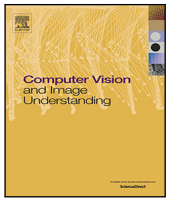
Computer Vision and Image Understanding 243 (2024) 103982

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

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



Learning single and multi-scene camera pose regression with transformer encoders

Yoli Shavit[1], Ron Ferens[1], Yosi Keller [*,1]

Faculty of Engineering, Bar Ilan University, Ramat-Gan, Israel

ARTICLE INFO

Communicated by Marius Leordeanu

Keywords:
Deep learning
Transformers
Camera pose estimation
Localization

ABSTRACT

Contemporary state-of-the-art localization methods perform feature matching against a structured scene model or learn to regress the scene 3D coordinates. The resulting matches between 2D query pixels and 3D scene coordinates are used to estimate the camera pose using PnP and RANSAC, requiring the camera intrinsics for both the query and reference images. An alternative approach is to directly regress the camera pose from the query image. Although less accurate, absolute camera pose regression does not require any additional information at inference time and is typically lightweight and fast. Recently, Transformers were proposed for learning multi-scene camera pose regression, employing encoders to attend to spatially varying deep features while using decoders to embed multiple scene queries at once. In this work, we show that Transformer Encoders can aggregate and extract task-informative latent representations for learning *both single-* and *multi-* scene camera pose regression, *without* Transformer-Decoders. Our approach is shown to reduce the runtime and memory of previous Transformer-based multi-scene solutions, while comparing favorably with contemporary pose regression schemes and achieving state-of-the-art accuracy on multiple indoor and outdoor regression benchmarks. In particular, to the best of our knowledge, our approach is the first absolute regression approach to attain sub-meter average accuracy across outdoor scenes. We make our code publicly available at: https://github.com/yolish/transposenet.

1. Introduction

Estimating the position and orientation of a camera using a captured image is an essential task in multiple contemporary computer vision applications, such as augmented reality, autonomous driving, and navigation in closed environments such as malls, museums, airports, and amusement parks, to name a few. Many of these applications require a fast and lightweight solution that can be deployed on low-end devices, such as cellphones. Recent state-of-the-art (SOTA) for visual pose estimation follows a structure-based paradigm (Taira et al., 2019; Sattler et al., 2019), where given a query image, local features that are extracted from the query and geo-referenced images are then matched. The features of the geo-referenced images are retrieved using image retrieval (IR) from a large predefined database, or matched against a database of precomputed features. The resulting 2D-2D matches are transformed into 2D-3D correspondences using depth and/or a 3D model acquired by Light Detection and Ranging (LiDAR) and Structure from Motion (SfM). Finally, the pose (or a set of poses) is estimated from 2D–3D matches using Perspective-n-Point (PnP) and RANSAC. Such approaches require the query camera's intrinsics, which might be unavailable or inaccurate. As they depend on the detection and tracking of feature points in images showing different perspectives (Sattler et al., 2017), they are best applied to urban scenes with many images and easily tracked feature points. However, relatively common scenes (Frahm et al., 2010), with repetitive patterns, such as brick facades and windows in modern buildings, as well as textureless image regions, are prone to false feature point matches, resulting in erroneous localization. Absolute regression-based approaches estimate the camera pose with a single forward pass, using the query image as input to a trained regressor. Thus, the end device has to store only the regressor parameters. Such methods are less accurate than structure-based schemes, but do not require the camera intrinsics, and do not depend on feature points detection and matching. Absolute pose regressors (APR) were first described in the seminal work of Kendall et al. (2015). The proposed architecture consisted of a convolutional backbone and a multilayer perceptron head (MLP), separately regressing position and orientation. Despite its lower accuracy compared to feature points-based schemes, the appealing 5 ms runtime and simplicity (a single component instead of a heavy pipeline) paved the way for numerous methods, intending to maintain low runtime and memory requirements, while improving the accuracy and generalization of the original method (Kendall et al., 2015; Kendall and Cipolla, 2016; Naseer and

* Corresponding author.
E-mail address: yosi.keller@biu.ac.il (Y. Keller).
[1] Equal Contribution.

https://doi.org/10.1016/j.cviu.2024.103982
Received 29 April 2023; Received in revised form 13 January 2024; Accepted 6 March 2024
Available online 15 March 2024

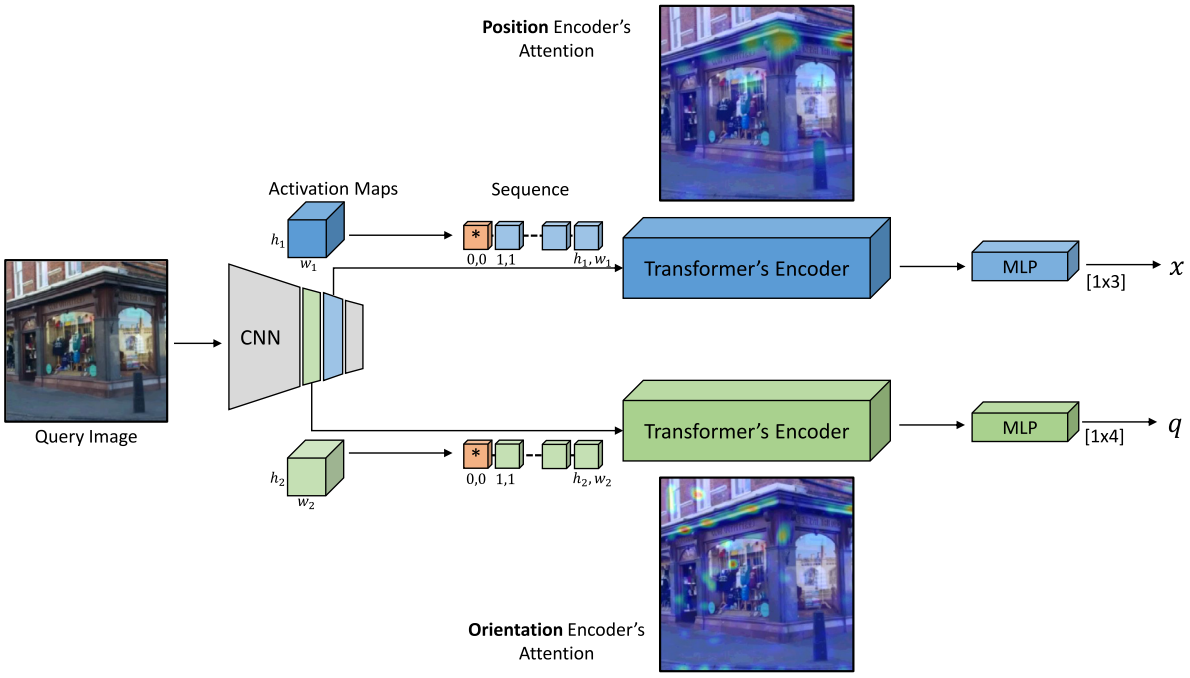


Fig. 1. The proposed attention-based regression localization scheme. The input image is first encoded by a convolutional backbone. Two activation maps, at different resolutions, are transformed into sequential representations. The two activation sequences are analyzed by dual Transformer encoders, one per regression task. We depict the attention weights via heatmaps. The position is best estimated by corner-like image features, while orientation is estimated by edge-like features. Each Transformer encoder output is used to regress the respective camera pose component (position x or orientation q).

Burgard, 2017; Walch et al., 2017; Wu et al., 2017; Shavit and Ferens, 2021; Wang et al., 2020; Shavit and Keller, 2022). Different camera pose regressors utilized different backbones (Melekhov et al., 2017; Shavit and Ferens, 2021), loss formulations (Kendall and Cipolla, 2017) and MLP architectures (Wu et al., 2017; Naseer and Burgard, 2017) as well as additional manipulations of the output (Walch et al., 2017; Wang et al., 2020). Common to all these methods is applying pose regression using a *single* global image encoding computed by the backbone CNN. Over the past few years, Transformers-Encoders (Vaswani et al., 2017), were shown to provide an efficient computational means for analyzing spatially varying image activations as sequences (Carion et al., 2020; Jiang et al., 2021; Sun et al., 2021). Recently, Transformers were proposed to learn multiscene absolute pose regression (Shavit et al., 2021) by employing Encoders to focus on pose-informative features, and Decoders for embedding multiple scenes in parallel. In this work, we present a Transformer-Encoder-based formulation for learning single- and multi-scene absolute pose regression. Instead of formulating multi-scene pose regression as a sequence-to-sequence problem, with Encoders and Decoders (Shavit et al., 2021), we propose to cast the camera pose regression problem as two *sequence-to-one* problems, where dedicated position and orientation tokens are used to adaptively aggregate task-specific visual features and a specialized scene token enables multi-scene learning. This, in turn, allows us to investigate separately learned image aggregations for position and orientation estimation for both the single- and multi-scene use cases using only Transformer-Encoders. Specifically, given a query image, intermediate activation maps serve as input to a pair of translation- and rotation-aware transformer encoders (Fig. 1). Similarly to Devlin et al. (2019) and Dosovitskiy et al. (2020), we use dedicated tokens for position-unbiased aggregation for each Encoder. The output of each token is used to regress the corresponding localization attribute. The gist of our approach is that each Transformer-Encoder captures task-specific informative image cues for either position or orientation estimation. The largest attention values of the Positional Encoder are blobs around corners, which are commonly used localization cues, as in the SIFT detector. Similarly, the Orientational Encoder emphasizes elongated edges that are informative visual cues for orientation estimation. Other image features that are less location-informative show weak attention activation. We illustrate this by overlaying the attention weights as heatmaps (Figs. 1 and 3). By adaptively detecting and weighting informative spatial activations, encoders improve localization accuracy. This single-scene formulation (shown in Fig. 1) is extended to learning multiple scenes (Fig. 2) with the addition of two fully connected layers. Instead of using dedicated Transformer Decoders as in Shavit et al. (2021), we embed the scene index with a single FC layer and add the encoded scene to the learnable position and orientation tokens. An additional FC layer is used to classify the scene before encoding it. We evaluate the single-scene and multi-scene variants of our approach on indoor and outdoor localization challenges. Our approach outperforms previous SOTA APRs, including our previous work (Shavit et al., 2021) and can run at 5 ms (per image) on a low-end GTX 1080 GPU. In summary, our contributions are as follows:

- We propose an Encoder-based formulation for camera pose regression using dual encoders with dedicated tokens, one per regression task.
- Our formulation can be extended to learn multiple scenes simultaneously with the addition of only two FC layers. As such, it offers a much simpler, faster, and lighter solution, compared to the current multi-scene SOTA (Shavit et al., 2021).
- Our approach has been shown to compare favorably with SOTA single- and multi-scene regression-based schemes, when applied to indoor and outdoor localization datasets.
- We further optimize our network's runtime and model size to showcase its applicability to real-life scenarios, in particular deployment on low-end devices, resulting in a 24.9 Mb model size and a runtime of 5 ms.

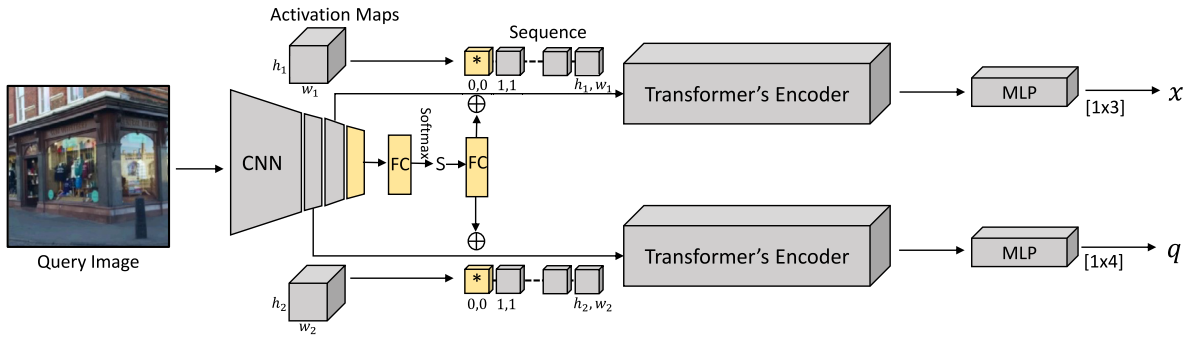


Fig. 2. Extending our proposed architecture for learning multiple scenes with a single model (highlighted in yellow). A classifier composed of a single FC layer and log Softmax is applied on the output of the convolutional backbone to predict the scene from which the image was taken. The scene index is encoded with a second FC layer and the encoding is added to the position and orientation tokens before applying the respective Encoders and MLPs to regress the camera pose. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

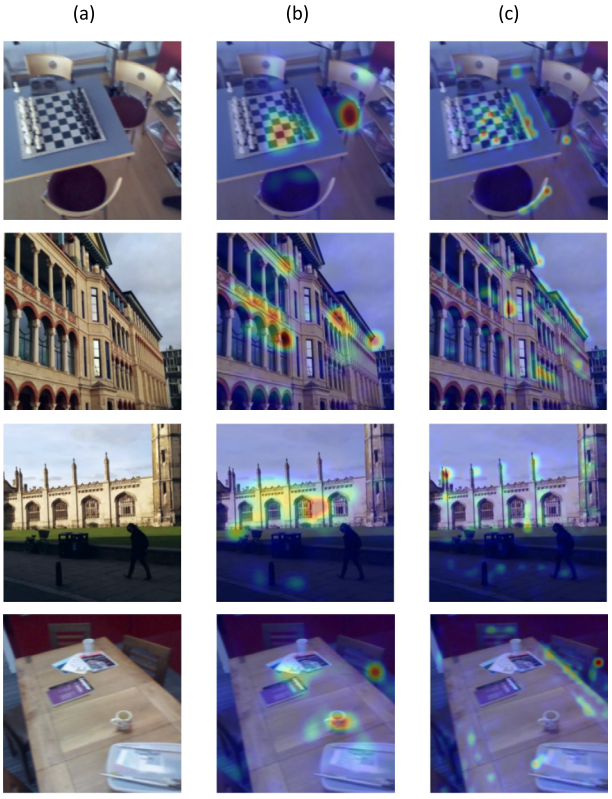


Fig. 3. Heatmap visualization of attention weights from the position and orientation encoders (columns (b) and (c)). Each row shows the query image (column (a)) along with the corresponding attention heatmaps.

2. Related work

2.1. Image-based camera pose estimation

2.1.1. Structure-based approaches

Methods in this class typically encode each query image to be localized using a CNN trained for image retrieval. A relatively small set of nearest neighbors is retrieved for image-to-image matching. Local feature matching can be applied using brute force methods (Taira et al., 2019) or using another learned component (Sarlin et al., 2020; Shi et al., 2022). To convert image matches to the world plane, spatial information needs to be available at inference time. The pose can then be estimated using PnP-RANSAC from 2D-3D correspondences. These methods provide SOTA pose estimation accuracy but require large-scale correspondence estimation, known camera intrinsics, and a structured 3D representation of the scene. They need to store the image descriptors for the retrieval step, 3D point positions, information about which 3D points are visible in which geo-referenced image, and feature descriptors for each 3D point. The scene model is commonly acquired by scanning with a depth sensor (Cavallari et al., 2017), or by using SfM (Sattler et al., 2017). Torii et al. (2018) showed that product quantization can be used to reduce memory consumption by an order of magnitude without degrading retrieval performance. Similarly, product quantization can be applied to the descriptors associated with the 3D points. Lynen et al. (2015) report that compressing the descriptors results in fewer frames being localized. They also note that this does not impact real-time on-device localization, as frame-to-frame tracking (either using visual or visual-inertial odometry) can compensate for not having global pose estimates for multiple frames. Another approach to reduce memory requirements is to use only a subset of all 3D points (Mera-Trujillo et al., 2020; Sattler et al., 2017). Sattler et al. (2017) match the image features extracted from the query image with the descriptors associated with 3D scene points. They propose a novel prioritized matching step that first considers features that are more likely to yield valid 2D-to-3D matches. A different approach is to directly regress the 3D scene coordinates from the image pixels so that the resulting 2D-3D matches can be used to estimate the pose with PnP-RANSAC. Originally proposed by Shotton et al. (2013), this approach was recently extended by Brachmann and Rother (Brachmann et al., 2017; Brachmann and Rother, 2018) by training a CNN to estimate the 3D locations corresponding to the pixels in the query image. This establishes 2D-3D correspondences that are utilized by a differentiable PnP-RANSAC to estimate the camera pose in both the training and testing phases. Such approaches achieve state-of-the-art accuracy. The PixLoc approach by Sarlin et al. (Sarlin et al., 2021) estimates the 6-DoF pose from an image and a 3D model. A CNN is used to learn robust and invariant multilevel visual features, with pixelwise confidence, for query and reference images.

2.1.2. Regression-based approaches

Camera pose regression was first suggested by Kendall et al. (2015). Following the success of CNN backbones in extracting task-informative visual features from images, the authors proposed applying them to directly regress the pose from the input image. This novel approach, albeit far less accurate than structure-based localization pipelines enables pose estimation with a single forward pass in just a few milliseconds. The paradigm of regressing camera pose using the final output of a CNN backbone was adopted by most APRs (Shavit and Ferens, 2019). Architecture variations focused on alternatives to the originally proposed CNN backbone (Melekhov et al., 2017; Naseer and Burgard, 2017; Wu et al., 2017; Shavit and Ferens, 2021) and on deeper branching architectures for the MLP head (Wu et al., 2017; Naseer and Burgard, 2017). Other works tried to address overfitting by averaging predictions from

models with randomly dropped activations (Kendall and Cipolla, 2016) or by reducing the dimensionality of global image encoding with Long-Short-Term-Memory (LSTM) layers (Walch et al., 2017). Multimodality fusion (for example, with inertial sensors) was also suggested as a means to improve accuracy (Brahmbhatt et al., 2018). Sattler et al. (2019) proposed an IR-based baseline for camera pose regression. With this baseline, pose predictions are made by taking a weighted average of poses from a set of nearest neighbors. Pose auto-encoders (Shavit and Keller, 2022) were also suggested for enhancing the performance of APRs at inference time.

Recently, attention and Transformers have been suggested for improving the accuracy of APRs. Wang et al. suggested using attention to guide the regression process (Wang et al., 2020). The authors applied dot product self-attention with SoftMax on the final output of the CNN backbone and updated it with the new attention-based representation (through summation), before regressing the pose with an MLP head. A transformer-based approach to multi-scene absolute pose regression was proposed by Shavit et al. (2021). The authors apply a full transformer to encode *multiple* scenes using a shared backbone. The scheme was shown to provide SOTA multi-scene pose accuracy compared to current APRs. In this work, we use Transformer-Encoders with dedicated regression tokens to learn single- and multi-scene camera pose regression. Our approach provides SOTA single-scene and multi-scene regression accuracy, outperforming (Shavit et al., 2021), while using a significantly simpler, smaller, and faster network.

A different, yet related, body of work estimates the absolute camera pose using the relative motion between the query image and a reference image (Balntas et al., 2018; Ding et al., 2019). The GL-Net graph neural network (GNN) was applied by Xue et al. (2020a) to multiframe absolute pose estimation, where the GNN allows the exchange of information between non-consecutive frames of a video clip. GNNs were also applied to multiframe relative localization by Turkoglu et al. (2021). Recent schemes combine image synthesis and localization (Xue et al., 2020a; Yen-Chen et al., 2021) by refining the pose regression network using differentiable rendering. Such schemes maximize the appearance similarity between the input query image and the scene image rendered using the predicted pose.

Recent works also suggested employing 2D maps to enable simultaneous localization and navigation from visual inputs (Leordeanu and Paraicu, 2021; Sarlin et al., 2023). In this line of work, the regression model predicts the 3-degrees-of-freedom pose representation (3DoF) concerning the 2D map, instead of estimating the 6DoF parameters. Leordeanu and Paraicu (2021) suggested a segmentation model, which takes a query image and predicts a dot mask, centered around the location of the image on a 2D geographic output map. This approach was also shown to be successful for positioning satellite images (Marcu et al., 2018). Map-aware pose regression can complement absolute pose regression methods and enhance their robustness in visual-based localization and navigation scenarios.

2.2. Attention and transformers in image analysis

Transformers were introduced by Vaswani et al. (2017) as a novel formulation of attention-based sequence encoding and analysis. Attention mechanisms (Bahdanau et al., 2015) are layers of neural networks that aggregate information from the entire input sequence. Transformers introduced attention layers that scan through each element of a sequence and update it by aggregating information from the whole sequence. Attention-based approaches have been shown to outperform recurrent neural networks (RNN) in encoding long sequences, and have been applied in multiple recent works in natural language processing (NLP) and computer vision (Devlin et al., 2019). In particular, in the context of image analysis, attention is related to the bilinear pooling approach of Zhang et al. (2016), and its multiple extensions (Gao et al., 2016). These schemes compute the inner products between all CNN activations and have been shown to significantly improve fine-grained recognition. Bilinear pooling and attention schemes allow implementing 'needle-in-a-haystack' approaches, where each entry in the CNN activation map is adaptively weighted. Thus, it allows us to numerically emphasize the contribution of the task-informative image locations, in contrast to the visual clutter. For instance, consider the attention weights visualization in Fig. 1, where we highlight the position and orientation of informative image locations. In this work, we utilize *self-attention,* where the inner products are computed between the different entries of each particular activation map. This is implemented by a *Transformer-Encoder*, in contrast to using a *Transformer-Decoder* where the inner products between the input data and an additional set of vectors, denoted as *queries*, are computed. Transformer-based attention was successfully applied to image feature matching (Jiang et al., 2021; Sun et al., 2021). Instead of applying image feature detection, description, and matching sequentially, the matches are computed using the Transformers' attention layers, in contrast to previous state-of-the-art matching schemes that use 4D correlation volumes (Teed and Deng, 2020).

3. Transformer-encoder for pose regression

In this work, we propose to localize a camera by estimating $\mathbf{p} = \langle \mathbf{x}, \mathbf{q} \rangle$, where $\mathbf{x} \in \mathbb{R}^3$ is the position of the camera in the world and $\mathbf{q} \in \mathbb{R}^4$ is the quaternion encoding its 3D orientation. Thus, the problem of camera pose regression is to jointly learn two regression tasks: regressing the camera position $\mathbf{x}$ and regressing the camera orientation $\mathbf{q}$. We model each task as a sequence-to-one problem, where the input is a sequential representation of a learned activation map and the output is an estimate of the corresponding pose component (either $\mathbf{x}$ or $\mathbf{q}$). Following the success of Transformers in learning multiscene pose regression (Shavit et al., 2021), we propose to use Transformer-Encoders for aggregating sequential representations into a single latent vector. Specifically, we employ position and orientation encoders with dedicated regression tokens that learn to emphasize informative visual cues for position and orientation inference, respectively. An MLP can then regress the resulting encoder output, replacing the classifier head used in standard sequence-to-one architectures (Devlin et al., 2019). An overview of the proposed scheme denoted *TransPoseNet* , is shown in Fig. 1, where the image is initially processed by a CNN backbone to compute activation maps that are analyzed by the dual encoders and the corresponding regression heads.

3.1. Network architecture

TransPoseNet is composed of a shared convolutional backbone and two branches for separately regressing the position and orientation of the camera. Each branch consists of a separate Transformer-Encoder and an MLP head. Using separate activation maps and Transformers allow us to attend to the different features at different resolutions, depending on the particular learned task.

Convolutional Backbone. The activation maps of the convolutional backbones capture latent visual features at different resolutions. Similarly to Shavit et al. (2021), given an image $\mathbf{I} \in \mathbb{R}^{H \times W \times C}$, we sample the backbone at two different resolutions and take an activation map per regression task: $M_{\mathbf{x}}$ and $M_{\mathbf{q}}$ respectively.

Sequential Representations of Activation Maps. We follow Carion et al. (2020) and Shavit et al. (2021) and convert an activation map $\mathbf{M} \in \mathbb{R}^{H_m \times W_m \times C_m}$ into a sequence $\hat{\mathbf{M}} \in \mathbb{R}^{H_m \cdot W_m \times C_t}$ using a 1×1 convolution (projecting to dimension C_t) followed by flattening. We further append a learned *orientation/position token* $\mathbf{t} \in \mathbb{R}^{C_t}$ to the sequence of map features, analogous to the *class token* in sequence-to-one NLP classification Transformer architectures (Devlin et al., 2019). Hence, the input to the encoder is given by:

$$E_{in} = [\mathbf{t}, \hat{\mathbf{M}}] \in \mathbb{R}^{(H_m \cdot W_m + 1) \times C_t}. \tag{1}$$

Positional Encoding. To preserve the spatial information of each location in the map and assign a particular positional encoding to the

token $\mathbf{t}$, the spatial positions are encoded using positional encoding as in Carion et al. (2020) and Shavit et al. (2021). We train two one-dimensional encodings for the X and Y axes separately, to reduce the number of learned positional parameters. Thus, for an activation map $\mathbf{M}$ we define the sets of trainable positional embedding vectors $\mathbf{E}_x \in \mathbb{R}^{(W_m+1)\times C_t/2}$ and $\mathbf{E}_y \in \mathbb{R}^{(H_m+1)\times C_t/2}$, such that a spatial position (i, j), $i \in 1 \dots H_m$, $j \in 1 \dots W_m$, is encoded by the concatenation of the corresponding embedding vector:

$$\mathbf{E}^{i,j}_{pos} = \begin{bmatrix} \mathbf{E}^j_x \\ \mathbf{E}^i_y \end{bmatrix} \in \mathbb{R}^{C_t}. \tag{2}$$

The pose token is assigned to the position $(0, 0)$ and the embedding vectors $(\mathbf{E}^0_x, \mathbf{E}^0_y)$, as these vectors will only be associated with that token. Similar to the activation map $\mathbf{M}$, the positional embeddings are reshaped as a sequence $\widehat{\mathbf{E}} \in \mathbb{R}^{(H_m \cdot W_m+1)\times C_t}$ and passed to the encoder and are learnt during the training process.

Transformer-Encoder. We use a standard Transformer-Encoder architecture (Devlin et al., 2019) consisting of n identical blocks. Each block contains a multi-head self-attention (MHA) module and an MLP with two layers and gelu nonlinearity. The input is passed through a LayerNorm (Vaswani et al., 2017) before each module (MHA/MLP) and added back to the output with residual connections Vaswani et al. (2017) and dropout. As in Carion et al. (2020) we add the positional encoding to the input before each layer and normalize the output of the final layer with an additional LayerNorm layer. The encoder output, $\mathbf{t}' \in \mathbb{R}^{C_t}$ at the position of the special token t, represents a global and context-sensitive summarization of the local features of the input activation map.

MLP Head. An MLP head with one hidden layer and gelu non-linearity regresses the target vector ($\mathbf{x}$ or $\mathbf{q}$) from $\mathbf{t}'$.

3.2. Camera pose loss

Camera pose regressors are optimized to minimize the deviation between the ground truth pose $\mathbf{p}_0 = \langle \mathbf{x}_0, \mathbf{q}_0 \rangle$ and the predicted pose $\mathbf{p} = \langle \mathbf{x}, \mathbf{q} \rangle$. The position loss $L_\mathbf{x}$ and the orientation loss $L_\mathbf{q}$ are the Euclidean distance between the ground truths and their estimates $L_\mathbf{x} = \|\mathbf{x}_0 - \mathbf{x}\|_2$ and $L_\mathbf{q} = \left\|\mathbf{q_0} - \frac{\mathbf{q}}{\|\mathbf{q}\|}\right\|_2$, where $\mathbf{q}$ is normalized to a unit vector to map to a valid rotation matrix. To balance between the two losses, while considering their scale differences, Kendall and Cipolla (2017) suggested learning their relative weight by modeling the uncertainty of each task:

$$L_\mathbf{p} = L_\mathbf{x} \exp(-s_\mathbf{x}) + s_\mathbf{x} + L_\mathbf{q} \exp(-s_\mathbf{q}) + s_\mathbf{q}, \tag{3}$$

where $s_\mathbf{x}$ and $s_\mathbf{q}$ are learned parameters. Recently, Shavit and Ferens (2021) showed the advantage of learning each task on its own. Here, we follow a combined approach, where we first train the entire model to minimize Eq. (3) and then fine-tune each MLP head solely with its respective loss.

3.3. Extension to multi-scene learning

Our model can be further extended to learn *multiple* scenes at once with a simple architectural modification (Fig. 2). We predict the scene identifier s by applying an FC layer and log SoftMax on the output of the convolutional backbone and then embed the index using a second FC layer. The resulting scene encoding, $s_e \in R^{C_t}$, is added to the position and orientation tokens before applying the respective Encoders and MLPs for regressing the camera pose. To optimize both pose regression and scene classification, we minimize the loss in Eq. (3) and the Negative Log-Likelihood (NLL) loss:

$$L_{\mathbf{p},\mathbf{s}} = L_p + NLL(s, s_0) \tag{4}$$

where s_0 is the ground truth scene index.

3.4. Implementation details

Our localization approach is implemented using EfficientNet-B0 (Tan and Le, 2019) as a convolutional backbone, previously shown to focus on informative visual features for image classification. The EfficientNet backbone has several reduction levels, each corresponding to activation maps with decreasing resolution and increasing depth. We use an open source implementation of EfficientNet (Melaskyriazi, 2019) and use the activation maps from two endpoints (reduction levels): $m_{rdct4} \in \mathbb{R}^{14\times14\times112}$ and $m_{rdct3} \in \mathbb{R}^{28\times28\times40}$. As in Shavit et al. (2021), we use m_{rdct4} and m_{rdct3} as inputs for the position and orientation branches, respectively. We linearly project each activation map to a common depth dimension $C_t = 256$ and learn the positional encoding of the same depth. For the Transformer-Encoder, we use a standard implementation with six blocks and a dropout of $p = 0.1$. Each block contains an MHA layer with four heads and an MLP preserving the input dimension C_t. The hidden layer of the MLP regressor heads expands the dimension from 256 to 1024 before regressing the output vector to a fully connected layer.

4. Experimental results

The proposed scheme focuses on improving the precision of the absolute pose regression (APR). As such, we evaluate it on multiple contemporary datasets used for benchmarking camera pose estimation, and compare the results to recent state-of-the-art localization methods. We compare multiple types of localization to provide a comprehensive overview, not only for regression-based camera pose estimation.

Datasets. The Cambridge Landmarks (Kendall et al., 2015) data set represents an urban environment and represents outdoor localization tasks with a spatial extent of $\sim$ 900–5500 m^2. We use four of its six scenes for comparative evaluation, as the other two scenes are not commonly benchmarked. The 7Scenes (Glocker et al., 2013) dataset consists of seven small-scale indoor scenes, depicting a spatial extent of $\sim$ 1–10 m^2. Both datasets present various localization challenges, such as occlusions, reflections, motion blur, lighting conditions, repetitive textures, and variations in viewpoints and trajectory.

Baseline Localization Comparisons. We experimentally compare our approach with contemporary localization schemes, divided into **six** algorithmic classes, following the comprehensive evaluation of Turkoglu et al. (2021). The different classes present different localization approaches and algorithmic trade-offs: feature vs. regression-based, absolute vs. relative, single vs. multiple image-based: (i) The structure-based DSAC* (Brachmann and Rother, 2021) approach that is the current accuracy SOTA. (ii) Sequence-based Absolute Pose Regression (APR), (iii) Image Retrieval (IR) and (iv) Relative Pose Regression (RPR). Our approach relates to the last two classes, consisting of single-scene (1S) multi-scene (MS) APR schemes.

Training. We optimize our model using Adam, with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-10}$ and a batch size of 8. In the first step of training, we minimize the loss defined in Eq. (3), whose parameters are initialized as in Valada et al. (2018). We use a learning rate of $\lambda = 10^{-4}$ and decrease it by a factor of 10 every 100 (200) epochs for indoor (outdoor) localization for up to 300 (600) epochs. We apply a weight decay of 10^{-4} and train the encoders with a dropout of $p = 0.1$. At the second training stage, we fine-tune each MLP head separately with its respective loss (Eq. (3)), while freezing all other weights in the network. This allows us to achieve better performance without having to trade off the two objectives. Since the position is dominant in localizing large-scale scenes, we fine-tune the orientation head with a latent position before the position Transformer-Encoder (changing the orientation MLP to take a concatenated input). For indoor localization, we train both heads independently and use a higher learning rate and a higher weight decay (10^{-3} and 10^{-2}, respectively). The training of multi-scene models is done as in Shavit et al. (2021). To allow our model to better generalize, we follow the augmentation procedure used

Table 1
Comparison to state-of-the-art methods: Cambridge Landmarks.

		College	Hospital	Shops	St. Mary	Avg.
	DSAC (Brachmann and Rother, 2021)	0.18, 0.3°	0.21, 0.4°	0.05, 0.3°	0.15, 0.5°	0.15, 0.4°
Seq.	MapNet (Brahmbhatt et al., 2018)	1.08, 1.9°	1.94, 3.9°	1.49, 4.2°	2.00, 4.5°	1.63, 3.6°
	GL-Net (Xue et al., 2020b)	0.59, 0.7°	1.88, 2.8°	0.50, 2.9°	1.90, 3.3°	1.22, 2.4°
IR	VLAD (Torii et al., 2015)	2.80, 5.7°	4.01, 7.1°	1.11, 7.6°	2.31, 8.0°	2.56, 7.1°
	VLAD+Inter (Sattler et al., 2019)	1.48, 4.5°	2.68, 4.6°	0.90, 4.3°	1.62, 6.1°	1.67, 4.9°
RPR	EssNet (Zhou et al., 2020)	0.76, 1.9°	1.39, 2.8°	0.84, 4.3°	1.32, 4.7°	1.08, 3.4°
	NC-EssNet (Zhou et al., 2020)	0.61, 1.6°	0.95, 2.7°	0.7, 3.4°	1.12, 3.6°	0.85, 2.8°
	RelocGNN (Brachmann and Rother, 2021)	0.48, 1.0°	1.14, 2.5°	0.48, 2.5°	1.52, 3.2°	0.91, 2.3°
1S APR	PoseNet (Kendall et al., 2015)	1.92, 5.40°	2.31, 5.38°	1.46, 8.08°	2.65, 8.48°	2.08, 6.83°
	BayesianPN (Kendall and Cipolla, 2016)	1.74, 4.06°	2.57, 5.14°	1.25, 7.54°	2.11, 8.38°	1.91, 6.28°
	LSTM-PN (Walch et al., 2017)	0.99, 3.65°	1.51, 4.29°	1.18, 7.44°	1.52, 6.68°	1.30, 5.57°
	SVS-Pose (Naseer and Burgard, 2017)	1.06, 2.81°	1.50, 4.03°	0.63, 5.73°	2.11, 8.11°	1.32, 5.17°
	GPoseNet (Cai et al., 2018)	1.61, 2.29°	2.62, 3.89°	1.14, 5.73°	2.93, 6.46°	2.07, 4.59°
	PoseNetLearn (Kendall and Cipolla, 2017)	0.99, 1.06°	2.17, 2.94°	1.05, 3.97°	1.49, 3.43°	1.42, 2.85°
	GeoPoseNet (Kendall and Cipolla, 2017)	0.88, **1.04°**	3.20, 3.29°	0.88, 3.78°	1.57, **3.32°**	1.63, 2.86°
	MapNet (Brahmbhatt et al., 2018)	1.07, 1.89°	1.94, 3.91°	1.49, 4.22°	2.00, 4.53°	1.62, 3.64°
	IRPNet (Shavit and Ferens, 2021)	1.18, 2.19°	1.87, 3.38°	0.72, 3.47°	1.87, 4.94°	1.41, 3.50°
	MS-Trans-1S (Shavit et al., 2021)	0.72, 2.55°	2.07, 3.24°	0.68, 3.66°	1.10, 5.26°	1.14, 3.68°
	Ours-1S	**0.60**, 2.43°	**1.45**, 3.08°	**0.55**, 3.49°	**1.09**, 4.99°	**0.92**, 3.50°
MS APR	MSPN (Blanton et al., 2020)	1.73, 3.65°	2.55, 4.05°	2.92, 7.49°	2.67, 6.18°	2.47, 5.34°
	MS-Trans (Shavit et al., 2021)	0.83, 1.47°	1.81, **2.39°**	0.86, **3.07°**	1.62, 3.99°	1.28, 2.73°
	Ours-MS	**0.80**, 2.86°	1.84, 3.82 °	0.80, 3.66°	**1.17**, 4.10°	**1.15**, 3.41°

We report the median position/orientation error in meters/degrees for each method. The best results for single-scene APR (1S APR) and multi-scene APR (MS APR) approaches are highlighted in bold.

by Kendall et al. (2015). At train time, we rescale the image such that its smaller edge is resized to 256 pixels and extract a random 224×224 crop. We then randomly jitter the brightness, contrast, and saturation. At the test time, we take the center crop after rescaling with no further augmentations. All models were trained on a single NVIDIA Tesla V100 GPU using the PyTorch framework.

4.1. Comparative analysis of camera pose regressors

Outdoor Localization. We report the median position and orientation errors for four outdoor scenes from the Cambridge Landmarks dataset (Table 1). We also include the results of structure-based methods, RPR, IR, and sequence-based approaches for reference. In general, DSAC (Brachmann and Rother, 2021) provides SOTA accuracy. When considering the class of single-scene APRs (1S APR), our proposed scheme yields the lowest positional error across all scenes and provides the lowest orientation error when evaluated on the Shop Facade scene. It also achieves the highest number of SOTA results across the dataset, among such schemes. To compare with our previous results on multi-scene localization (Shavit et al., 2021), we trained the MS-Transformer model (Shavit et al., 2021) using a single scene at a time (denoted as MS-Trans-1S). The single-scene results are summarized in Table 3, where our method is the only regressor that achieves sub-meter accuracy. It ranks first in terms of average position error and third in orientation. In addition, when considering the total rank (the average of the position and orientation ranks), our model again ranks first among all single-scene APR schemes. When considering multi-scene methods (MS APR), our multi-scene variant achieves comparable results with the current SOTA, ranking first and second in terms of position and orientation accuracy, respectively. We also consider an IR baseline, recently shown to present a lower error bound on the performance of current pose regressors (Sattler et al., 2019). Table 4 reports the percentage of scenes that achieve lower position and orientation errors, compared to this baseline. Our model and MS-Transformer are the only absolute pose regression methods which are consistently below the IR baseline's error bound, for both single and multi-scene variants.

Indoor Localization. Table 2 shows the position and orientation errors of our model and other reference localization methods, when evaluated on the 7Scenes dataset. Similarly to outdoor localization, DSAC (Brachmann and Rother, 2021) provides overall SOTA accuracy. When considering 1S-APR approaches, our proposed method achieves the best positional error in six out of the seven scenes and the highest number of state-of-the-art results overall. It ranks favorably in terms of position and orientation errors, achieving the first and second places, respectively, thus, ranking first in total in its category (Table 3). Another attention-based method (Wang et al., 2020) also achieves the first place in the total ranking but with fewer SOTA results (two versus seven achieved by our single-scene approach). Furthermore, compared to the IR baseline (Table 4), our model achieves results comparable to SOTA single-scene APR and MS-Transformer.

Run-time and model size. We report the runtime and memory footprint of multiple localization schemes in Table 5. MS-Transformer and our proposed scheme were implemented using the EfficientNet-B0 backbone and have six layer encoders (and decoders for MS-Transformer). For MS-Transformer we consider a model encoding four scenes. For our model, we consider the single-scene variant, as the additional required memory and runtime are negligible for the multi-scene model. The runtime of Duong et al. (2018), Brachmann et al. (2016) and DSAC (Brachmann et al., 2017) are cited from Duong et al. (2018), while the DSAC* timing is cited from Brachmann and Rother (2021). All timing measurements were made using GTX1080 and TeslaK80 GPUs that are of similar processing speed. We further applied NVIDIA's TensorRT[2] (TRT) to optimize our network's runtime and model size, using TRT 8.0.3.4 with CUDANN 8.2 and default settings. The optimized model has a size of 24.9 Mb (58% reduction) and an average runtime of 5 ms in deployment mode (×5 speedup), We note that this optimization had a negligible effect on performance, resulting in slightly reduced position accuracy (+1 cm error) and a slightly improved orientation estimation (−1 degree).

Our proposed *unoptimized* scheme is 50% faster than MS-Transformer (Shavit et al., 2021), and twice as fast as the accelerated DSAC* (Brachmann and Rother, 2021). However, it is three times slower than PoseNet (Kendall et al., 2015). Thus, our model extends the timing-accuracy trade-off, ranging from PoseNet (very fast and less accurate) to MS-Transformer (mid-range accuracy and timing) and up to DSAC which is SOTA accurate. We note that DSAC can be accelerated similarly to our model (as demonstrated in the DSAC* variant). As for

[2] https://developer.nvidia.com/tensorrt

Table 2
Comparison with state-of-the-art methods: 7Scenes.

		Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs
	DSAC* (Brachmann and Rother, 2021)	0.02, 1.1°	0.02, 1.0°	0.01, 1.8°	0.03, 1.2°	0.04, 1.4°	0.03, 1.7°	0.04, 1.4°
Seq.	LsG (Xue et al., 2019)	0.09, 3.3°	0.26, 10.9°	0.17, 12.7°	0.18, 5.5°	0.20, 3.7°	0.23, 4.9°	0.23, 11.3°
	MapNet (Brahmbhatt et al., 2018)	0.08, 3.3°	0.27, 11.7°	0.18, 13.3°	0.17, 5.2°	0.22, 4.0°	0.23, 4.9°	0.30, 12.1°
	GL-Net (Xue et al., 2020b)	0.08, 2.8°	0.26, 8.9°	0.17, 11.4°	0.18, 13.3°	0.15, 2.8°	0.25, 4.5°	0.23, 8.8°
IR	VLAD (Torii et al., 2015)	0.21, 12.5°	0.33, 13.8°	0.15, 14.9°	0.28, 11.2°	0.31, 11.2°	0.30, 11.3°	0.25, 12.3°
	VLAD+Inter (Sattler et al., 2019)	0.18, 10.0°	0.33, 12.4°	0.14, 14.3°	0.25, 10.1°	0.26, 9.4°	0.27, 11.1°	0.24, 14.7°
RPR	NN-Net (Laskar et al., 2017)	0.13, 6.5°	0.26, 12.7°	0.14, 12.3°	0.21, 7.4°	0.24, 6.4°	0.24, 8.0°	0.27, 11.8°
	RelocNet (Balntas et al., 2018)	0.12, 4.1°	0.26, 10.4°	0.14, 10.5°	0.18, 5.3°	0.26, 4.2°	0.23, 5.1°	0.28, 7.5°
	EssNet (Zhou et al., 2020)	0.13, 5.1°	0.27, 10.1°	0.15, 9.9°	0.21, 6.9°	0.22, 6.1°	0.23, 6.9°	0.32, 11.2°
	NC-EssNet (Zhou et al., 2020)	0.12, 5.6°	0.26, 9.6°	0.14, 10.7 °	0.20, 6.7°	0.22, 5.7°	0.22, 6.3°	0.31, 7.9°
	RelocGNN (Brachmann and Rother, 2021)	0.08, 2.7°	0.21, 7.5°	0.13, 8.70°	0.15, 4.1°	0.15, 3.5°	0.19, 3.7°	0.22, 6.5°
1S APR	PoseNet (Kendall et al., 2015)	0.32, 8.12°	0.47, 14.4°	0.29, 12.0°	0.48, 7.68°	0.47, 8.42°	0.59, 8.64°	0.47, 13.8°
	BayesianPN (Kendall and Cipolla, 2016)	0.37, 7.24°	0.43, 13.7°	0.31, 12.0°	0.48, 8.04°	0.61, 7.08°	0.58, 7.54°	0.48, 13.1°
	LSTM-PN (Walch et al., 2017)	0.24, 5.77°	0.34, 11.9°	0.21, 13.7°	0.30, 8.08°	0.33, 7.00°	0.37, 8.83°	0.40, 13.7°
	GPoseNet (Cai et al., 2018)	0.20, 7.11°	0.38, 12.3°	0.21, 13.8°	0.28, 8.83°	0.37, 6.94°	0.35, 8.15°	0.37, 12.5°
	PoseNetLearn (Kendall and Cipolla, 2017)	0.14, 4.50°	0.27, 11.8°	0.18, 12.1°	0.20, 5.77°	0.25, 4.82°	0.24, 5.52°	0.37, 10.6°
	GeoPoseNet (Kendall and Cipolla, 2017)	0.13, 4.48°	0.27, 11.3°	0.17, 13.0°	0.19, 5.55°	0.26, 4.75°	0.23, 5.35°	0.35, 12.4°
	MapNet (Brahmbhatt et al., 2018)	**0.08, 3.25°**	0.27, 11.7°	0.18, 13.3°	**0.17, 5.15°**	0.22, **4.02** °	0.23, **4.93°**	0.30, 12.1°
	IRPNet (Shavit and Ferens, 2021)	0.13, 5.64°	0.25, 9.67°	0.15, 13.1°	0.24, 6.33°	0.22, 5.78°	0.30, 7.29°	0.34, 11.6°
	AttLoc (Wang et al., 2020)	0.10, 4.07°	0.25, 11.4°	0.16, **11.8°**	**0.17**, 5.34°	0.21, 4.37°	0.23, 5.42°	**0.26**, 10.5°
	MS-Trans-1S (Shavit et al., 2021)	0.10, 5.86°	**0.23**, 11.0°	0.15, 12.8°	**0.17**, 6.56°	0.18, 5.32°	**0.17**, 6.30°	**0.26**, 11.3°
	Ours-1S	**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°**
MS APR	MSPN (Blanton et al., 2020)	**0.09**, 4.76°	0.29, 10.5°	0.16, 13.1°	**0.16**, 6.80°	0.19, 5.50°	0.21, 6.61°	0.31, 11.63°
	MS-Trans (Shavit et al., 2021)	0.11, **4.66°**	0.24, 9.60°	0.14, 12.2°	0.17, **5.66°**	0.18, **4.44°**	**0.17**, **5.94°**	0.26, 8.45°
	Ours-MS	0.11, 5.00°	**0.24**, **9.45°**	**0.13**, **11.8°**	0.18, 5.92°	0.19, 4.62°	**0.17**, 5.97°	**0.26**, **7.92°**

We report the median position/orientation error in meters/degrees for each method. The best results for single-scene APR (1S APR) and multi-scene APR (MS APR) approaches are highlighted in bold.

Table 3
Ranking results for the Cambridge Landmarks and 7Scenes datasets.

Method	Cambridge landmarks dataset			7Scenes dataset		
	Average [m/deg]	Ranks	Final rank	Average [m/deg]	Ranks	Final rank
PoseNet	2.09, 6.84°	10/10	10	0.44, 10.4°	9/10	10
BayesianPN	1.92, 6.28°	8/9	9	0.47, 9.81°	10/7	9
LSTM-PN	1.30, 5.52°	2/8	5	0.31, 9.86°	7/8	7
SVS-Pose	1.33, 5.17°	4/8	6	NA	NA	NA
GPoseNet	2.08, 4.59°	10/7	9	0.24, 7.87°	6/4	6
PoseNetLearn	1.43, 2.85°	6/**1**	3	0.23, 8.12°	4/5	4
GeoPoseNet	1.63, 2.86°	7/2	4	0.21, 7.78°	3/2	3
MapNet	1.63, 3.64°	7/5	8	0.23, 8.49°	4/7	6
IRPNet	1.42, 3.45°	5/3	4	0.20, 7.56°	2/**1**	**1**
MS-Trans1S	1.14, 3.68°	2/6	3	**0.18**, 8.45°	1/6	3
Ours-1S	**0.91**, 3.47°	**1**/4	**1**	**0.18**, 7.78°	**1**/2	**1**
MSPN	2.67,6.18°	3/3	3	0.20,8.41°	3/3	3
MS-Trans	1.28, **2.73°**	2/**1**	**1**	**0.18**, 7.28°	**1**/2	2
Ours-MS	**1.15**, 3.41°	**1**/2	**1**	**1.18, 7.21°**	**1/1**	**1**

We report the average position/orientation errors in meters/degrees and the respective rankings for APR approaches, separately for single- and multi-scene approaches (1S and MS APR classes, respectively). The best results are highlighted in bold. The final ranking is set according to the average rank.

the memory footprint, MS-Transformer is 73% larger than our *unoptimized* model, that is 30% larger than DSAC* (Brachmann and Rother, 2021) (43 Mb vs. 30 Mb). The model size and runtime of our TRT-based optimized network, along with its SOTA APR performance, suggest that our proposed scheme can be further optimized and deployed in real-time applications on low-end devices.

4.2. Paying attention to local features

A key element in our proposed method is the use of an encoder per task (position and orientation estimation) to process the activation maps. The motivation behind this design is to allow each encoder to attend to different features in the image. We show that the network will focus on features such as corners and blobs to estimate the relative translation, and on edges and lines to estimate the relative rotation. Fig. 3 shows the attention heatmaps for three query images taken from different scenes in the Cambridge Landmarks and the 7Scenes datasets. Each row depicts the query image and the upsampled attention weights taken from the last layers of the position and orientation encoders (second and third column, respectively). The position encoder tends to assign higher weights to features that are intuitively more distinctive for coarser localization and position estimation. In contrast, the orientation encoder focuses on lines and long edges that are informative for estimating rotational motion. For instance, consider the triplet in the third row; the position encoder focuses on the windows at the image's center of mass, while the orientation encoder attends to the elongated spires. The observed differences between the two encoders reinforce our dual-head model architecture. We further evaluated the contribution of attention-based and (our) dual-encoder-based architectures by comparing the performance of a PoseNet architecture with and without stacked MHA layers against our model, all using the same CNN backbone (EfficientNet-B0). Table 6 shows this comparison evaluated using the Kings College scene. Using the attention to jointly learn both tasks (a single encoder of stacked MHA layers) achieves better

Table 4
Comparison of APR methods to the IR baseline of Sattler et al. (2019).

	Cambridge Landmarks		7Scenes	
Method	$< err_x$	$< err_q$	$< err_x$	$< err_q$
PoseNet	25%	0%	0%	85.7%
BayesianPN	25%	25%	0%	85.7%
LSTM-PN	75%	50%	0%	100%
SVS-Pose	75%	50%	NA	NA
GPoseNet	25%	50%	0%	100%
PoseNetLearn	100%	75%	85.7%	100%
GeoPoseNet	75%	100%	85.7%	100%
MapNet	50%	100%	85.7%	100%
IRPNet	75%	100%	85.7%	100%
MS-Trans1S	100%	100%	85.7%	100%
Ours-1S	100%	100%	85.7%	100%
MSPN	50%	75%	85.7%	100%
MS-Trans	100%	100%	85.7%	100%
Ours-MS	100%	100%	85.7%	100%

For each method, we report the percentage of scenes in the Cambridge Landmarks and 7Scenes datasets, for which the error was smaller than the respective position error ($< err_x$) and orientation error ($< err_q$) of the baseline.

Table 5
Runtime and memory footprint comparison. The GeForce 1080 and Tesla K80 GPUs are of similar computing power.

Network	Time [ms]	Mem [Mb]	Device
DSAC*	30	30	2080TI
DSAC*	75	30	K80
DSAC	1500	–	1080
Brachmann	1000	–	1080
Duong	50	–	1080
PoseNet	7.8	26.7	1080
MS-Transformer	35	74.6	1080
Ours	22.5	42.9	1080
Ours-TRT	5.0	24.9	1080

Table 6
Evaluating the contribution of attention and dual-encoder architecture.

Model	Position [meters]	Orientation [degrees]
PoseNet	0.94	2.90
PoseNet+MHA	0.85	2.65
Ours	**0.60**	**2.43**

PoseNet+MHA is implemented with a single encoder. We report the median position and orientation errors using the King's College scene.

Table 7
Ablation study of CNN backbones evaluated on the Shop Facade scene. We report the median position and orientation errors. Both EfficientNet-based CNNs outperform the Resnet50 CNN. Using the EfficientNetB1 backbone degrades the accuracy due to overfitting.

Backbone	Position [meters]	Orientation [degrees]
Resnet50	0.81	4.76
EfficientNetB0	**0.65**	**3.63**
EfficientNetB1	0.69	4.03

results compared to the standard PoseNet architecture. When attending separately to position and orientation with a dual encoder architecture (ours), the respective errors decrease by 29.4% and 8.3%, respectively.

4.3. Ablation study

We study the importance and impact of four aspects of our model design: (1) backbone choice (2) the attention mechanism, (3) resolution and depth of the activation maps and (4) scene classification. In each evaluation, we start from the network used in Section 4.1 and modified

Table 8
Ablations of the Transformer-Encoder configuration evaluated on the Shop Facade scene. We report the median position and orientation errors. Using six layers optimizes the accuracy and avoids overfitting.

#Encoder layers	Position [meters]	Orientation [degrees]
2	0.92	4.61
4	0.93	4.2
6	**0.65**	**3.63**
8	0.89	3.75

Table 9
Ablations of model's embedding dimensionality, evaluated on the Shop Facade scene. We report the median position and orientation errors.

Encoder dimension	Position [meters]	Orientation [degrees]
64	0.79	4.35
128	0.81	3.99
256	**0.65**	**3.63**
512	0.68	4.18

Table 10
Ablations of activation maps evaluated on the Shop Facade scene. We evaluated different combinations of $m_{rdct3} \in \mathbb{R}^{28\times28\times40}$ and $m_{rdct4} \in \mathbb{R}^{14\times14\times112}$ of the encoder heads and report the median position and orientation errors.

Reduction position/orientation	Position [meters]	Orientation [degrees]
m_{rdct3}/m_{rdct3}	1.08	3.60
m_{rdct3}/m_{rdct4}	0.99	4.32
$\mathbf{m_{rdct4}/m_{rdct3}}$	**0.65**	**3.63**
m_{rdct4}/m_{rdct4}	0.81	4.05

a *single* algorithmic component or hyperparameter. For Ablations 1–3, all experiments were performed using the shop facade scene with our single-scene scheme. For scene classification ablation, we considered the multiscene variants of the Cambridge Landmarks dataset.

Backbone. We considered the Resnet50, EfficientNet-B0 and EfficientNet-B1 CNNs to study the choice of the backbone, where both EfficientNet variants use significantly fewer parameters than Resnet50. In all experiments, we report the results before fine-tuning the orientation branch. Table 7 indicates that both EfficientNet variants significantly outperform the Resnet50 CNN. Following our simulations, it is unclear whether this is due to reduced overfitting or improved architecture. The smaller EfficientNetB0 backbone provides the best results.

The Role of Attention and Transformers. To study the parameters of the Transformer-Encoders, we trained our scheme with a varying number of Transformer-Encoder layers. The more layers are used, the better the learning capacity. But, an over-parameterized Transformer-Encoder might lead to overfitting. The results reported in Table 8 show a 'sweet spot' when using six layers. We also varied the Transformer-Encoders' inner embedding dimension to verify our choice. The results in Table 9 show that, similarly to varying the number of layers, there exists a 'sweet spot' using $\mathbb{R}^{256}$ as the encoder's latent dimension.

Localization using Coarse and Fine Activation Maps. Activation maps from different endpoints of a CNN backbone capture different features in the input image. Deeper levels of reduction cover growing areas in the image with increased depth. As orientation and position estimation may require different feature complexity, we train our model with different combinations of $m_{rdct3} \in \mathbb{R}^{28\times28\times40}$ and $m_{rdct4} \in \mathbb{R}^{14\times14\times112}$ from the EfficientNet backbone, as input to the position and orientation encoders (without fine-tuning). Table 10 shows the results of four combinations evaluated on the Shop Facade scene from the Cambridge Landmarks dataset. We either provide m_{rdct4}/m_{rdct3} to both encoder heads or use a combination (providing m_{rdct3} to one encoder and m_{rdct4} to the other). Assigning the position encoder with coarser activation

maps yields better accuracy. The opposite trend is observed for the activation maps used by the orientation encoder.

Scene Classification. Our model can be extended to learn multiple scenes in parallel by adding the encoded scene index to the position and orientation tokens. In our design, we predict the scene index by applying a single FC layer to the output of the convolutional backbone. However, the scene identity can also be provided through an external localization means such as a GPS. We thus evaluate the performance of our model when using the ground truth scene index. Interestingly, our model achieved the same accuracy when using the predicted or ground truth scene index, suggesting that our classifier can well distinguish between different scenes.

5. Conclusions

We presented a Transformer-encoder-based approach for *both single- and multi-scene* absolute pose regression, with dedicated tokens for adaptively aggregating sequential representations of multiscale activation maps. By applying dual multi-head attention encoders, our model can focus on different local features depending on the learned task. Our model is shown to compare favorably with previous single and multi-scene APR schemes and provides a new trade-off between localization accuracy and computational complexity. It is the only APR to date to achieve sub-meter accuracy on outdoor benchmarks and demonstrates a more consistent performance with respect to recent APR baselines. It is shown to run at 5 ms per image, having a memory footprint of 24.9 Mb, allowing its deployment on low-end devices.

CRediT authorship contribution statement

Yoli Shavit: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Ron Ferens:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Yosi Keller:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing.

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.

Data availability

Data will be made available on request.

References

Bahdanau, D., Cho, K., Bengio, Y., 2015. Neural machine translation by jointly learning to align and translate. In: Proceedings of the International Conference on Learning Representations. ICLR.

Balntas, V., Li, S., Prisacariu, V., 2018. Relocnet: Continuous metric learning relocalisation using neural nets. In: Proceedings of the European Conference on Computer Vision. ECCV.

Blanton, H., Greenwell, C., Workman, S., Jacobs, N., 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, E., Krull, A., Nowozin, S., Shotton, J., Michel, F., Gumhold, S., Rother, C., 2017. DSAC - differentiable RANSAC for camera localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, Los Alamitos, CA, USA, pp. 2492–2500.

Brachmann, E., Michel, F., Krull, A., Yang, M.Y., Gumhold, S., Rother, C., 2016. Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 3364–3372.

Brachmann, E., Rother, C., 2018. Learning less is more - 6d camera localization via 3d surface regression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 4654–4662.

Brachmann, E., Rother, C., 2021. Visual camera re-localization from RGB and RGB-D images using DSAC. IEEE Trans. Pattern Anal. Mach. Intell. 1.

Brahmbhatt, S., Gu, J., Kim, K., Hays, J., Kautz, J., 2018. Geometry-aware learning of maps for camera localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR.

Cai, M., Shen, C., Reid, I., 2018. A hybrid probabilistic model for camera relocalization. In: Proceedings of the British Machine Vision Conference. p. 238.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S., 2020. End-to-end object detection with transformers. In: Proceedings of the European Conference on Computer Vision. ECCV, Cham, pp. 213–229.

Cavallari, T., Golodetz, S., Lord, N.A., Valentin, J.P.C., d. Stefano, L., Torr, P.H.S., 2017. On-the-fly adaptation of regression forests for online camera relocalisation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, IEEE Computer Society, pp. 218–227.

Devlin, J., Chang, M.W., Lee, K., Toutanova, K., 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1, Minneapolis, Minnesota, pp. 4171–4186.

Ding, M., Wang, Z., Sun, J., Shi, J., Luo, P., 2019. CamNet: Coarse-to-fine retrieval for camera re-localization. In: Proceedings of the IEEE International Conference on Computer Vision. ICCV.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al., 2020. An image is worth 16 × 16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

Duong, N.D., Kacete, A., Soladie, C., Richard, P.Y., Royan, J., 2018. Accurate sparse feature regression forest learning for real-time camera relocalization. In: Proceedings of the International Conference on 3D Vision. 3DV, pp. 643–652.

Frahm, J.M., Fite-Georgel, P., Gallup, D., Johnson, T., Raguram, R., Wu, C., Jen, Y.H., Dunn, E., Clipp, B., Lazebnik, S., Pollefeys, M., 2010. Building rome on a cloudless day. In: Daniilidis, K., Maragos, P., Paragios, N. (Eds.), Computer Vision – ECCV 2010. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 368–381.

Gao, Y., Beijbom, O., Zhang, N., Darrell, T., 2016. Compact bilinear pooling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 317–326.

Glocker, B., Izadi, S., Shotton, J., Criminisi, A., 2013. Real-time rgb-d camera relocalization. In: 2013 IEEE International Symposium on Mixed and Augmented Reality. ISMAR, pp. 173–179.

Jiang, W., Trulls, E., Hosang, J., Tagliasacchi, A., Yi, K.M., 2021. COTR: Correspondence transformer for matching across images. In: Proceedings of the IEEE International Conference on Computer Vision. ICCV.

Kendall, A., Cipolla, R., 2016. Modelling uncertainty in deep learning for camera relocalization. In: Proceedings of the IEEE International Conference on Robotics and Automation. ICRA, pp. 4762–4769.

Kendall, A., Cipolla, R., 2017. Geometric loss functions for camera pose regression with deep learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 6555–6564.

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

Laskar, Z., Melekhov, I., Kalia, S., Kannala, J., 2017. Camera relocalization by computing pairwise relative poses using convolutional neural network. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. ICCVW, pp. 920–929.

Leordeanu, M., Paraicu, I., 2021. Driven by vision: Learning navigation by visual localization and trajectory prediction. Sensors 21, 852.

Lynen, S., Sattler, T., Bosse, M., Hesch, J.A., Pollefeys, M., Siegwart, R., 2015. Get out of my lab: Large-scale, real-time visual-inertial localization. In: Kavraki, L.E., Hsu, D., Buchli, J. (Eds.), Robotics: Science and Systems XI.

Marcu, A., Costea, D., Slusanschi, E., Leordeanu, M., 2018. A multi-stage multi-task neural network for aerial scene interpretation and geolocalization. arXiv preprint arXiv:1804.01322.

Melaskyriazi, L., 2019. efficientnet-pytorch.

Melekhov, I., Ylioinas, J., Kannala, J., Rahtu, E., 2017. Image-based localization using hourglass networks. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. ICCVW, pp. 870–877.

Mera-Trujillo, M., Smith, B., Fragoso, V., 2020. Efficient scene compression for visual-based localization. In: Proceedings of the International Conference on 3D Vision. 3DV, IEEE Computer Society, Los Alamitos, CA, USA, pp. 1–10.

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

Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A., 2020. Superglue: Learning feature matching with graph neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 4938–4947.

Sarlin, P.E., DeTone, D., Yang, T.Y., Avetisyan, A., Straub, J., Malisiewicz, T., Bulò, R., Kontschieder, P., Balntas, V., 2023. Orienternet: Visual localization in 2d public maps with neural matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21632–21642.

Sarlin, P.E., Unagar, A., Larsson, M., Germain, H., Toft, C., Larsson, V., Pollefeys, M., Lepetit, V., Hammarstrand, L., Kahl, F., Sattler, T., 2021. Back to the feature: Learning robust camera localization from pixels to pose. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR.

Sattler, T., Leibe, B., Kobbelt, L., 2017. Efficient effective prioritized matching for large-scale image-based localization. IEEE Trans. Pattern Anal. Mach. Intell. 39, 1744–1756.

Sattler, T., Zhou, Q., Pollefeys, M., Leal-Taixe, L., 2019. Understanding the limitations of CNN-based absolute camera pose regression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, IEEE Computer Society, Los Alamitos, CA, USA, pp. 3297–3307.

Shavit, Y., Ferens, R., 2019. Introduction to camera pose estimation with deep learning. arXiv preprint arXiv:1907.05272.

Shavit, Y., Ferens, R., 2021. Do we really need scene-specific pose encoders? In: Proceedings of the International Conference on Pattern Recognition. ICPR, IEEE, pp. 3186–3192.

Shavit, Y., Ferens, R., Keller, Y., 2021. Learning multi-scene absolute pose regression with transformers. In: Proceedings of the IEEE International Conference on Computer Vision. ICCV, pp. 2713–2722.

Shavit, Y., Keller, Y., 2022. Camera pose auto-encoders for improving pose regression. In: Proceedings of the European Conference on Computer Vision. ECCV, pp. 140–157.

Shi, Y., Cai, J.X., Shavit, Y., Mu, T.J., Feng, W., Zhang, K., 2022. Clustergnn: Cluster-based coarse-to-fine graph neural network for efficient feature matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12517–12526.

Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., Fitzgibbon, A., 2013. Scene coordinate regression forests for camera relocalization in RGB-D images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 2930–2937.

Sun, J., Shen, Z., Wang, Y., Bao, H., Zhou, X., 2021. LoFTR: Detector-free local feature matching with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR.

Taira, H., Okutomi, M., Sattler, T., Cimpoi, M., Pollefeys, M., Sivic, J., Pajdla, T., Torii, A., 2019. Inloc: Indoor visual localization with dense matching and view synthesis. IEEE Trans. Pattern Anal. Mach. Intell. 1.

Tan, M., Le, Q., 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. Long Beach, California, USA, pp. 6105–6114.

Teed, Z., Deng, J., 2020. RAFT: Recurrent all-pairs field transforms for optical flow. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J. (Eds.), Proceedings of the European Conference on Computer Vision. ECCV, Springer, pp. 402–419.

Torii, A., Arandjelovic, R., Sivic, J., Okutomi, M., Pajdla, T., 2015. 24/7 place recognition by view synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR.

Torii, A., Arandjelovic, R., Sivic, J., Okutomi, M., Pajdla, T., 2018. 24/7 place recognition by view synthesis. IEEE Trans. Pattern Anal. Mach. Intell. 40, 257–271.

Turkoglu, M., Brachmann, E., Schindler, K., Brostow, G.J., Monszpart, A., 2021. Visual camera re-localization using graph neural networks and relative pose supervision. In: Proceedings of the International Conference on 3D Vision. 3DV, Los Alamitos, CA, USA, pp. 145–155.

Valada, A., Radwan, N., Burgard, W., 2018. Deep auxiliary learning for visual localization and odometry. In: Proceedings of the IEEE International Conference on Robotics and Automation. ICRA, pp. 6939–6946.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I., 2017. Attention is all you need. In: Advances in Neural Information Processing Systems. NIPS, pp. 5998–6008.

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

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

Wu, J., Ma, L., Hu, X., 2017. Delving deeper into convolutional neural networks for camera relocalization. In: Proceedings of the IEEE International Conference on Robotics and Automation. ICRA, pp. 5644–5651.

Xue, F., Wang, X., Yan, Z., Wang, Q., Wang, J., Zha, H., 2019. Local supports global: Deep camera relocalization with sequence enhancement. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR.

Xue, F., Wu, X., Cai, S., Wang, J., 2020a. Learning multi-view camera relocalization with graph neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 11372–11381.

Xue, F., Wu, X., Cai, S., Wang, J., 2020b. Learning multi-view camera relocalization with graph neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR.

Yen-Chen, L., Florence, P., Barron, J.T., Rodriguez, A., Isola, P., Lin, T.Y., 2021. iNeRF: Inverting neural radiance fields for pose estimation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS.

Zhang, X., Xiong, H., Zhou, W., Lin, W., Tian, Q., 2016. Picking deep filter responses for fine-grained image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 1134–1142.

Zhou, Q., Sattler, T., Pollefeys, M., Leal-Taixe, L., 2020. To learn or not to learn: Visual localization from essential matrices. In: Proceedings of the IEEE International Conference on Robotics and Automation. ICRA, IEEE, pp. 3319–3326.