Learning Multi-Scene Absolute Pose Regression with Transformers

Yoli Shavit Ron Ferens Yosi Keller
Faculty of Engineering, Bar Ilan University, Ramat-Gan, Israel
{yolisha, ronferens, yosi.keller}@gmail.com

Abstract

Absolute camera pose regressors estimate the position and orientation of a camera from the captured image alone. Typically, a convolutional backbone with a multi-layer perceptron head is trained using images and pose labels to embed a single reference scene at a time. Recently, this scheme was extended for learning multiple scenes by replacing the MLP head with a set of fully connected layers. In this work, we propose to learn multi-scene absolute camera pose regression with Transformers, where encoders are used to aggregate activation maps with self-attention and decoders transform latent features and scenes encoding into candidate pose predictions. This mechanism allows our model to focus on general features that are informative for localization while embedding multiple scenes in parallel. We evaluate our method on commonly benchmarked indoor and outdoor datasets and show that it surpasses both multi-scene and state-of-the-art single-scene absolute pose regressors. We make our code publicly available from https://github.com/yolish/multi-scene-pose-transformer.

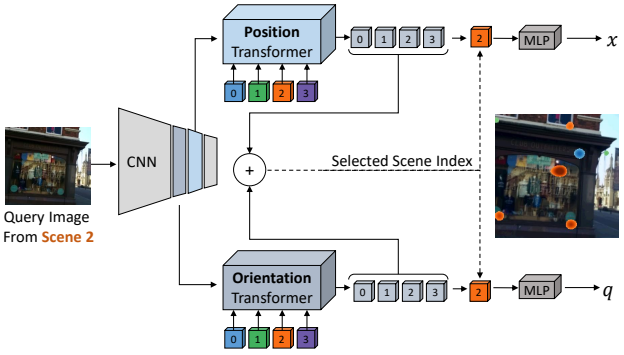


Figure 1: Multi-scene absolute pose regression with Transformers. Two Transformers separately attend to position- and orientation- informative features from a convolutional backbone. Scene-specific queries $(0-3)$ are further encoded with aggregated activation maps into latent representations, from which a single output is selected. The strongest response, shown as an overlaid color-coded heatmap of attention weights, is obtained with the output associated with the input image's scene. The selected outputs are used to regress the position x and the orientation q.

1. Introduction

Localizing a camera using a query image is a key task in many computer vision applications, such as indoor navigation, augmented reality and autonomous driving, to name a few. Contemporary approaches for estimating the position and orientation of a camera offer different trade-offs between accuracy, runtime, and memory. For example, hierarchical localization pipelines [26, 30, 25] achieve state-of-the-art (SOTA) pose accuracy, but are relatively slow (response time of hundreds of milliseconds) and require a large memory footprint and client-server connectivity. These approaches employ image retrieval (IR) to fetch images that are similar to the query image, followed by local features extraction and matching. The extracted 2D-2D matches are mapped to 2D-3D correspondences via depth or a 3D point cloud, and are then used to estimate the camera pose with Perspective-n-Point (PnP) and RANSAC [13]. Absolute pose regressors (APRs), on the other hand, estimate the camera pose with a single forward pass, using only the query image. They are an order of magnitude faster and can be deployed as a standalone application on a thin client due to their small memory footprint. Unfortunately, APRs are also an order of magnitude less accurate compared to hierarchical localization pipelines and to other methods utilizing 3D data at inference time [27]. Moreover, most APRs are designed to embed a single scene at a time, implying that for a dataset with N scenes (for example, a hospital with many wards and rooms), N models are required to be trained, deployed and chosen from during inference. In this work, we focus on improving the accuracy of APRs while extending the current single-scene paradigm for learning multiple scenes in parallel.

The formulation of absolute camera pose regression was first suggested by Kendall et al. [17]. Following the success of convolutional neural networks (CNNs) in learning different computer vision tasks, the authors suggested to adapt

a GoogLeNet architecture to camera pose regression by attaching a multi-layer perceptron (MLP) head to regress the camera position and orientation. The proposed architecture denoted PoseNet, offered a novel, fast and lightweight solution for camera localization. However, it also suffered from low accuracy and limited generalization. Various absolute pose regression methods were suggested to address these issues, proposing modifications to the backbone and MLP architecture [20, 21, 35, 37, 29, 36, 8], as well as different loss formulations and optimization strategies [15, 16, 28]. Despite their variety, such APRs share two common traits: (1) employing a CNN backbone to output a single global latent vector which is used for regressing the pose (2) training a model per scene (scene-specific APRs).

Recently, Blanton et al. [3] suggested a method for extending single-scene absolute pose regression to a multi-scene paradigm. Similarly to existing APRs, this method applies a CNN backbone for generating a latent global descriptor of the image. However, instead of using a single scene-specific MLP, it trains a set of Fully Connected (FC) layers, with a layer per scene, which is indexed based on the predicted scene identifier. While offering a new general framework for optimizing a single model for multiple scenes, this method was unable to match the accuracy of contemporary SOTA APRs.

In this work, we propose a novel formulation of multi-scene absolute pose regression, inspired by recent successful applications of Transformers to computer vision tasks such as object detection [9] and image recognition [11]. These works demonstrated the effectivity of *encoders* in focusing on latent features (in image patches or activation maps) that are informative for particular tasks, through self-attention aggregation. In addition, *decoders* were shown to successfully generate multiple independent predictions, corresponding to queries, based on the input embedding [9]. Similarly, we propose to apply Transformers to multi-scene absolute pose regression, using *encoders* to focus on pose-informative features and *decoders* to transform encoded scene identifiers to latent pose representations (Fig. 1). As pose estimation involves two different tasks (position and orientation estimation), related to different visual cues, we apply a shared convolutional backbone at two different resolutions and use two different Transformers, one per task. The decoders' outputs are used to classify the scene and select the respective position and orientation embeddings, from which the position and orientation vectors are regressed.

We evaluate our approach on two commonly benchmarked datasets, consisting of multiple outdoor and indoor localization challenges. We show that our method not only provides a new SOTA accuracy for *multi-scene APR* localization, but importantly provides a new SOTA for *single-scene APRs*. Moreover, we show that our approach achieves competitive results even when trained across multiple datasets of significantly different characteristics. We further conduct multiple ablations to evaluate the sensitivity of our model to different design choices and analyze its scalability in terms of runtime and memory. In summary, our main contributions are as follows:

- We propose a novel formulation for multi-scene absolute pose regression using Transformers.
- We experimentally demonstrate that self-attention allows aggregation of positional and rotational image cues.
- Our approach is shown to achieve new SOTA accuracy for both multi-scene and single-scene APRs across contemporary outdoor and indoor localization benchmarks.

2. Related Work

Camera pose estimation methods can be divided into several families, depending on the inputs at inference time and on their algorithmic characteristics.

Image Retrieval methods learn global image descriptors for retrieving database images that depict the vicinity of the area captured by the query image. They are commonly employed by pose estimation methods such as structure-based hierarchical localization pipelines [30, 25, 22, 12] and relative pose regression methods [2, 18, 10]. IR can also be applied for estimating the camera pose by taking the pose of the most similar fetched image, or by interpolating the poses of several visually close images. Such approaches require both storing and searching through large databases with pose labels. Recently, Sattler et al. [27] proposed an IR-based baseline for camera pose regression to illustrate the limitations of APRs, as no regressor was able to consistently surpass it on multiple localization tasks.

3D-based Localization methods, also referred to as structure-based methods [26, 27], include camera pose estimation techniques that utilize the correspondences between 2D image positions and 3D world coordinates for camera localization with PnP and RANSAC. Hierarchical pose estimation pipelines [30, 25, 22, 12] are based on a two-phase approach, utilizing global (IR) and local matching. Each query image to be localized, is first encoded using a CNN trained for IR, and a relatively small set of nearest neighbors is retrieved from a large-scale image dataset. Tentative 2D-2D correspondences are estimated by matching local image features, and are then mapped into 2D-3D matches. The resulting matches are passed to PnP-RANSAC for estimating the camera pose. Such pipelines were shown to achieve SOTA accuracy on large-scale benchmarks with challenging conditions [30, 25]. However, these approaches are slower than other localization methods by an order or

two orders of magnitude and are typically deployed with a client-server architecture due to the large storage space needed. A different body of works directly regresses the 3D coordinates from 2D positions in the image. Brachmann and Rother derived the DSAC [4] and the follow-up DSAC++ [5] schemes, where a CNN architecture is trained to estimate the 3D locations of the pixels in the query image, in order to establish 2D-3D correspondences for estimating the camera pose with PnP-RANSAC. These methods only require the query image at inference time, and achieve SOTA accuracy, which is competitive with hierarchical localization pipelines. However, similar to APRs, a model needs to be trained per scene. In addition, these method are challenging to implement, require a long time to converge and are slower (100ms) by order of magnitude compared to absolute pose regression approaches (10ms) at inference time [3]. They also suffer from a non-deterministic behavior due to the inherent randomness of RANSAC.

Relative Pose Regression methods combine camera pose regression with an IR scheme. The absolute camera pose is computed by first estimating the *relative* motion (translation and rotation) between the query image and a set of reference images, for which the ground truth pose is known. An IR scheme is applied to retrieve a set of nearest neighbors images, and a relative motion regression is separately computed between the query image and each of the retrieved images, followed by pose interpolation. The learning thus focuses on regressing the relative pose given a pair of images [2, 18, 10]. These relative pose regressors (RPRs) were shown to generalize better than APRs and to improve the accuracy on small scale indoor benchmarks [10]. However, similar to other IR-based schemes, such approaches require a retrieval phase and a pose-labelled database during inference time. When the images are sequentially acquired over time, combining relative and absolute regression was shown to significantly improve the pose accuracy [33, 24].

Absolute Pose Regression was first proposed by [17] to directly regress the position and orientation of the camera, given the input image, by attaching an MLP head to a GoogLeNet backbone. The resulting architecture, named PoseNet, was far less accurate than 3D-based methods, but enabled pose estimation using a single forward pass, offering a much lighter and faster localization alternative. In order to improve the localization accuracy, contemporary APRs studied different CNN backbones [20, 21, 37, 29] and MLP heads [37, 21]. Overfitting was addressed by averaging the predictions of multiple models with randomly dropped activations [15], or by reducing the dimensionality of the global image encoding using Long-Short-Term-Memory (LSTM) layers [35]. Other works focused on the loss formulation for absolute pose regression in order to enable adaptive weighing of the position and orientation associated errors. Kendall et al. [16] suggested to optimize the parameters balancing both losses for improving accuracy and avoiding manual fine-tuning. This formulation was adopted by many pose regressors. Alternative orientation representations were also proposed to improve the pose loss [37, 7]. The use of additional sensors, such as inertial sensors, was also suggested to improve the localization accuracy [7]. More recently, Wang et al. [36] proposed to use attention for guiding the regression process by applying self-attention on the output of the CNN backbone. The new attention-based representation was used to regress the pose with an MLP head. While many modifications were proposed to the architecture and loss originally formulated by Kendall et al., the main paradigm remained the same: (1) employing a CNN backbone to output a global latent vector which is used for absolute pose regression (2) training a separate model per scene.

Multi-Scene Absolute Pose Regression methods aim to extend the absolute pose regression paradigm for learning a *single* model on *multiple* scenes. Blanton et al. proposed the Multi-Scene PoseNet (MSPN), a novel multi-scene absolute pose regression approach [3], where the network first classifies the particular scene related to the input image, and then uses it to index a set of scene-specific weights for regressing the pose. An activation map from a CNN backbone, which is shared across scenes, is used both for scene classification and regressing the pose. A fully connected layer with SoftMax predicts the scene and is trained via Binary Cross Entropy. A set of FC layers, one per scene, is trained for absolute camera pose regression with a set of scene-specific parameterized losses. The notion of multi-scene camera pose estimation was also applied to 3D-based methods which regresses the 3D coordinates from image pixels. However, the suggested framework still involved training multiple models (one per scene) and then selecting the most appropriate model using a mixture-of-experts strategy [6].

In this work, we focus on learning a *single* unified deep learning model for absolute pose regression across multiple scenes. Our method is thus closely related to single- and multi-scene absolute pose regression and we compare it to leading architectures (APRs) in this field. We include additional background on Attention and Transformers in the supplementary material.

3. Multi-Scene Absolute Camera Pose Regression with Transformers

A single/multi-scene APR localizes the capturing camera with a forward pass on the input image. The camera pose $\mathbf{p}$ is typically represented by the tuple $< \mathbf{x}, \mathbf{q} >$ where $\mathbf{x} \in \mathbb{R}^3$ is the position of the camera in the world coordinates and $\mathbf{q} \in \mathbb{R}^4$ is the quaternion encoding of its 3D orientation. Following the success of recent visual Transformers [9, 11], we employ separate positional and orien-

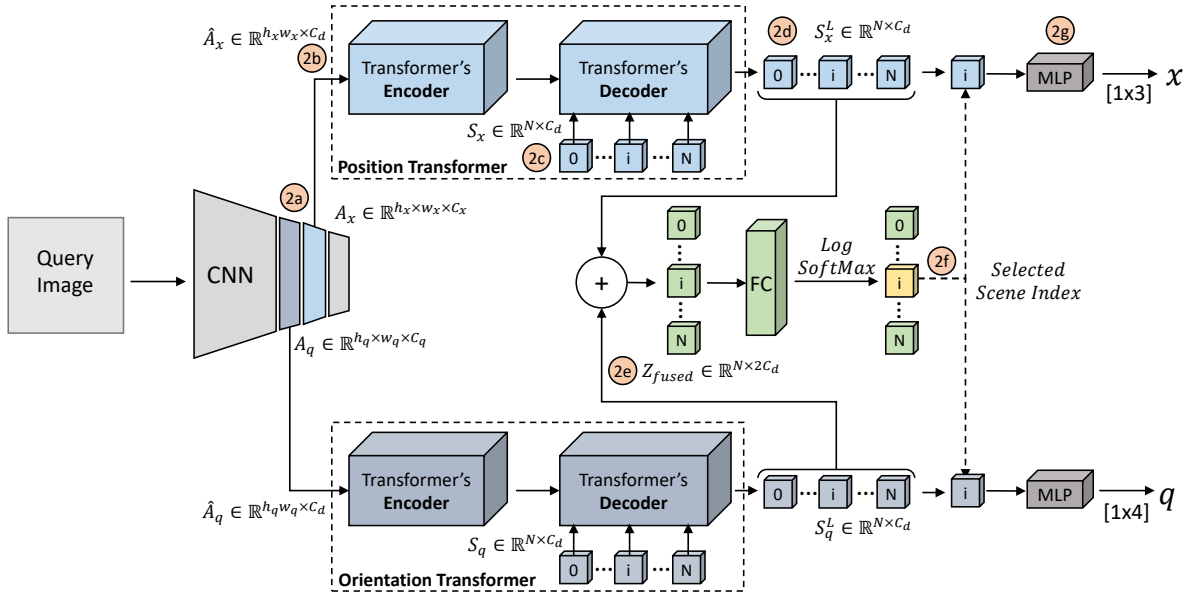


Figure 2: The architecture of our proposed model.

tational *Transformer Encoders* for adaptive aggregation of (flattened) intermediate activation maps computed by a convolutional backbone. In particular, as depicted in Fig. 3, the positional and orientational encoders emphasize different image cues: corner and blob-like image cues are position-informative, in contrast to the elongated edges emphasized by the orientational encoder. In order to attend to N scenes we also apply separate positional and orientational *Transformer Decoders*, that are queried by $\{\mathbf{x}_i\}_1^N$ and $\{\mathbf{q}_i\}_1^N$, for the position and orientation embeddings per scene, respectively. The corresponding output sequences, $\{\mathbf{X}_i\}_1^N$ and $\{\mathbf{Q}_i\}_1^N$, respectively, encode the localization parameters *per scene*. This architecture is inspired by the DETR approach [9], where a single activation map is queried by multiple queries, each related to a different task. Together, the encoder-decoder Transformer architecture allows to attend to localization-informative image content while learning multiple scenes at once. In order to regress the pose, the scene is first classified by concatenating $\{\mathbf{X}_i\}_1^N$ and $\{\mathbf{Q}_i\}_1^N$, and the embeddings of the detected scene $\{\mathbf{X}_i, \mathbf{Q}_i\}$ are regressed by the MLP heads.

3.1. Network Architecture

The architecture of our model is shown in Fig. 2. Given an image $\mathbf{I} \in \mathbb{R}^{H \times W \times C}$, we sample a convolutional backbone at two different resolutions and take an activation map per regression task: $A_{\mathbf{x}}$ and $A_{\mathbf{q}}$, for position and orientation regression, respectively (Fig. 2a). In order to transform activation maps into Transformer-compatible inputs, we follow the same sequence preparation procedure as in [9]. An activation map $\mathbf{A} \in \mathbb{R}^{H_a \times W_a \times C_a}$ is first converted to a sequential representation $\widehat{\mathbf{A}} \in \mathbb{R}^{H_a \cdot W_a \times C_d}$ using a 1×1 convolution (projecting to dimension C_d) followed by flattening (Figure 2b). Each position in the activation map is further assigned with a learned encoding to preserve the spatial information of each location. In order to reduce the number of parameters, two one-dimensional encodings are separately learned for the X,Y axes. Specifically, for an activation map $\mathbf{A}$ we define the sets of positional embedding vectors $\mathbf{E}_u \in \mathbb{R}^{(W_a) \times C_d/2}$ and $\mathbf{E}_v \in \mathbb{R}^{(H_a) \times C_d/2}$, such that a spatial position (i, j), $i \in 1..H_a$, $j \in 1..W_a$, is encoded by the concatenating of the two corresponding embedding vectors:

$$\mathbf{E}_{pos}^{i,j} = \begin{bmatrix} \mathbf{E}_u^j \\ \mathbf{E}_v^i \end{bmatrix} \in \mathbb{R}^{C_d}. \tag{1}$$

The processed sequence, serving as input to the Transformer is thus given by:

$$\mathbf{Z}_{\widehat{A}}^0 = \widehat{\mathbf{A}} + \mathbf{E}_{\mathbf{A}} \in \mathbb{R}^{H_a \cdot W_a \times C_d} \tag{2}$$

where $\mathbf{E}_{\mathbf{A}}$ is the positional encoding of A. This processing is applied separately for each of the two activation maps (for the position and orientation Transformers, respectively).
We use the Transformer architecture described in [9], using standard Encoder and Decoder architectures modified to add the position encodings at each attention layer. A Transformer Encoder is composed of L identical layers each consisting of multi-head attention (MHA) and multi-layer perceptron (MLP) modules. Each layer l, $l = 1..L$, performs the following computation by applying a LayerNorm (LN)

[1] before each module and adding back the input with residual connections:

$$\mathbf{Z}^{l'}_{\widehat{A}} = MHA(LN(\mathbf{Z}^{l-1}_{\widehat{A}})) + \mathbf{Z}^{\mathbf{l-1}}_{\widehat{\mathbf{A}}} \in \mathbb{R}^{H_a \cdot W_a \times C_d} \quad (3)$$

$$\mathbf{Z}^{l}_{\widehat{A}} = MLP(LN(\mathbf{Z}^{l'}_{\widehat{A}})) + \mathbf{Z}^{\mathbf{l'}}_{\widehat{\mathbf{A}}} \in \mathbb{R}^{H_a \cdot W_a \times C_d} \quad (4)$$

At the final layer, L the output is passed through an additional normalization:

$$\mathbf{Z}^{L}_{\widehat{A}} = LN(\mathbf{Z}^{L}_{\widehat{A}}) \quad (5)$$

Given a dataset with N scenes, the Transformer Decoders first applies self-attention as in Eq. 3 to the two learnt query sequences, $\{\mathbf{x}_i\}_1^N$ and $\{\mathbf{q}_i\}_1^N$ (Fig. 2c), for the position and orientation decoders, respectively. Eqs. 3-4 are then applied again, but this time computing encoder-decoder attention instead of self-attention. As oppose to earlier auto-regressive decoders [34], this architecture outputs predictions in parallel for all positions. We refer the reader to [34, 9] for detailed definitions of the MHA operation and parallel decoding. The Transformers' Decoders output the sequences $\{\mathbf{X}_i\}_1^N$ and $\{\mathbf{Q}_i\}_1^N$, with a latent embedding for each scene (Fig. 2d). However, given a query image, only one position corresponds to the scene from which the image was taken. In order to select the appropriate scene, we append the outputs of the two transformers (Fig. 2e) as $\{\mathbf{Z}_i\}_1^N$ such that

$$\mathbf{Z}_i = \begin{bmatrix} \mathbf{X}_i \\ \mathbf{Y}_i \end{bmatrix} \in \mathbb{R}^{2C_d}, \quad (6)$$

and pass them through a fully connected layer followed by Log SoftMax. The vectors at the position with the maximal probability are then chosen (Fig. 2f). The selected Transformers outputs $\{\mathbf{X}_i, \mathbf{Q}_i\}$ are passed to a respective MLP head with one hidden layer and gelu non-linearity to regress the target vectors, $\mathbf{x}$ (Fig. 2g) or $\mathbf{q}$.

3.2. Multi-Scene Camera Pose Loss

We train our model to minimize both the position loss $L_{\mathbf{x}}$ and the orientation loss $L_{\mathbf{q}}$, with respect to a ground truth pose $\mathbf{p}_0 =< \mathbf{x}_0, \mathbf{q}_0 >$, given by:

$$L_{\mathbf{x}} = ||\mathbf{x}_0 - \mathbf{x}||_2 \quad (7)$$

$$L_{\mathbf{q}} = ||\mathbf{q_0} - \frac{\mathbf{q}}{||\mathbf{q}||}||_2 \quad (8)$$

where q is normalized to a unit norm quaternion in order ensure it is a valid orientation encoding. We combine the two losses using the camera pose loss formulation suggested by Kendall et al. [16]:

$$L_{\mathbf{p}} = L_{\mathbf{x}} \exp(-s_{\mathbf{x}}) + s_{\mathbf{x}} + L_{\mathbf{q}} \exp(-s_{\mathbf{q}}) + s_{\mathbf{q}} \quad (9)$$

where $s_{\mathbf{x}}$ and $s_{\mathbf{q}}$ are learned parameters controlling the balance between the two losses. Since our model is also required to classify the scene from which the image was taken, we further add a Negative Log Likelihood (NLL) loss term, computed with respect to the ground truth scene index s_0. Given an estimated pose p and the log probability distribution s of the predicted scene, our overall loss is given by:

$$L_{\mathbf{multi-scene}} = L_{\mathbf{p}} + NLL(\mathbf{s}, \mathbf{s_0}) \quad (10)$$

3.3. Implementation Details

Our model is implemented in PyTorch [23]. We use a pre-trained EfficientNet-B0 [31], available through an open source implementation [19], and take $A_{\mathbf{x}}$ and $A_{\mathbf{q}}$ at two different resolutions: $A_{\mathbf{x}} \in \mathbb{R}^{14\times14\times112}$ and $A_{\mathbf{q}} \in \mathbb{R}^{28\times28\times40}$. We set $C_d = 256$ for the dimension of inputs of the Transformer components. All encoders and decoders consist of six layers with gelu non-linearity and with a dropout of $p = 0.1$. Each (encoder/decoder) layer uses four heads MHA and a two-layers MLP with a hidden dimension $C_h = C_d$.
The two MLP heads, regressing the position and orientation vectors, respectively, expand the decoder dimension to 1024 with a single hidden layer. Our code is publicly available from https://github.com/yolish/multi-scene-pose-transformer, providing the model implementation along with a training and evaluation framework.

4. Experimental Results

4.1. Experimental Setup

Datasets. We evaluate our approach using the Cambridge Landmarks [17] and the 7Scenes [14] datasets, which are commonly used for evaluating pose regression methods. The Cambridge Landmarks dataset consists of six medium-sized scenes ($\sim 900 - 5500m^2$) set in an urban environment. For our comparative analysis, we consider four scenes that are typically benchmarked by APRs. The 7Scenes dataset includes seven small-scale scenes ($\sim 1 - 10m^2$) set in an office indoor environment.
Training Details. We optimize our model to minimize the loss in Eq. 10 using Adam, with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-10}$. The loss parameters (Eq. 9) are initialized as in [33]. Throughout all experiments we use a batch size of 8 and an initial learning rate of $\lambda = 10^{-4}$. At train time, the decoder output is selected using the ground truth scene index, and the estimated scene index is used only for evaluating the NLL loss. During inference, the scene index is not known and we rely on the prediction of our model, taking the index with the highest (log) probability. Note that instead of training a model per scene (typically with scene-specific training hyper-parameters) as in single APRs, here we train a single model for *all* the scenes together. Ad-

Table 1: Comparative analysis of MSPN and our method on the Cambridge Landmarks dataset (outdoor localization). We report the median position/orientation error in meters/degrees for each method. Bold highlighting indicates better performance.

Method	K. College	Old Hospital	Shop Facade	St. Mary
MSPN [3]	1.73/3.65	2.55/4.05	2.92/7.49	2.67/6.18
MS-Transformer (ours)	**0.83/1.47**	**1.81/2.39**	**0.86/3.07**	**1.62/3.99**

Table 2: Comparative analysis of our method and MSPN on the 7Scenes dataset (indoor localization). We report the median position/orientation error in meters/degrees for each method. Bold highlighting indicates better performance.

Method	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs
MSPN[3]	**0.09**/4.76	0.29/10.5	0.16/13.1	**0.16**/6.8	0.19/5.5	**0.21**/6.61	0.31/11.63
MS-Transformer (ours)	0.11/**4.66**	**0.24/9.6**	**0.14/12.19**	0.17/**5.66**	**0.18/4.44**	**0.17/5.94**	**0.26/8.45**

Table 3: Localization results for the Cambridge Landmarks dataset. We report the average of median position/orientation errors in meters/degrees and the respective rankings. Best results are highlighted in bold.

Method	Average [m/deg]	Ranks
Single-scene APRs		
PoseNet [17]	2.09/6.84	10/11
BayesianPN [15]	1.92/6.28	8/10
LSTM-PN [35]	1.30/5.52	2/9
SVS-Pose [21]	1.33/5.17	3/7
GPoseNet [8]	2.08/4.59	6/3
PoseNet-Learnable [16]	1.43/2.85	5/2
GeoPoseNet [16]	1.63/2.86	6/3
MapNet [7]	1.63/3.64	6/5
IRPNet [29]	1.42/3.45	4/4
Multi-scene APRs		
MSPN [3]	2.47/5.34	11/8
MS-Transformer (Ours)	**1.28/2.73**	**1/1**

ditional details of augmentation and training are provided in the supplementary material. All experiments reported in this paper were performed on an 8Gb NVIDIA GeForce GTX 1080 GPU.

4.2. Comparative Analysis of APRs

Our method aims to offer a multi-scene absolute pose regression paradigm which also improves the accuracy obtained by current APRs. Hence, we compare our approach both to MSPN, which is, to the best of our knowledge, the only other multi-scene APR, as well as to leading single-scene APRs. We do not include the localization schemes detailed in Section 2, which are an order of magnitude slower (3D-based scene coordinate regression [4, 5]), or that utilize additional data at inference time (localization pipelines [30, 25, 22, 12] and relative pose regression [2, 18, 10]). Tables 1 and 2 show the results obtained with our method (MS-Transformer) and with MSPN on the CambridgeLandmarks and the 7Scenes datasets, respectively.

Table 4: Localization results for the 7Scenes dataset. We report the average of median position/orientation errors in meters/degrees and the respective rankings. Best results are highlighted in bold.

Method	Average [m/deg]	Ranks
Single-scene APRs		
PoseNet [17]	0.44/10.4	10/11
BayesianPN [15]	0.47/9.81	11/8
LSTM-PN [35]	0.31/9.86	8/9
GPoseNet [8]	0.31/9.95	8/8
PoseNet-Learnable [16]	0.24/7.87	7/4
GeoPoseNet [16]	0.23/8.12	5/5
MapNet [7]	0.21/7.78	4/3
IRPNet [29]	0.23/8.49	5/7
AttLoc [36]	0.20/7.56	2/2
Multi-scene APRs		
MSPN [3]	0.20/8.41	2/6
MS-Transformer (Ours)	**0.18/ 7.28**	**1/1**

Table 5: Localization results with single-scene, multi-scene and multi-dataset learning. We report the average of median position/orientation errors in meters/degrees for the CambridgeLandmarks and 7Scenes datasets.

APR Method	CambridgeLand. [m/deg]	7Scenes [m/deg]
Single-scene [16]	1.43/2.85	0.24/7.87
Multi-scene (Ours)	1.28/2.73	0.18/7.28
Multi-dataset (Ours)	1.50/ 2.57	0.22/6.78

Since MSPN was trained on different scene combinations from the CambridgeLandmarks dataset, we take the best performing model reported by the authors on this dataset [3]. Our method consistently outperforms MSPN across outdoor and indoor scenes, reducing both position and orientation errors.

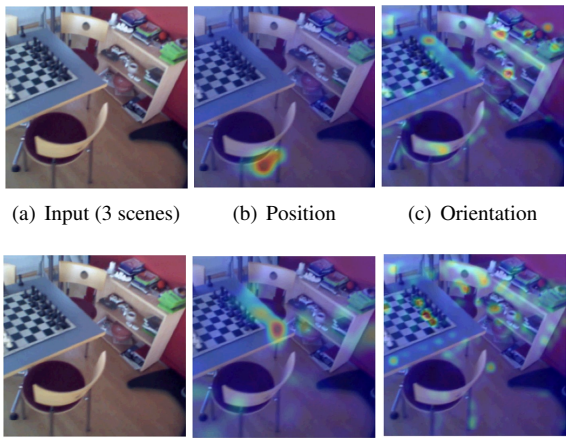
We further compare our results to contemporary absolute pose regression solutions. Tables 3-4 show the performance obtained by different APRs, MSPN and our method on

the CambridgeLandmarks and the 7Scenes datasets, respectively. We report the average of median position and orientation errors across all scenes in each dataset and the respective ranking (where top-1 corresponds to the smallest error). Our method ranks first on both indoor and outdoor localization, achieving the smallest position and orientation errors. Interestingly, the two best performing APRs on the 7Scenes dataset (AttLoc and our method) both use the attention mechanism for pose regression.

We can further extend the notion of multi-scene learning to multi-dataset learning, where a single model is trained on completely separate datasets, potentially displaying different challenges and attributes. In order to evaluate the effect of such an extension, we train our model on both the 7Scenes and Cambridge Landmarks datasets together. Table 5 shows the average pose error per dataset for a *state-of-the-art* single-scene APR [16] and our model, trained in multi-scene and in multi-dataset modes. Although some degradation is observed when training on both datasets together, our model still maintains competitive performance and outperforms the single-scene model. This is despite the two datasets depicting significantly different environments and challenges (mid-scale outdoor versus small-scale indoor). We also evaluate the ability of our model to correctly classify the scene of the input query image. Our model achieves an average accuracy of 98.9% (across scenes) allowing for a reliable selection of the decoder output, which is essential for regressing the pose. Additional analysis of our model scalability (runtime and memory) is provided in the supplementary material.

4.3. Attention Maps Visualization and Interpretation

The visualization of attention maps in attention-based schemes provides an intuitive interpretation of the visual cues captured by the Transformer Encoder. For this purpose, we visualize the upsampled attention weights of the last encoder layer as heatmaps that are overlaid on the input images. Figure 3 shows the attention map of an image taken from the *Chess* scene in the 7Scenes dataset. We show the activations when training on three and on seven scenes (top and bottom row, respectively). Training on more scenes allows the network to better capture informative image cues for both positional and orientational embedding. In particular, positional attention focuses on corner-like objects, while orientational attention emphasizes elongated edges. We also visualize the attentions $\{\mathbf{X}_i\}_1^N$ at the outputs of the positional decoder for an image from the *OldHospital* scene (Fig. 4). Each activation corresponds to a particular scene. Indeed, the activations corresponding to the OldHospital scene (Fig. 4b) are significantly stronger. We include additional analysis of the decoder attention in the supplementary material.



(a) Input (3 scenes) (b) Position (c) Orientation

(d) Input (7 scenes) (e) Position (f) Orientation

Figure 3: Transformer Encoder attention visualizations for a varying number of training scenes (three and seven). As we train our scheme using more scenes (second row), the positional attention is able to better localize corner-like image cues (e compared to b). The orientational attention is able to better localize elongated edges (f compared to c).

(a) Kings C. (b) Old Hospital (c) Shop Facade (d) St. Mary

Figure 4: Translational Decoder attention visualization $\{\mathbf{X}_i\}_1^N$. Each activation relates to a different scene. The activations are due to an input image from the old Hospital scene. The activations of the corresponding scene are notably stronger.

Table 6: Ablations of the convolutional backbone of our model, evaluated on the 7Scenes dataset. We report the average of median position and orientation errors, across all scenes. The model choice is highlighted in bold.

Backbone	**Position [meters]**	**Orientation [degrees]**
Resnet50	0.19	8.6
EfficientNetB0	0.18	7.28
EfficientNetB1	0.17	7.26

4.4. Ablation Study

In order to study the effect of different architecture design choices, we conduct multiple ablation experiments on the 7Scenes dataset (Tables 6-9). On each experi-

Table 7: Ablations of activation maps evaluated on the 7Scenes dataset. $\mathbf{A_x}$ and $\mathbf{A_q}$ are sampled at different resolutions and passed to the respective transformer head. We report the average of median position and orientation errors, across all scenes. The model choice is highlighted in bold.

Resolution $\mathbf{A_x}/\mathbf{A_q}$	Position [meters]	Orientation [degrees]
28x28x40/14x14x112	0.22	7.47
14x14x112/28x28x40	0.18	7.28
14x14x112/14x14x112	0.19	7.78

Table 8: Ablations of the number of layers in the Encoder and Decoder components, evaluated on the 7Scenes dataset. We report the average of median position and orientation errors, across all scenes. The model choice is highlighted in bold.

Encoder/Decoder # Layers	Position [meters]	Orientation [degrees]
2	0.19	7.48
4	0.18	6.94
6	0.18	7.28
8	0.18	6.92

Table 9: Ablations of the transformer's latent dimension, C_d, evaluated on the 7Scenes dataset. We report the average of median position and orientation errors, across all scenes. The model choice is highlighted in bold.

Transformer Dimension	Position [meters]	Orientation [degrees]
64	0.18	8.06
128	0.19	7.56
256	0.18	7.28
512	0.18	7.19

ment, we start from the architecture used for our comparative analysis (Section 4.2) and modify a single algorithmic component/hyper-parameter. We compute the median position and orientation errors for each scene and report the average across scenes. Our ablation study focuses on two main aspects of our approach: (a) the derivation of activation maps (backbone and resolution) and (b) the transformer architecture.

Convolutional Backbone. We consider three convolutional encoders for our backbone choice: ResNet50, EfficientNetB0 and EfficientNetB1. The results obtained with these backbones are shown in Table 6. The two EfficientNet variants achieve a better performance compared to the ResNet50 backbone, either due to overfitting (26M parameters for ResNet50 compared to 5.3M and 7.8M for EfficientNetB0 and EfficientNetB1, respectively [32]) or a better learning capacity [32]. The best performance is achieved with the EfficientNetB1 backbone, suggesting that further improvements in accuracy can be obtained with appropriate deeper models (e.g., deeper EfficientNet models), on the expense of memory and runtime.

Resolution of Activation Maps. The EfficientNet backbone can be sampled at different endpoints. As we move along these endpoints, the receptive field and the depth of each entry grow. Thus, activation maps sampled at different levels capture different features which may vary in how informative they are for position and orientation estimation. In order to evaluate this effect, we train our model by sampling the position and orientation activation maps, A_x and A_q, at different endpoints. Specifically, we consider sampling both A_x and A_q from the same endpoint, with a resolution of $14 \times 14 \times 112$ or when segregating the sampling from two different resolutions: $14 \times 14 \times 112$ and $28 \times 28 \times 40$. Table 7 shows the results of these three combinations. The best performance is obtained when providing a combination of coarse and fine activation maps, for the position and orientation transformers, respectively.

Transformer Architecture The main hyper-parameters of our Transformer architecture follow the standard choice. Thus, we further evaluate the sensitivity of our model performance to changes in two main hyper-parameters: the number of layers in the encoder and decoder components and the transformer dimension, C_d. The results are shown in Table 8 and Table 9, respectively. All the considered variants, in terms of layer number and transformer dimension, maintain SOTA position and orientation accuracy, compared to other APR solutions (Table 4). The performance improves with the Transformer dimension, suggesting that larger models may achieve further improvement to localization accuracy. We note that regardless of the number of layers (Table 8), our model outperforms other solutions (see Table 4). We chose the standard 6-layers model for our ablations as we found it to be the most robust.

5. Conclusions

In this work we proposed a novel transformer-based approach for multi-scene absolute pose regression. Self-attention using two Transformer Encoders attends separately to the positional and orientational informative image cues. Thus, aggregating the activation maps computed by the backbone CNN in a task-adaptive way. Our formulation allows to agglomerate non-scene-specific information in the backbone CNN and Transformer Encoders. The scene-specific information is encoded by Transformer-Decoders, and is queried per scene. Our approach is shown to provide a new state-of-the-art localization accuracy for both single and multi-scene absolute regression approaches, across outdoor and indoor datasets.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[2] Vassileios Balntas, Shuda Li, and Victor Prisacariu. Relocnet: Continuous metric learning relocalisation using neural nets. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

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

[4] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother. Dsac â€" differentiable ransac for camera localization. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2492–2500, Los Alamitos, CA, USA, jul 2017. IEEE Computer Society.

[5] E. Brachmann and C. Rother. Learning less is more - 6d camera localization via 3d surface regression. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4654–4662, 2018.

[6] Eric Brachmann and Carsten Rother. Expert sample consensus applied to camera re-localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7525–7534, 2019.

[7] Samarth Brahmbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. Geometry-aware learning of maps for camera localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[8] Ming Cai, Chunhua Shen, and Ian Reid. A hybrid probabilistic model for camera relocalization. In *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*, page 238. BMVA Press, 2018.

[9] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 213–229, Cham, 2020. Springer International Publishing.

[10] Mingyu Ding, Zhe Wang, Jiankai Sun, Jianping Shi, and Ping Luo. Camnet: Coarse-to-fine retrieval for camera relocalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2020.

[12] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler. D2-net: A trainable cnn for joint description and detection of local features. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8084–8093, 2019.

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

[14] B. Glocker, S. Izadi, J. Shotton, and A. Criminisi. Real-time rgb-d camera relocalization. In *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 173–179, 2013.

[15] Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. In Danica Kragic, Antonio Bicchi, and Alessandro De Luca, editors, *2016 IEEE International Conference on Robotics and Automation, ICRA 2016, Stockholm, Sweden, May 16-21, 2016*, pages 4762–4769. IEEE, 2016.

[16] A. Kendall and R. Cipolla. Geometric loss functions for camera pose regression with deep learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6555–6564, 2017.

[17] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2938–2946, 2015.

[18] Z. Laskar, I. Melekhov, S. Kalia, and J. Kannala. Camera relocalization by computing pairwise relative poses using convolutional neural network. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 920–929, 2017.

[19] Luke Melaskyriazi. efficientnet-pytorch. `https://pypi.org/project/efficientnet-pytorch/`, 2019.

[20] Iaroslav Melekhov, Juha Ylioinas, Juho Kannala, and Esa Rahtu. Image-based localization using hourglass networks. In *2017 IEEE International Conference on Computer Vision Workshops, ICCV Workshops 2017, Venice, Italy, October 22-29, 2017*, pages 870–877. IEEE Computer Society, 2017.

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

[22] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han. Large-scale image retrieval with attentive deep local features. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3476–3485, 2017.

[23] et al. Paszke, Adam. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. Alche-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, pages 8026–8037. Curran Associates, Inc., 2019.

[24] N. Radwan, A. Valada, and W. Burgard. Vlocnet++: Deep multitask learning for semantic visual localization and odometry. *IEEE Robotics and Automation Letters*, 3(4):4407–4414, 2018.

[25] P. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12708–12717, 2019.

[26] T. Sattler, B. Leibe, and L. Kobbelt. Efficient effective prioritized matching for large-scale image-based localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1744–1756, 2017.

[27] T. Sattler, Q. Zhou, M. Pollefeys, and L. Leal-Taixé. Understanding the limitations of cnn-based absolute camera pose regression. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3297–3307, 2019.

[28] Yoli Shavit and Ron Ferens. Introduction to camera pose estimation with deep learning. In *arXiv preprint arXiv:1907.05272*, 2019.

[29] Yoli Shavit and Ron Ferens. Do we really need scene-specific pose encoders? In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 3186–3192. IEEE, 2021.

[30] H. Taira, M. Okutomi, T. Sattler, M. Cimpoi, M. Pollefeys, J. Sivic, T. Pajdla, and A. Torii. Inloc: Indoor visual localization with dense matching and view synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019.

[31] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[32] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

[33] Abhinav Valada, Noha Radwan, and Wolfram Burgard. Deep auxiliary learning for visual localization and odometry. *ICRA*, pages 6939–6946, 2018.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc., 2017.

[35] F. Walch, C. Hazirbas, L. Leal-Taixé, T. Sattler, S. Hilsenbeck, and D. Cremers. Image-based localization using lstms for structured feature correlation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 627–637, 2017.

[36] Bing Wang, Changhao Chen, Chris Xiaoxuan Lu, Peijun Zhao, Niki Trigoni, and Andrew Markham. Atloc: Attention guided camera localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10393–10401, 2020.

[37] J. Wu, L. Ma, and X. Hu. Delving deeper into convolutional neural networks for camera relocalization. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5644–5651, 2017.