conv(Y_p \oplus Y_c) \tag{12}$$

The Dual-Stream Attention Module (DAM) enhances the quality and expressive power of feature representation through dual weighting of channel and spatial attention. This weighting mechanism enables the model to more effectively capture important features, thereby improving its understanding and generalization capabilities towards input data.

3.3. Loss function

We employ a multi-layer perceptron (MLP) to learn the non-linear mapping relationships of features. By adding two fully connected layers at the end of the network, serving as the pose regression module, we achieve regression for both position and orientation. The MLP maps features extracted by the visual encoder and DAM, enabling better capture and representation of semantic information and spatial relationships in the images. It maps the predicted features Y_{res} of the model to the position $t \in R^3$ and orientation $q \in R^4$, expressed as:

$$[t, q] = MLP(Y_{res}) \tag{13}$$

For a query image I and its corresponding ground truth pose $[\hat{t}, \hat{q}]$ and predicted pose $[t, q]$, the pose regression module is employed to reduce the error between them. For translation, the L1 distance is used. For rotation, we use a quaternion-based formulation that reflects the angular difference through the squared inner product of normalized quaternions. The final loss is expressed as:

$$\text{Loss}(I) = \|t - \hat{t}\|_1 \cdot e^{-\beta} + \beta + (1 - \langle q, \hat{q} \rangle^2) \cdot e^{-\gamma} + \gamma \tag{14}$$

where $\langle q, \hat{q} \rangle$ denotes the inner product between the predicted and ground truth unit quaternions, and β, γ are trainable parameters that balance the translation and rotation losses. Their initial values are set to $\beta_0 = 0$ and $\gamma_0 = -3.0$, and they are learned during training.

4. Experiments

To address the blurriness caused by dynamic objects and viewpoint changes, we have designed an absolute pose regression method based on cascaded attention modules. This method aims to enhance the camera relocalization performance. We conducted comprehensive evaluations on the Oxford RobotCar and 7-Scenes public datasets to validate its performance. Each dataset comprises multiple scenes.

4.1. Implementation details

We utilized the NVIDIA GeForce RTX 3090 GPU as the training platform and implemented the proposed algorithm using the PyTorch framework. To ensure efficient and robust training, we employed the Adam optimizer with a learning rate set to 10^{-4}. Various strategies were employed for image cropping. During testing, a random cropping strategy was adopted to capture more variations in the images. During training, a center cropping method was chosen to ensure that the model learns the main features of the images. As a result, input images of size 256×256 pixels were obtained. The specific parameter settings used by this network are as follows: batch size of 64, dropout probability of 0.5, and learning parameters to balance between the two losses initialized as $\beta_0 = 0.0$ and $\gamma_0 = -3.0$. Additionally, to adapt to outdoor lighting variations, data augmentation was employed, with a color jitter parameter set to 0.7. Color jitter involves random transformations of exposure, saturation, and hue.

4.2. Datasets

We evaluated our algorithm on two public datasets, including the indoor dataset 7-Scenes (Shotton et al., 2013) and the outdoor dataset Oxford RobotCar (Maddern et al., 2017).

The 7-Scenes dataset was captured by researchers at the University of Cambridge using the Microsoft Kinect series of RGB-D cameras. Each scene contains several sequences, with each sequence comprising 500 to 1000 frames of images at a resolution of 640 × 480 pixels. This dataset focuses on the challenges and characteristics faced by each scene, including but not limited to varying lighting conditions, occlusions obstructing the camera's view, and repetitive or sparse texture features in different scenes. The indoor dataset aims to provide a diverse, real-world visual environment for evaluating localization algorithms.

The Oxford RobotCar dataset consists of nearly 20 million images captured by researchers using a self-driving car, which traveled over 1000 kilometers and relied on six cameras mounted on the vehicle. This dataset contains various weather and lighting conditions, including heavy rain, nighttime, direct sunlight, and snowfall. Significant changes in road segments occurred due to road and building construction throughout the year of data collection. Additionally, handling potential dynamic objects such as pedestrians and vehicles presents challenges for camera pose estimation tasks. In this paper, we use a subset of the RobotCar dataset called FULL as the experimental dataset.

4.3. Experiments on the 7-Scenes dataset

On the 7-Scenes dataset, we use the median error of position and orientation as evaluation metrics. We compare our proposed CA-Net with other absolute pose regression methods such as PoseNet (Kendall et al., 2015), Direct-PN (Chen et al., 2021), AtLoc (Wang et al., 2020a), EFRNet-VL (Wang et al., 2024) etc., P2I-NET (Kang et al., 2024), as detailed in Table 2. The experimental results demonstrate that our proposed network model exhibits excellent performance in areas with sparse textures such as "fire" and "pumpkin", as well as scenes with significant repetitive texture characteristics like "stairs".

PoseNet (Kendall et al., 2015) served as a seminal work in this field, but exhibited significant errors in accuracy. Subsequent to this, PoseNetLearn (Kendall and Cipolla, 2017) proposed a method for automatically learning optimal balance weights by improving the loss function. Direct-PN (Chen et al., 2021) refined the pose regression network by utilizing photometric supervision signals through differentiable rendering, thereby improving the accuracy of pose regression. On the other hand, Zangeneh et al. (2023) proposed a probabilistic framework that first captures useful visual features through distributed visual localization, and then solves for pose distribution, thereby enhancing the localization method's ability to distinguish ambiguous scenes. In contrast, in the feature extraction part, we obtained multi-scale feature maps of channels by using segmentation operations, and then generated corresponding weights and feature maps through subsequent connection operations, prompting the model to focus on the relationship between global and local features and establish long-range channel dependencies. Subsequently, the design of the dual-stream attention enabled the network to fully exploit the close correlation between spatial and channel semantics, thereby further improving feature representation and obtaining high-precision and robust camera localization results.

Table 2
Performance comparison in terms of the median position error (meters) and orientation error (degrees) on 7-Scenes dataset. The best results are highlighted in bold.

Method	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs	Average
PoseNet (Kendall et al., 2015)	0.32/8.12	0.47/14.40	0.29/12.00	0.48/7.68	0.47/8.42	0.59/8.64	0.47/13.80	0.45/9.94
PoseNetLearn (Kendall and Cipolla, 2017)	0.14/4.50	0.27/11.80	0.18/12.10	0.20/5.77	0.25/4.82	0.24/5.52	0.37/10.60	0.24/7.87
GPoseNet (Cai et al., 2019)	0.20/7.11	0.38/12.30	0.21/13.80	0.28/8.83	0.37/6.94	0.35/8.15	0.37/12.50	0.31/9.95
Direct-PN (Chen et al., 2021)	0.10/**3.52**	0.27/8.66	0.17/13.10	**0.16**/5.96	0.19/**3.85**	0.22/5.13	0.32/10.61	0.20/**7.26**
ViPR (Ott et al., 2020)	0.22/7.89	0.38/12.74	0.21/16.41	0.35/9.59	0.37/8.45	0.40/9.32	0.31/12.65	0.32/11.01
Xu et al. (2020)	0.13/4.67	0.32/10.50	0.17/12.70	0.22/6.29	0.21/5.64	0.19/6.27	0.40/10.10	0.23/8.01
IRPNet (Shavit and Ferens, 2021)	0.13/5.64	0.25/9.67	0.15/13.10	0.24/6.33	0.22/5.78	0.30/7.29	0.34/11.60	0.23/8.49
AtLoc (Wang et al., 2020a)	0.10/4.07	0.25/11.40	0.16/11.80	0.17/5.34	0.21/4.37	0.23/5.42	0.26/10.50	0.20/7.56
Zangeneh et al. (2023)	0.17/6.90	0.30/14.1	0.17/14.5	0.24/9.30	0.30/8.33	0.26/10.2	0.47/15.5	0.27/11.26
TransBoNet (Song et al., 2024)	0.11/4.48	0.25/12.46	0.18/14.0	0.20/5.08	0.19/4.77	**0.17**/5.35	0.30/13.04	0.20/8.45
EFRNet-VL (Wang et al., 2024)	0.11/3.96	0.26/11.11	0.17/12.87	0.17/5.46	0.20/4.48	0.22/5.95	0.25/10.28	0.20/7.73
P2I-NET (Kang et al., 2024)	0.12/6.34	0.36/9.83	0.25/**10.83**	0.27/7.12	0.38/6.31	0.35/7.46	**0.23**/7.08	0.28/7.85
Shavit et al. (2024)	**0.08**/5.68	0.24/10.6	**0.13**/12.7	0.17/6.34	**0.17**/5.60	0.19/6.75	0.30/**7.02**	**0.19**/7.82
CA-Net(Ours)	0.09/5.35	**0.23/8.63**	0.19/13.56	0.21/**5.03**	**0.17**/4.91	**0.17/4.93**	0.24/9.68	**0.19**/7.44

Table 3
Performance comparison of mean and median positional errors (meters) and orientation errors (degrees) on the Oxford RobotCar dataset. The best results are highlighted in bold.

FULL	PoseNet+(Kendall and Cipolla, 2017)		MapNet (Brahmbhatt et al., 2018)		LsG (Xue et al., 2019)		CA-Net	
–	Mean	Median	Mean	Median	Mean	Median	Mean	Median
FULL1	125.6/27.1	107.6/22.5	41.4/12.5	17.94/6.68	31.65/**4.51**	–	**28.89**/7.86	**5.94/1.31**
FULL2	131.1/26.05	101.8/20.1	59.3/14.8	20.04/6.39	53.45/**8.60**	–	**49.68**/9.72	**8.23/1.58**
Average	128.4/26.58	104.7/21.3	50.4/13.6	18.99/6.54	42.55/**6.56**	–	**39.29**/8.79	**7.09/1.45**

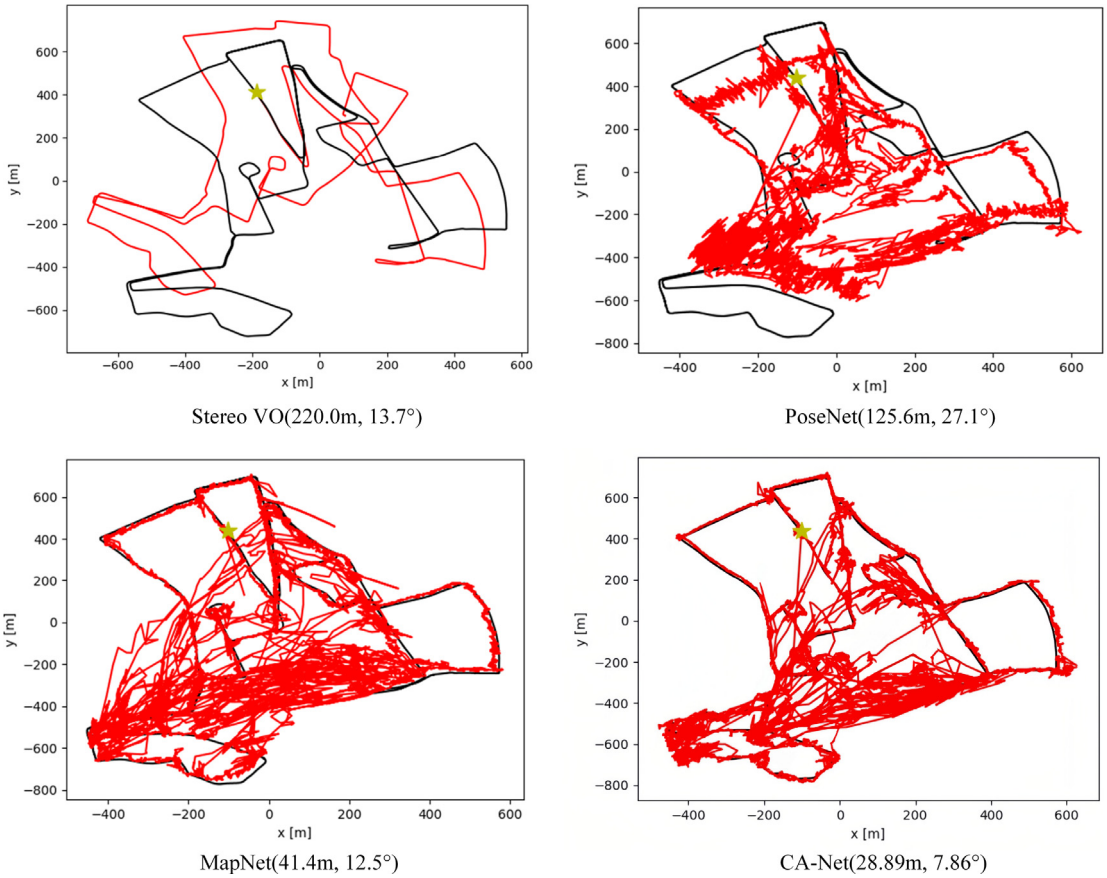


Fig. 4. Visualization of the trajectory for the FULL1 scene in the Oxford RobotCar dataset. The black line and the red line represent the ground truth and predicted trajectories, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.4. Experiments on the RobotCar dataset

On the Oxford RobotCar dataset, we employed median and mean errors as metrics and compared our model with PoseNet (Kendall et al., 2015; Kendall and Cipolla, 2016, 2017), MapNet (Brahmbhatt et al., 2018), and LsG (Xue et al., 2019). Experimental results are presented in Table 3. Due to variations in the time and weather conditions of the dataset's captures, the disparities between the training and testing scenes posed challenges for PoseNet, resulting in numerous unstable estimates. MapNet significantly reduced the occurrence of outliers by

Table 4
We set β and γ to different values, and compare their median and mean positional/orientation errors (in meters/degrees) on 7-Scenes and Oxford RobotCar datasets. The best results are highlighted in bold.

Scene	$\beta = 0.0, \gamma = -2.0$	$\beta = 0.0, \gamma = -3.0$	$\beta = 0.0, \gamma = -4.0$	$\beta = 1.0, \gamma = -3.0$
		7-Scenes		
Chess	0.10/ 5.37	**0.09/ 5.19**	0.11/ 5.54	0.10/ 5.27
Fire	0.24/ 10.37	**0.23/ 8.79**	0.25/ 9.56	**0.23**/ 9.24
Heads	0.19/ 13.60	0.19/ **13.56**	0.19/ 14.03	**0.18**/ 13.75
Office	0.22/ 6.53	**0.21/ 5.03**	**0.21**/ 6.94	0.22/ 7.08
Pumpkin	**0.17**/ 5.74	**0.17/ 4.91**	0.18/ 5.49	0.17/ 5.16
Kitchen	0.19/ 5.67	**0.17/ 4.93**	0.18/ 6.15	0.19/ 5.17
Stairs	**0.23**/ 9.93	0.24/ **9.68**	0.24/ 10.75	0.25/ 10.22
		Oxford RobotCar		
FULL1	39.53/ **7.61**	**28.89**/ 7.86	34.28/ 8.37	38.27/ 8.95
FULL2	57.29/ 13.15	**49.68/ 9.72**	56.93/ 11.42	54.67/ 12.13

introducing relative poses as constraints. Meanwhile, LsG further improved localization accuracy through content enhancement methods. In comparison, our approach demonstrated superior adaptability to scene variations, as depicted by the average localization error shown in Fig. 4 .

4.5. Parameter sensitivity analysis

During the training of our model, the learning parameters β and γ are utilized to adjust the relative weights of the position and orientation losses, respectively. To evaluate the impact of these hyperparameter settings on the pose regression model, we conducted experiments with various initial values and assessed their experimental results, as detailed in Table 4. The initialization settings were guided by the hyperparameter configurations of networks such as MapNet and AtLoc. We chose $\beta = 0.0$ and $\gamma = -3.0$ as the initial training hyperparameters, weakening the weight of position regression to enhance the model's performance in absolute pose regression tasks.

We adjusted the hyperparameters within predefined ranges and validated the experimental results on both 7-Scenes and RobotCar datasets. As indicated by the results in Table 4, different initializations of β and γ had minimal impact on the localization error outcomes, overall remaining above the baseline level. Notably, when set to $\beta = 0.0$ and $\gamma = -3.0$, the network exhibited optimal performance. Additionally, sensitivity experiments confirmed the robustness of the hyperparameters β and γ in pose regression tasks. In practical applications, fine-tuning these hyperparameters based on the actual environment can enhance the method's versatility to adapt to various localization tasks.

4.6. Efficiency analysis

To validate the computational efficiency of our proposed CA-Net, we compare its parameter size and inference speed with ResNet50. All experiments are conducted on an NVIDIA GeForce RTX 3090 GPU. As shown in Table 5, CA-Net introduces only 1.2M additional parameters compared to the original ResNet50 (25.6M parameters), resulting in a total of 26.8M parameters. This modest increase stems from the lightweight design of the CAM module, where group convolutions and channel concatenation operations minimize parameter overhead.

We evaluated the inference speed of CA-Net using input images of size 256 × 256, achieving 112 FPS. This is slightly lower than the 128 FPS achieved by the baseline ResNet50 model under the same conditions. Despite the minor decrease, CA-Net maintains real-time performance, indicating that the cascaded attention mechanism enhances feature representation without significantly compromising computational efficiency.

Table 5
Comparison of parameter size and inference speed.

Model	Parameters (M)	FPS
ResNet50	25.6	128
CA-Net (Ours)	26.8	112

Table 6
Ablation study of our model on 7-Scenes and Oxford RobotCar datasets. The model (None) denotes the model without any attention modules. We calculate the median translation error (meters) and orientation error (degrees) on 7-Scenes and Oxford RobotCar datasets. The best results are highlighted.

Scene	None	None+CBAM (Woo et al., 2018)	None+ACmix (Pan et al., 2022)	None+CABo	None+CABo+DAM
			7-Scenes		
Chess	0.17/ 10.12	0.13/ 8.91	0.14/ 8.65	0.15/ 7.56	**0.09/ 5.19**
Fire	0.31/ 9.37	0.27/ 8.52	0.26/ 8.35	0.25/ **8.01**	**0.23**/ 8.79
Heads	0.24/ 17.26	0.21/ 15.83	0.21/ 14.20	**0.19**/ 13.85	**0.19/ 13.56**
Office	0.31/ 8.51	0.26/ 7.84	0.25/ 8.09	0.22/ 8.39	**0.21/ 5.03**
Pumpkin	0.25/ 6.81	0.21/ 6.23	0.22/ 6.00	0.23/ 7.02	**0.17/ 4.91**
Kitchen	0.22/ 8.75	0.19/ 7.64	0.20/ 7.48	0.19/ 6.18	**0.17/ 4.93**
Stairs	0.39/ 10.35	0.34/ 9.87	0.33/ 9.66	0.29/ **9.42**	**0.24**/ 9.68
Average	0.27/ 10.17	0.23/ 8.85	0.23/ 8.35	0.22/ 8.63	**0.19/ 7.44**
			Oxford RobotCar		
FULL1	53.07/ 13.04	41.25/ 11.56	39.95/ 11.10	36.57/ 11.26	**28.89/ 7.86**
FULL2	82.56/ 18.17	68.34/ 15.43	64.85/ 14.85	65.68/ 14.62	**49.68/ 9.72**
Average	67.82/ 15.61	54.80/ 13.50	52.40/ 12.98	51.13/ 12.94	**39.29/ 8.79**

Table 7
Ablation study using ResNet34 backbone on 7-Scenes and Oxford RobotCar datasets.

Scene	None	None+CABo	None+CABo+DAM
		7-Scenes	
Chess	0.18/ 10.25	0.15/ 8.12	**0.11/ 5.87**
Fire	0.35/ 14.60	0.28/ 12.34	**0.25/ 9.15**
Heads	0.27/ 17.82	**0.21/ 13.70**	0.22/ 13.91
Office	0.34/ 9.75	0.25/ 8.97	**0.23/ 5.65**
Pumpkin	0.28/ 7.53	0.24/ 7.88	**0.19/ 5.34**
Kitchen	0.25/ 9.12	0.21/ 6.75	**0.18/ 5.21**
Stairs	0.42/ 11.06	**0.28/ 9.92**	0.30/ 10.12
Average	0.30/ 11.30	0.24/ 9.57	**0.22/ 7.89**
		Oxford RobotCar	
FULL1	58.14/ 14.37	40.25/ 12.09	**31.45/ 8.95**
FULL2	88.73/ 19.25	70.62/ 15.84	**52.31/ 10.67**
Average	73.44/ 16.81	55.93/ 11.74	**41.88/ 9.81**

Table 8
Ablation study using ResNet101 backbone on 7-Scenes and Oxford RobotCar datasets.

Scene	None	None+CABo	None+CABo+DAM
		7-Scenes	
Chess	0.16/ 9.87	0.12/ 6.45	**0.08/ 4.92**
Fire	0.29/ 13.45	0.24/ 10.22	**0.21/ 8.41**
Heads	0.22/ 16.31	0.18/ 13.05	**0.16/ 12.87**
Office	0.29/ 8.62	0.19/ 6.03	**0.16/ 4.85**
Pumpkin	0.23/ 7.12	0.18/ 5.74	**0.15/ 4.52**
Kitchen	0.20/ 8.95	**0.15/ 4.78**	0.16/ 5.11
Stairs	0.35/ 10.84	0.27/ 9.63	**0.22/ 9.05**
Average	0.25/ 10.74	0.19/ 8.03	**0.16/ 7.06**
		Oxford RobotCar	
FULL1	48.92/ 12.58	30.17/ 9.34	**25.14/ 6.82**
FULL2	75.36/ 17.93	58.24/ 13.45	**42.19/ 8.95**
Average	62.14/ 15.26	43.07/ 10.19	**33.67/ 7.89**

4.7. Ablation study

To demonstrate the effectiveness of each module within CA-Net, a series of ablation analyses were conducted on the proposed Cascaded Attention Block (CABo) and Dual Attention Module (DAM), followed by performance evaluations on indoor and outdoor datasets. Specifically, this section designed three versions to assess the effectiveness of CABo and DAM: (1). None: Base module (without CABo and DAM). (2). None

+ CABo: Base version with Cascaded Attention Block (without DAM). (3). None + CABo + DAM: The proposed method CA-Net, comprising both CABo and DAM.

To further highlight the advantage of the proposed CAM module, we also replaced the CAM in CABo with two popular attention mechanisms, CBAM and ACmix, under the same residual structure setting. As shown in Table 6, the CAM-based design achieves consistently competitive results, demonstrating its effectiveness compared to other attention strategies.

Experimental results demonstrate that the CABo block and DAM proposed in this chapter significantly improve the network's localization performance. Compared to the baseline version None, the proposed CA-Net (None + CABo + DAM) achieves a 30% improvement in translation accuracy and a 27% improvement in rotation accuracy on the 7-Scenes dataset. Additionally, on the FULL subset of the RobotCar dataset, it achieves a 42% improvement in translation accuracy and a 44% improvement in rotation accuracy. These results validate the effectiveness of the proposed modules. These modules offer new insights for other absolute pose estimation tasks.

In our main framework, ResNet50 is selected as the default visual encoder. This decision is based on a trade-off between model complexity and performance. ResNet50, as a mid-sized backbone, provides a good balance between parameter efficiency and representational power. Compared to ResNet34, it captures more expressive features without introducing significant computational overhead, and it is also notably lighter than deeper models like ResNet101, which may require more memory and longer training time. Our empirical results show that ResNet50 enables stable training and yields competitive accuracy across multiple datasets.

Furthermore, to evaluate the generality and scalability of the proposed modules, we extend the ablation experiments to use ResNet34 and ResNet101 as alternative backbones. These two architectures represent shallower and deeper configurations, respectively, allowing us to assess the performance of CAM and DAM modules under varying network capacities. The experimental results are shown in Tables 7 and 8.

Both alternative backbones demonstrate consistent improvements with the integration of our attention modules. In particular, the CAM and DAM significantly reduce both position and orientation errors across all tested scenes and datasets. Notably, ResNet101 achieves the best absolute performance due to its deeper structure, while ResNet34 shows that the method remains effective even in lightweight models. These results confirm that the proposed design is adaptable and robust across different network configurations.

5. Conclusion

In this paper, we generate a robust feature representation suitable for absolute pose regression by performing multi-scale feature extraction and constructing dependencies between spatial and channel attention. Specifically, in the CABo module, the splitting and concatenation operations on the feature maps allow the network to capture global information while better preserving and fusing local features at different scales. Additionally, long-range dependencies between features are established to enhance the model's understanding of global context. The dual-stream attention module (DAM) models the semantic interdependencies in the spatial and channel dimensions, adaptively integrating local features with their global dependencies, thus mitigating the impact of dynamic objects and illumination changes on localization performance. Experimental results show that CA-Net significantly improves localization accuracy on both the 7-Scenes and RobotCar datasets. However, we observe that the performance improvement in some challenging scenes, such as *Heads*, is relatively limited. This can be attributed to extreme texture repetitiveness or highly ambiguous geometry in these environments, which makes it difficult for the model to distinguish between visually similar regions, even with enhanced attention mechanisms. While the proposed modules effectively model multi-scale spatial-channel dependencies, future extensions could explore integrating geometric constraints or hybrid attention strategies to further enhance robustness in these cases. In the future, we plan to extend our method to cross-scenario applications and validate it on additional datasets.

CRediT authorship contribution statement

Xiaogang Song: Writing – review & editing, Methodology. **Junjie Tang:** Writing – original draft, Software. **Kaixuan Yang:** Writing – review & editing. **Weixuan Guo:** Writing – review & editing. **Xiaofeng Lu:** Writing – review & editing. **Xinhong Hei:** Writing – review & editing, Supervision.

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 by the National Natural Science Foundation of China (No. 52372418, 62076201), and the Key Research and Development Program of Shaanxi Province of China (No. 2023GXLH-043).

Data availability

The authors do not have permission to share data.

References

Arandjelovic, Relja, Gronat, Petr, Torii, Akihiko, Pajdla, Tomas, Sivic, Josef, 2016. NetVLAD: CNN architecture for weakly supervised place recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5297–5307.

Balntas, Vassileios, Li, Shuda, Prisacariu, Victor, 2018. Relocnet: Continuous metric learning relocalisation using neural nets. In: Proceedings of the European Conference on Computer Vision. ECCV, pp. 751–767.

Blanton, Hunter, Greenwell, Connor, Workman, Scott, Jacobs, Nathan, 2020. Extending absolute pose regression to multiple scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 38–39.

Brachmann, Eric, Krull, Alexander, Nowozin, Sebastian, Shotton, Jamie, Michel, Frank, Gumhold, Stefan, Rother, Carsten, 2017. Dsac-differentiable ransac for camera localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6684–6692.

Brachmann, Eric, Rother, Carsten, 2018. Learning less is more-6D camera localization via 3D surface regression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4654–4662.

Brachmann, Eric, Rother, Carsten, 2021. Visual camera re-localization from RGB and RGB-D images using DSAC. IEEE Trans. Pattern Anal. Mach. Intell. 44 (9), 5847–5865.

Brahmbhatt, Samarth, Gu, Jinwei, Kim, Kihwan, Hays, James, Kautz, Jan, 2018. Geometry-aware learning of maps for camera localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2616–2625.

Cai, Ming, Shen, Chunhua, Reid, Ian, 2019. A Hybrid Probabilistic Model for Camera Relocalization. BMVC Press.

Calonder, Michael, Lepetit, Vincent, Strecha, Christoph, Fua, Pascal, 2010. Brief: Binary robust independent elementary features. In: Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11. Springer, pp. 778–792.

Charco, Jorge L, Sappa, Angel D, Vintimilla, Boris X, Velesaca, Henry O, 2021. Camera pose estimation in multi-view environments: From virtual scenarios to the real world. Image Vis. Comput. 110, 104182.

Chen, Shuai, Wang, Zirui, Prisacariu, Victor, 2021. Direct-posenet: Absolute pose regression with photometric consistency. In: 2021 International Conference on 3D Vision. 3DV, IEEE, pp. 1175–1185.

Dai, Kun, Xie, Tao, Wang, Ke, Jiang, Zhiqiang, Liu, Dedong, Li, Ruifeng, Wang, Jiahe, 2023. Eaainet: An element-wise attention network with global affinity information for accurate indoor visual localization. IEEE Robot. Autom. Lett. 8 (6), 3166–3173.

DeTone, Daniel, Malisiewicz, Tomasz, Rabinovich, Andrew, 2018. Superpoint: Self-supervised interest point detection and description. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 224–236.

Ding, Mingyu, Wang, Zhe, Sun, Jiankai, Shi, Jianping, Luo, Ping, 2019. CamNet: Coarse-to-fine retrieval for camera re-localization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2871–2880.

Esfahani, Mahdi Abolfazli, Wu, Keyu, Yuan, Shenghai, Wang, Han, 2019. DeepD-SAIR: Deep 6-DOF camera relocalization using deblurred semantic-aware image representation for large-scale outdoor environments. Image Vis. Comput. 89, 120–130.

Fischler, Martin A., Bolles, Robert C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24 (6), 381–395.

Gao, Zilin, Xie, Jiangtao, Wang, Qilong, Li, Peihua, 2019. Global second-order pooling convolutional networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3024–3033.

Hu, Jie, Shen, Li, Sun, Gang, 2018. Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7132–7141.

Kang, Xujie, Liu, Kangling, Duan, Jiang, Gong, Yuanhao, Qiu, Guoping, 2024. Adversarial learning-based camera pose-to-image mapping network for synthesizing new view in real indoor environments. ISPRS J. Photogramm. Remote Sens. 212, 27–41.

Kendall, Alex, Cipolla, Roberto, 2016. Modelling uncertainty in deep learning for camera relocalization. In: 2016 IEEE International Conference on Robotics and Automation. ICRA, IEEE, pp. 4762–4769.

Kendall, Alex, Cipolla, Roberto, 2017. Geometric loss functions for camera pose regression with deep learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5974–5983.

Kendall, Alex, Grimes, Matthew, Cipolla, Roberto, 2015. Posenet: A convolutional network for real-time 6-dof camera relocalization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2938–2946.

Kneip, Laurent, Scaramuzza, Davide, Siegwart, Roland, 2011. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In: CVPR 2011. IEEE, pp. 2969–2976.

Li, X., Hu, X., Yang, J., Spatial group-wise enhance: Improving semantic feature learning in convolutional networks. arXiv 2019, arXiv preprint arXiv:1905.09646.

Li, Penghao, Tao, Huanjie, Zhou, Hui, Zhou, Ping, Deng, Yishi, 2025. Enhanced multiview attention network with random interpolation resize for few-shot surface defect detection. Multimedia Syst. 31 (1), 36.

Lowe, David G., 2004. Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. 60, 91–110.

Maddern, Will, Pascoe, Geoffrey, Linegar, Chris, Newman, Paul, 2017. 1 year, 1000 km: The Oxford robotcar dataset. Int. J. Robot. Res. 36 (1), 3–15.

Melekhov, Iaroslav, Ylioinas, Juha, Kannala, Juho, Rahtu, Esa, 2017. Image-based localization using hourglass networks. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 879–886.

Mera-Trujillo, Marcela, Patel, Shivang, Gu, Yu, Doretto, Gianfranco, 2023. Self-supervised interest point detection and description for fisheye and perspective images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6497–6506.

Mur-Artal, Raul, Montiel, Jose Maria Martinez, Tardos, Juan D., 2015. ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Trans. Robot. 31 (5), 1147–1163.

Murillo, Ana Cris, Guerrero, José Jesús, Sagues, C., 2007. Surf features for efficient robot localization with omnidirectional images. In: Proceedings 2007 IEEE International Conference on Robotics and Automation. IEEE, pp. 3901–3907.

Ott, Felix, Feigl, Tobias, Loffler, Christoffer, Mutschler, Christopher, 2020. Vipr: Visual-odometry-aided pose regression for 6DoF camera localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 42–43.

Pan, Xuran, Ge, Chunjiang, Lu, Rui, Song, Shiji, Chen, Guanfu, Huang, Zeyi, Huang, Gao, 2022. On the integration of self-attention and convolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 815–825.

Radwan, Noha, Valada, Abhinav, Burgard, Wolfram, 2018. Vlocnet++: Deep multitask learning for semantic visual localization and odometry. IEEE Robot. Autom. Lett. 3 (4), 4407–4414.

Revaud, Jerome, De Souza, Cesar, Humenberger, Martin, Weinzaepfel, Philippe, 2019. R2d2: Reliable and repeatable detector and descriptor. Adv. Neural Inf. Process. Syst. 32.

Sarlin, Paul-Edouard, Cadena, Cesar, Siegwart, Roland, Dymczyk, Marcin, 2019. From coarse to fine: Robust hierarchical localization at large scale. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12716–12725.

Sarlin, Paul-Edouard, DeTone, Daniel, Malisiewicz, Tomasz, Rabinovich, Andrew, 2020. Superglue: Learning feature matching with graph neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4938–4947.

Sattler, Torsten, Maddern, Will, Carlone, Luca, Henein, Michael, Toft, Carl, Hammarstrand, Lars, Stenborg, Erik, Schön, Thomas B., Pollefeys, Marc, Sivic, Josef, et al., 2019. Understanding the limitations of CNN-based absolute camera pose regression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3302–3312.

Shavit, Yoli, Ferens, Ron, 2021. Do we really need scene-specific pose encoders? In: 2020 25th International Conference on Pattern Recognition. ICPR, IEEE, pp. 3186–3192.

Shavit, Yoli, Ferens, Ron, Keller, Yosi, 2024. Learning single and multi-scene camera pose regression with transformer encoders. Comput. Vis. Image Underst. 243, 103982.

Shotton, Jamie, Glocker, Ben, Zach, Christopher, Izadi, Shahram, Criminisi, Antonio, Fitzgibbon, Andrew, 2013. Scene coordinate regression forests for camera relocalization in RGB-D images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2930–2937.

Song, Xiaogang, Li, Hongjuan, Liang, Li, Shi, Weiwei, Xie, Guo, Lu, Xiaofeng, Hei, Xinhong, 2024. TransBoNet: Learning camera localization with transformer bottleneck and attention. Pattern Recognit. 146, 109975.

Sun, Jiaming, Shen, Zehong, Wang, Yuang, Bao, Hujun, Zhou, Xiaowei, 2021. LoFTR: Detector-free local feature matching with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8922–8931.

Taira, Hajime, Okutomi, Masatoshi, Sattler, Torsten, Cimpoi, Mircea, Pollefeys, Marc, Sivic, Josef, Pajdla, Tomas, Torii, Akihiko, 2018. InLoc: Indoor visual localization with dense matching and view synthesis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7199–7209.

Valada, Abhinav, Radwan, Noha, Burgard, Wolfram, 2018. Deep auxiliary learning for visual localization and odometry. In: 2018 IEEE International Conference on Robotics and Automation. ICRA, IEEE, pp. 6939–6946.

Wang, Bing, Chen, Changhao, Lu, Chris Xiaoxuan, Zhao, Peijun, Trigoni, Niki, Markham, Andrew, 2020a. Atloc: Attention guided camera localization. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 06, pp. 10393–10401.

Wang, Zhenguang, Tao, Huanjie, Zhou, Hui, Deng, Yishi, Zhou, Ping, 2025. A content-style control network with style contrastive learning for underwater image enhancement. Multimedia Syst. 31 (1), 1–13.

Wang, Qilong, Wu, Banggu, Zhu, Pengfei, Li, Peihua, Zuo, Wangmeng, Hu, Qinghua, 2020b. ECA-Net: Efficient channel attention for deep convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11534–11542.

Wang, Jingwen, Yu, Hongshan, Lin, Xuefei, Li, Zechuan, Sun, Wei, Akhtar, Naveed, 2024. EFRNet-VL: An end-to-end feature refinement network for monocular visual localization in dynamic environments. Expert Syst. Appl. 243, 122755.

Wang, Qing, Zhang, Jiaming, Yang, Kailun, Peng, Kunyu, Stiefelhagen, Rainer, 2022. Matchformer: Interleaving attention in transformers for feature matching. In: Proceedings of the Asian Conference on Computer Vision. pp. 2746–2762.

Woo, Sanghyun, Park, Jongchan, Lee, Joon-Young, Kweon, In So, 2018. Cbam: Convolutional block attention module. In: Proceedings of the European Conference on Computer Vision. ECCV, pp. 3–19.

Wu, Jian, Ma, Liwei, Hu, Xiaolin, 2017. Delving deeper into convolutional neural networks for camera relocalization. In: 2017 IEEE International Conference on Robotics and Automation. ICRA, IEEE, pp. 5644–5651.

Xie, Tao, Dai, Kun, Wang, Ke, Li, Ruifeng, Wang, Jiahe, Tang, Xinyue, Zhao, Lijun, 2022. A deep feature aggregation network for accurate indoor camera localization. IEEE Robot. Autom. Lett. 7 (2), 3687–3694.

Xu, Meng, Wang, Lingfeng, Ren, Jian, Poslad, Stefan, 2020. Use of LSTM regression and rotation classification to improve camera pose localization estimation. In: 2020 IEEE 14th International Conference on Anti-Counterfeiting, Security, and Identification. ASID, IEEE, pp. 6–10.

Xue, Fei, Wang, Xin, Yan, Zike, Wang, Qiuyuan, Wang, Junqiu, Zha, Hongbin, 2019. Local supports global: Deep camera relocalization with sequence enhancement. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2841–2850.

Zangeneh, Fereidoon, Bruns, Leonard, Dekel, Amit, Pieropan, Alessandro, Jensfelt, Patric, 2023. A probabilistic framework for visual localization in ambiguous scenes. In: 2023 IEEE International Conference on Robotics and Automation. ICRA, IEEE, pp. 3969–3975.

Zhang, Hang, Dana, Kristin, Shi, Jianping, Zhang, Zhongyue, Wang, Xiaogang, Tyagi, Ambrish, Agrawal, Amit, 2018. Context encoding for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7151–7160.

Zhang, Han, Goodfellow, Ian, Metaxas, Dimitris, Odena, Augustus, 2019. Self-attention generative adversarial networks. In: International Conference on Machine Learning. PMLR, pp. 7354–7363.

Zhang, Hu, Zu, Keke, Lu, Jian, Zou, Yuru, Meng, Deyu, 2021. Epsanet: An efficient pyramid split attention block on convolutional neural network. arXiv preprint arXiv:2105.14447.

Zhou, Qunjie, Sattler, Torsten, Leal-Taixe, Laura, 2021. Patch2pix: Epipolar-guided pixel-level correspondences. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4669–4678.