Coarse-to-Fine Multi-Scene Pose Regression With Transformers

Yoli Shavit, Ron Ferens, and Yosi Keller

***Abstract*—Absolute camera pose regressors estimate the position and orientation of a camera given the captured image alone. Typically, a convolutional backbone with a multi-layer perceptron (MLP) head is trained using images and pose labels to embed a single reference scene at a time. Recently, this scheme was extended to learn 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 pose predictions. This allows our model to focus on general features that are informative for localization, while embedding multiple scenes in parallel. We extend our previous MS-Transformer approach Shavit et al. (2021) by introducing a mixed classification-regression architecture that improves the localization accuracy. Our method is evaluated on commonly benchmark indoor and outdoor datasets and has been shown to exceed both multi-scene and state-of-the-art single-scene absolute pose regressors.**

***Index Terms*—Absolute pose regression, coarse-to-fine camera localization, deep learning.**

I. Introduction

LOCALIZING a camera using a query image is essential for a variety of computer vision applications, including indoor navigation, augmented reality, and autonomous driving. Current approaches to estimating a camera's position and orientation offer different trade-offs between accuracy, runtime, and memory requirements. For example, hierarchical Structure-based localization pipelines (SbP) [2], [3], [4] achieve state-of-the-art (SOTA) pose accuracy but require a response time of hundreds of milliseconds, a large memory footprint, and client-server connectivity. These approaches employ image retrieval (IR) to find images similar to the query image while extracting and matching local image features. The 2D-2D matches extracted are mapped to 2D-3D correspondences through depth or a 3D point cloud that are used to estimate the camera pose with Perspective-n-Point (PnP) and RANSAC [5]. Absolute pose regressors (APRs), on the other hand, estimate the camera pose in a single forward pass, using only the query image. They are faster and can be deployed on a thin client as standalone applications due to their small memory footprint. However, APRs are also an order of magnitude less accurate than SbP approaches and other methods that use 3D data at inference time [6]. Furthermore, most APRs are designed to embed a single scene at a time, implying that, for a dataset with N scenes (such as a hospital with many wards and rooms), N models must be trained, deployed and selected during inference. This paper aims at improving the accuracy of APRs by extending the current single-scene paradigm to simultaneously learn multiple scenes. Absolute camera pose regression was first suggested by Kendall et al. [7]. Following the success of convolutional neural networks (CNNs) in multiple computer vision tasks, the authors suggested adapting a GoogleNet architecture to camera pose regression by attaching a multilayer perceptron (MLP) head. A proposed architecture called PoseNet provided a fast and lightweight solution for camera localization. However, it also suffered from poor accuracy and limited generalization. Various absolute pose regression methods were suggested to address these issues, proposing to modify the backbone and MLP architecture [8], [9], [10], [11], [12], as well as different loss formulations and optimization strategies [13], [14], [15]. APRs share two common traits: (1) using a CNN backbone to produce a global latent vector to regress the pose, and (2) training a model for each scene (scene-specific APRs). Blanton et al. [16] extended single-scene absolute pose regression to a multi-scene paradigm. Similarly to existing APRs, this method applies a CNN backbone to generate 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 chosen 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 the recent successful applications of Transformers to computer vision tasks such as object detection [17] and image recognition [18]. These works demonstrated the effectivity of *encoders* in focusing on latent features (in image patches or activation maps) that are informative for particular tasks, by self-attention aggregation. In addition, *decoders* were shown to successfully generate multiple independent predictions, corresponding to queries, based on the input embedding [17]. Similarly, we propose to apply Transformers to multiscene absolute pose regression, using *encoders* to focus on pose-informative features and *decoders* to transform

Manuscript received 12 September 2022; revised 3 April 2023; accepted 17 August 2023. Date of publication 31 August 2023; date of current version 3 November 2023. Recommended for acceptance by V. Lepetit. *(Corresponding author: Yosi Keller.)*

The authors are with the Faculty of Engineering, Bar Ilan University, Ramat-Gan 5290002, Israel (e-mail: yolisha@gmail.com; ronferens@gmail.com; yosi.keller@gmail.com).

We make our code publicly available from https://github.com/yolish/c2f-ms-transformer.

Digital Object Identifier 10.1109/TPAMI.2023.3310929

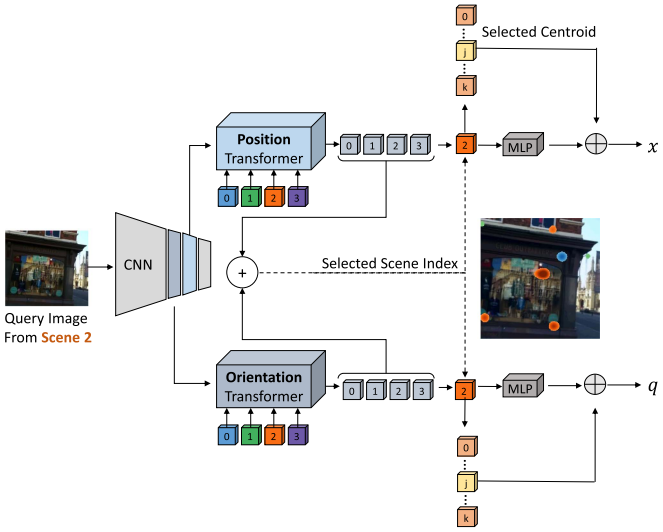


Fig. 1. Coarse-to-Fine Multi-scene absolute pose regression with Transformers. Two Transformers separately attend to position- and orientation- informative features from a convolutional backbone. Scene-specific queries (0–3) are further encoded with aggregated activation maps into latent representations, 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, at the scene level, are used to further select position and orientation centroids and regress the respective residuals. The position x and the orientation q are given by the sum of selected centroids and the regressed residuals.

encoded scene identifiers into 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 decoder's 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 further extend our earlier research findings [1], by reframing camera pose regression as a classification-regression problem with a coarse-to-fine approach. Specifically, by quantizing the position and orientation domains of a scene, APR can be formulated as a classification problem. Although classification-based regression schemes were found to be robust [19], [20], [21], they require a refinement phase to overcome quantization errors. Therefore, we propose a mixed classification-regression architecture. We first classify the coarse camera location (scene) and then predict the clusters within the scene based on the selected decoders' embeddings, which provide an initial coarse estimate of camera orientation and position. The regression network finetunes these initial estimates, by learning the residuals with respect to the predicted quantized clusters' centers. To the best of our knowledge, our approach is the first to suggest such a coarse-to-fine classification-regression approach for camera pose regression.

We evaluated our approach on two commonly reference 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 also, importantly, provides a new SOTA for *single-scene APRs,* outperforming our previous SOTA results [1]. Furthermore, we show that our approach achieves competitive results even when trained on multiple datasets with 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 extend and build upon our earlier findings [1] by approaching the APR problem through a coarse-to-fine classification-regression framework, and presenting a mixed architecture that combines both classification and regression.
- 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.

II. Related Work

A. Camera Localization

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 [3], [4], [22] and relative pose regression methods [23], [24], [25]. IR can also be applied to estimate 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. [6] 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 [2], [6], 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 Structure-based pose estimation pipelines (SbP) [3], [4], [22] are based on a two-phase approach, using 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 then mapped into 2D-3D matches. The resulting matches are passed to PnP-RANSAC for estimating the camera pose. Such pipelines have been shown to achieve SOTA accuracy on large-scale benchmarks with challenging conditions [3], [4]. A different body of works directly regresses the 3D scene coordinates from 2D positions in the image. Brachmann and Rother derived the DSAC [26] and the follow-up DSAC++

[27] 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 Scene Coordinate Regression (SCR) methods require the query image and the intrinsics of the query camera as input, and achieve SOTA accuracy comparable to SbP methods. Similarly to single-scene APRs, a model must be trained per scene.

Relative Pose Regression methods typically 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-neighbor images, and a relative motion regression is computed separately 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 [23], [24], [25]. These relative pose regressors (RPR) have been shown to generalize better than APR and improve accuracy in small-scale indoor benchmarks [25]. Rather than using (encoded) images with pose labels, Aha et al. [28] suggested employing anchor points that are uniformly distributed throughout the scene. The proposed method, named AnchorNet, predicts which anchor points appear in the query image along with their relative location, allowing one to compute an anchor-based absolute pose estimate. The query pose is then computed as a weighted average of all poses. Unlike other RPRs, AnchorNet is a single-scene approach and is trained per scene. Some RPRs rely on sequential acquisition of images over time [29], [30], while combining relative and absolute regression, achieving an improved pose accuracy [29], [30].

Absolute Pose Regression was first proposed by [7] 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 much less accurate than the 3D-based methods, but enabled pose estimation using a single forward pass. In order to improve the localization accuracy, contemporary APRs studied different CNN backbones [8], [10] and MLP heads [8]. Overfitting was addressed by averaging the predictions of multiple models with randomly dropped activations [13], or by reducing the dimensionality of the global image encoding using Long-Short-Term-Memory (LSTM) layers [9]. Other work focused on the loss formulation for absolute pose regression to allow adaptive weighting of position- and orientation-associated errors. Kendall et al. [14] suggested optimizing the parameters that balance both losses to improve accuracy and avoid manual fine-tuning. This formulation was adopted by many pose regressors. Alternative orientation representations were also proposed to improve pose loss [31]. The use of additional sensors, such as inertial sensors, was also suggested to improve localization accuracy [31]. More recently, Wang et al. [11] proposed using attention to guide the regression process by applying self-attention to the output of the CNN backbone. The new attention-based representation was used to regress the pose with an MLP head. Although 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, and (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 Multi-Scene PoseNet (MSPN), a novel approach to multi-scene absolute pose regression [16], 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 to regress the pose. An activation map of a CNN backbone, which is shared across scenes, is used both for scene classification and to regress 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 [32].

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.

B. Attention and Transformers

Attention mechanisms [33] are layers of neural networks that aggregate information from the entire input sequence. The aggregation is often computed by a sequence-to-sequence architecture, where the inner-products (interactions) between the two sequences are used to compute the aggregation weights. Attention models consist of an Encoder and Decoder. The Encoder implements self-attention that maps the input sequence into a higher-dimensional space that is fed into the Decoder alongside a query sequence, outputting the result sequence. Attention allows to numerically emphasize the contribution of the task-informative image locations, in contrast to the visual clutter. Transformers were introduced by Vaswani et al. [34] as a novel formulation of attention-based Encoders and Decoders for sequence encoding that does not use RNN layers such as LSTM and GRU. Transformers consist of multiple stacked Multi-Head Attention and Feed Forward layers. As no recurrent layers are used, the relative position and sequential order of the sequence elements are induced by adding positional encodings to the embedded representation. Transformers were shown to outperform RNNs in encoding long sequences, and were applied in multiple recent works in natural language processing (NLP) [35], [36] and computer vision [17], [18]. In this work, we propose a hybrid CNN-Transformer architecture, inspired by recent advances in visual transformers for multi-object detection [17].

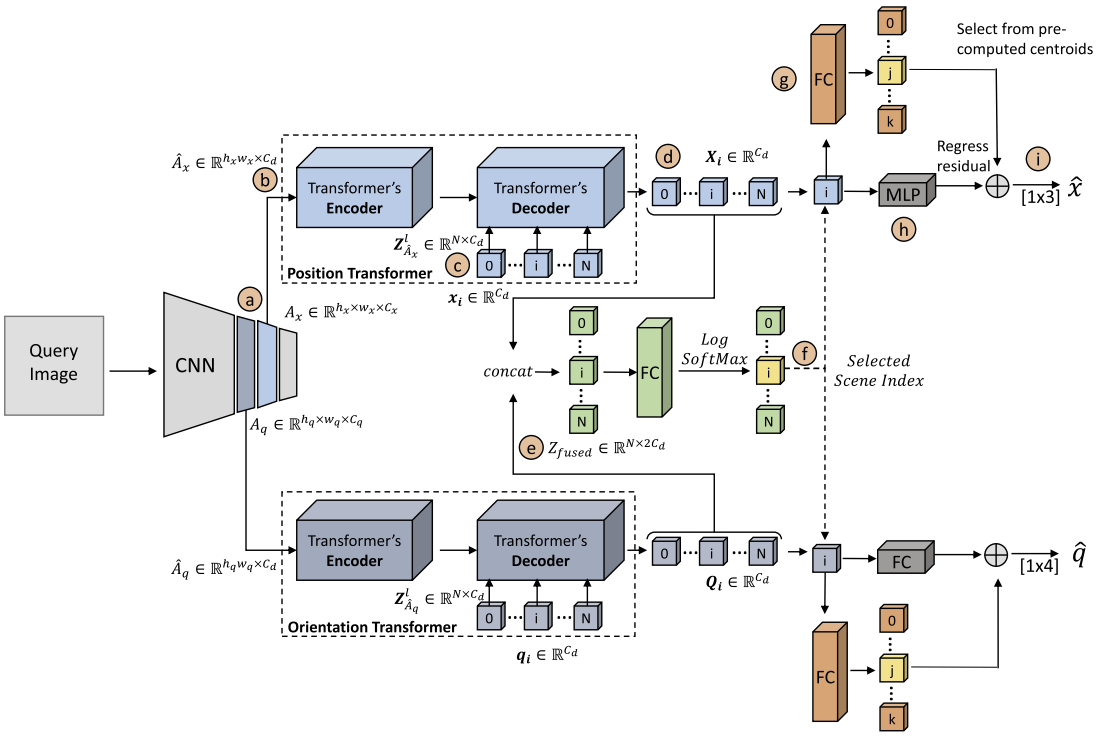


Fig. 2. Architecture of our proposed model.

We employ encoders to adaptively aggregate activation maps for position and orientation regression and use decoders to decode aggregated representations with respect to query scenes encoding.

III. 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 pose of the camera $\mathbf{p}$ 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 that encodes its 3D orientation. Following the success of recent visual Transformers [17], [18], we employ separate position and orientational *Transformer Encoders* for adaptive aggregation of (flattened) intermediate activation maps computed by a convolutional backbone. In particular, as depicted in Figs. 3 and 4 and Section III-A, 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 orientation encoder.

To attend to N scenes, we also apply separate positional and orientational *Transformer Decoders*, which 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 [17], where a single activation map is queried by multiple queries, each related to a different task. Together, the encoder-decoder Transformer architecture allows attending localization-informative image content while learning multiple scenes at once. To regress the pose, the scene is first classified as described in Section III-B. By concatenating $\{\mathbf{X}_i\}_1^N$ and $\{\mathbf{Q}_i\}_1^N$, the embeddings of the detected scene $\{\mathbf{X}_i, \mathbf{Q}_i\}$ are regressed by the MLP heads, as detailed in Section III-C. The architecture of our model is shown in Fig. 2.

A. Image Encoding Using Transformers

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 the position and orientation regression, respectively (Fig. 2(a)).

In order to transform activation maps into Transformer-compatible inputs, we follow the same sequence preparation procedure as in [17]. 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 (Fig. 2(b)). 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 concatenating the two corresponding embedding vectors:

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

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

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

where $\mathbf{E_A}$ 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 [17], with standard Encoder and Decoders modified to add the positional encoding at each attention layer. A Transformer Encoder is composed of L identical layers, each consisting of multi-head attention (MHA) and multilayer perceptron (MLP) modules. Each layer l, $l = 1..L$, performs the following computation by applying a LayerNorm (LN) [37] before each module and adding back the input with residual connections:

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

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

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

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

In our model, $A_{\mathbf{x}}$ and $A_{\mathbf{q}}$ are passed to a separate Transformer Encoders which apply the aforementioned mechanism. We denote the outputs $\mathbf{Z}^{L}_{\widehat{A_x}}$ and $\mathbf{Z}^{L}_{\widehat{A_q}}$, respectively (Fig. 2(c)).

B. Scene Classification

Given a dataset with N scenes, the Transformer Decoders first apply self-attention, as in (3), to the two learned query sequences $\{\mathbf{x}_i\}_1^N$ and $\{\mathbf{q}_i\}_1^N$ (Fig. 2(d)), for the position and orientation decoders, respectively. Equations (3)–(4) are then applied again, but this time computing encoder-decoder attention instead of self-attention. Unlike the earlier autoregressive decoders [34], this architecture outputs predictions in parallel for all positions. We refer the reader to [17], [34] for detailed definitions of the MHA operation and parallel decoding.

The Transformers Decoders output the sequences $\{\mathbf{X}_i\}_1^N$ and $\{\mathbf{Q}_i\}_1^N$, encoding each scene with a latent embedding (Fig. 2(e)). Since a query image corresponds to a single scene from which the image was taken, the respective latent embedding need to be selected. In order to select the corresponding scene, we append the outputs of the two transformers (Fig. 2(f)) as $\{\mathbf{Z}_{i\_fused}\}_1^N$, such that

$$\mathbf{Z}_{i\_fused} = \begin{bmatrix} \mathbf{X}_i \\ \mathbf{Q}_i \end{bmatrix} \in \mathbb{R}^{2C_d}, \tag{6}$$

and pass them through a fully connected layer followed by Log SoftMax. The embedding vectors $\{\mathbf{X}_i, \mathbf{Q}_i\}$ corresponding to the maximal probability of scene classification are then chosen (Fig. 2(g)).

C. Pose Classification-Regression

The selected Transformers outputs $\{\mathbf{X}_i, \mathbf{Q}_i\}$ are used to refine the location of the camera, from scene level to cluster level, and to perform residual regression. During training, we precompute K position and orientation centroids $\{c^x_k, c^q_k\}_1^K$, respectively, for each scene using the K -means algorithm [38]. Given a query image, a fully connected layer and SoftMax are applied to $\{\mathbf{X}_i, \mathbf{Q}_i\}$ to select the respective centroids $\{c^x_{k_0}, c^q_{k_0}\}$ (Fig. 2(h)). To further regress the residuals of $\{\mathbf{X}_i, \mathbf{Q}_i\}$, we apply two MLP heads with a single hidden layer and gelu non-linearity (Fig. 2(i)). The first is applied to regress the position refinement $\mathbf{\Delta x}$, while the other MLP refines the orientation by $\mathbf{\Delta q}$. The position and orientation vectors (Fig. 2(j)), $\mathbf{x}$ and $\mathbf{q}$, are thus given by

$$\mathbf{x} = c^x_{k_0} + \mathbf{\Delta x}, \mathbf{q} = c^q_{k_0} + \mathbf{\Delta q}. \tag{7}$$

D. 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:

$$L_{\mathbf{x}} = ||\mathbf{x}_0 - \mathbf{x}||_2 \tag{8}$$

$$L_{\mathbf{q}} = ||\mathbf{q_0} - \frac{\mathbf{q}}{||\mathbf{q}||}||_2 \tag{9}$$

where q is normalized to a unit norm quaternion to ensure that it is a valid orientation encoding. We combine the two losses using the camera pose loss formulation suggested by Kendall et al. [14]:

$$L_{\mathbf{p}} = L_{\mathbf{x}} \exp(-s_{\mathbf{x}}) + s_{\mathbf{x}} + L_{\mathbf{q}} \exp(-s_{\mathbf{q}}) + s_{\mathbf{q}} \tag{10}$$

where $s_{\mathbf{x}}$ and $s_{\mathbf{q}}$ are learned parameters that control the balance between the two losses. As our model is also required to classify the scene from which the image was taken and the centroid within each scene, we further add the Negative Log-likelihood (NLL) loss term, computed with respect to the ground truth scene index s_0 and the ground truth position and orientation centroids $\mathbf{c}_{x,0}$ and $\mathbf{c}_{q,0}$, respectively. Given an estimated pose p and the log-probability distributions s, c_x and c_q of the predicted scene, position and orientation centroids, our overall loss is given by:

$$\begin{aligned} L_{\mathbf{multi-scene}} = L_{\mathbf{p}} &+ NLL(\mathbf{s}, \mathbf{s_0}) \\ &+ NLL(\mathbf{c_x}, \mathbf{c_{x,0}}) + NLL(\mathbf{c_q}, \mathbf{c_{q,0}}) \end{aligned} \tag{11}$$

E. Implementation Details

Our model is implemented in PyTorch [39]. We use a pre-trained EfficientNet-B0 [40], 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 nonlinearity and with a dropout of $p = 0.1$. Each layer (encoder / decoder) uses four MHA heads and a two-layer MLP with a hidden dimension $C_h = C_d$.

The two MLP heads, regressing the position and orientation residual vectors, respectively, expand the decoder dimension to $\mathbb{R}^{1024}$ with a single hidden layer. Our code is publicly available,[1] providing the model implementation along with a

[1] https://github.com/yolish/c2f-ms-transformer

TABLE I
TRAINING TIME OF OUR MODEL

Dataset	#Images	#Epochs	Training Time [hours]
Cambridge	3,834	550	24
7Scenes	26,000	30	10

training and evaluation framework, detailed configuration files, and pre-trained models.

IV. EXPERIMENTAL RESULTS

A. Experimental Setup

Datasets: We evaluate our approach using the Cambridge Landmarks [7] and 7Scenes [41] datasets, which are commonly used to evaluate pose regression methods. The Cambridge Landmarks dataset consists of six medium-sized scenes ($\sim 900 - 5500\,m^2$) 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 ($\sim 1 - 10\,m^2$) set in an office indoor environment.

Training Details: We optimize our model to minimize the loss in (11) using Adam, with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-10}$. The loss parameters ((10)) are initialized as in [29]. Throughout all experiments, we use a batch size of 8 and an initial learning rate of $\lambda = 10^{-4}$. For each dataset, we initialize our model from the respective pre-trained MS-Transformer model [1], which we extend in this work. We use $k = 4$ for the number of position and orientation clusters for the CambridgeLandmarks dataset and $k = 1$, $k = 2$ for the number of position and orientation clusters for the 7Scenes dataset. 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 unknown, and we rely on the prediction of our model, taking the index with the highest (log) probability. The same logic is applied for training the classifiers of the position and orientation centroids. Note that instead of training a model per scene (typically with scene-specific training hyperparameters) as in single APRs, here we train a single model for *all* the scenes together. All experiments reported in this paper were performed on an 8GB NVIDIA GeForce GTX 1080 GPU. We follow the augmentation procedure described in [7]. During training, the images are first rescaled so that the smaller edge is resized to 256 pixels and then randomly cropped to a 224×224-sized image. Additionally, brightness, contrast, and saturation are randomly jittered. At test time, the center crop is taken after rescaling without any further augmentations. For the 7Scenes dataset, we train with the aforementioned augmentation scheme for 30 epochs and reduce the learning rate by half every 10 epochs. For the Cambridge Landmarks dataset, we train for 550 epochs and reduce the learning rate by half every 200 epochs. For both datasets, we fine-tune the respective models from [1]. Table I shows the training times for these models.

B. Comparative Localization Analysis

Our proposed coarse-to-fine approach (c2f-MS-Transformer) is a *multiscene* absolute pose regression (MS-APR) paradigm. As such, we compare it to single-scene (SS-APR) and MS-APRs using the CambridgeLandmarks and 7Scenes datasets (Tables II and III, respectively). For our approach, we report the results when training from scratch and when initializing from the respective MS-Transformer [1] model. To provide a wider experimental overview and context, we also include the results of representative methods from other classes of localization schemes (see Section II), namely: SbP, SCR, Sequence-based (Seq.) and IR. We report the median position and orientation errors and the average of errors across scenes. The DSAC* approach (SCR family) achieves the overall SOTA accuracy by a significant margin for the CambridgeLandmarks and 7Scenes datasets, where the closest second is the sequential RPR-based VLocNet++ [30]. Within the APR family, our method is the most accurate across datasets. MSPN [16], is, to the best of our knowledge, the only other MS-APR approach to date. Since it was trained on a different combination of scenes from the CambridgeLandmarks dataset, we applied the best performing model reported by the authors on this dataset [16]. Our method consistently outperforms MSPN across outdoor and indoor scenes, reducing both position and orientation errors. Moreover, compared to single- and multiple-scene APRs, the c2f-MS-Transformer achieves SOTA positional accuracy in 10 of the 11 scenes (through the CambridgeLandmarks and 7Scenes datasets). Furthermore, our method achieves the lowest average positional error for both indoor and outdoor localization and is the only APR approach to report an average error below one meter in outdoor scenes. In most of the scenes reported, c2f-MS-Transformer outperforms our previous method [1] in both position and orientation accuracy. Interestingly, the two best-performing APRs on the 7Scenes dataset (AttLoc and our method) both use the attention mechanism for pose regression.

We note that different localization approaches offer different trade-offs between accuracy and other attributes such as generalization and storage. Table IV summarizes the key characteristics of the classes of localization schemes compared in Tables II and III. We consider whether the localization scheme can localize using only the query image (Query only) as opposed to requiring a database or a sequential acquisition (as in SbP, Seq, IR and RPR methods), whether it can localize without a 3D model (No 3D), whether it can learn multiple scenes at once (Multi-scene), generalize to unseen scenes (Generalization) and whether it provides high accuracy (sub-degree error and < 0.3 meters and < 0.05 meters for mid- and small-scale scenes, respectively). Multi-Scene APR approaches can operate without 3D information using the query image alone, while learning multiple scenes at once. However, they are less accurate than SbP and Sequential and SCR methods and cannot generalize to unseen scenes, as opposed to SbP, IR and RPR approaches. Tables XII and XIII further compare runtime, storage, and the model size of our method and other representative approaches.

We can further extend the notion of multi-scene learning to multi-dataset learning, where a single model is trained on

TABLE II
LOCALIZATION RESULTS FOR CAMBRIDGE LANDMARKS DATASET

	Method	College	Hospital	Shops	St. Mary's	Avg.
SbP	ActiveSearch	0.42,0.6°	0.44,1.0°	0.12,0.40°	0.19,0.5°	0.29,0.63°
	InLoc[3]	0.18,0.6°	1.2,0.6°	0.48,1.0°	0.46,0.8°	0.11,0.5°
SCR	DSAC [26]	0.30,0.5°	0.33,0.6°	0.09,0.4°	0.55,1.6°	0.31,0.78°
	DSAC++ [27]	**0.18,0.3°**	**0.20,0.3°**	0.06,**0.3°**	**0.13,0.4°**	**0.14,0.33°**
	DSAC* [42] [42]	**0.18,0.3°**	0.21,0.4°	**0.05,0.3°**	0.15,0.5°	0.15,0.4°
Seq.	MapNet [31]	1.08,1.9°	1.94,3.9°	1.49,4.2°	2.00,4.5°	1.63,3.6°
	GL-Net [43]	0.59,0.7°	1.88,2.8°	0.50,2.9°	1.90,3.3°	1.22,2.4°
	VLocNet [29]	0.83/1.42°	1.07/2.41°	0.59/3.53°	0.63/3.91°	0.78,2.81°
IR	VLAD [44]	2.80,5.7°	4.01,7.1°	1.11,7.6°	2.31,8.0°	2.56,7.1°
	VLAD+Inter [6]	1.48,4.5°	2.68,4.6°	0.90,4.3°	1.62,6.1°	1.67,4.9°
RPR	EssNet [45]	0.76,1.9°	1.39,2.8°	0.84,4.3°	1.32,4.7°	1.08,3.4°
	NC-EssNet [45]	0.61,1.6°	0.95,2.7°	0.70,3.4°	1.12,3.6°	0.85,2.8°
	RelocGNN[42]	0.48,1.0°	1.14,2.5°	0.48,2.5°	1.52,3.2°	0.91,2.3°
SS-APR	PoseNet [7]	1.92,5.40°	2.31,5.38°	1.46,8.08°	2.65,8.48°	2.08,6.83°
	BayesianPN [13]	1.74,4.06°	2.57,5.14°	1.25,7.54°	2.11,8.38°	1.91,6.28°
	LSTM-PN [9]	0.99,3.65°	1.51,4.29°	1.18,7.44°	1.52,6.68°	1.30,5.57°
	SVS-Pose [8]	1.06,2.81°	1.50,4.03°	0.63,5.73°	2.11,8.11°	1.32,5.17°
	GPoseNet [12]	1.61,2.29°	2.62,3.89°	1.14,5.73°	2.93,6.46°	2.07,4.59°
	PoseNetLearn [14]	0.99,1.06°	2.17,2.94°	1.05,3.97°	1.49,3.43°	1.42,2.85°
	GeoPoseNet [14]	0.88,**1.04°**	3.20,3.29°	0.88,3.78°	1.57,**3.32°**	1.63,2.86°
	MapNet [31]	1.07,1.89°	1.94,3.91°	1.49,4.22°	2.00,4.53°	1.62,3.64°
	IRPNet [10]	1.18,2.19°	1.87,3.38°	0.72,3.47°	1.87,4.94°	1.41,3.50°
MS-APR	MSPN [16]	1.73/3.65°	2.55/4.05°	2.92/7.49°	2.67/6.18°	2.47/5.34°
	MS-Trans[1]	0.83,1.47°	1.81,**2.39°**	0.86,3.07°	1.62,3.99°	1.28,**2.73°**
	c2f-MS-Trans w/o init (ours)	0.82,2.34°	2.10,2.79°	0.90,3.21°	1.27/3.69°	1.27,**3.01**
	c2f-MS-Trans (ours)	**0.70**,2.69°	**1.48**,2.94°	**0.59,2.88°**	**1.14**,3.88°	**0.98**/3.10°

We report the average median position/orientation errors in meters/degrees and the respective rankings. The most accurate global results are highlighted in underlined bold, while the most accurate APR results are highlighted in bold.

completely separate datasets, potentially having different challenges and attributes. To evaluate the effect of such an extension, we jointly trained our model on both the 7Scenes and Cambridge Landmarks datasets. Table V shows the average pose error per dataset for a *state-of-the-art* single-scene APR [14] and our model, trained in multi-scene and in multi-dataset modes. Although some degradation is observed when both datasets are jointly trained, our model still maintains competitive performance and outperforms the single-scene model. This is despite the fact that the two datasets represent significantly different environments and challenges (medium-scale outdoor versus small-scale indoor). We also evaluated 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.

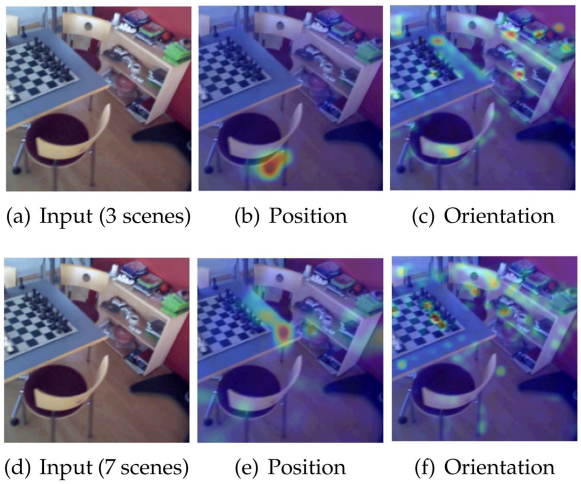


Fig. 3. Transformer Encoder attention visualizations for a varying number of training scenes (three and seven). As we train our scheme using more scenes (second row), the positional attention is able to better localize corner-like image cues (e compared to b). The orientational attention is able to better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

C. Attention Maps Visualization and Interpretation

In attention-based schemes, attention maps provide intuitive interpretation and understanding of the visual cues captured by Transformer Encoders. Using heatmaps overlaid on the input images, we visualize the upsampled attention weights of the last encoder layer. Fig. 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 seven scenes (top and bottom rows, respectively). Training on more (seven) scenes allows the network to better capture informative image cues for both positional and orientational embedding. In particular, positional attention focuses on corners and blob-like objects, while orientational attention emphasizes elongated edges. We also visualize the attentions $\{\mathbf{X}_i\}_1^N$ on the output of the positional decoder for an image from the *OldHospital* scene (Fig. 4). Each activation corresponds to a particular scene. In fact, the

TABLE III
LOCALIZATION RESULTS FOR 7SCENES DATASET

		Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs	Avg.
SbP	Active Search [2]	0.04,2.0°	0.03,1.5°	0.02,1.5°	0.09,3.6°	0.08,3.1°	0.07,3.4°	0.03,2.2°	0.05,2.47°
	InLoc[3]	0.03,1.05°	0.03,1.07°	0.02,1.16°	0.03,1.05°	0.05,1.55°	0.04,1.31°	0.09,2.47°	0.04,1.44°
SCR	DSAC [26]	**0.02**,0.7°	0.03,1.0°	0.02,1.30°	0.03, 1.0°	0.05,1.30°	0.05,1.5°	1.90,49.4°	0.30,8.03°
	DSAC++[27]	**0.02**,**0.5**°	0.02 ,**0.9**°	**0.01**,**0.8**°	0.03,**0.7**°	0.04,**1.1**°	0.04,**1.1**°	0.09,2.6°	0.04,**1.10**°
	DSAC* [42]	**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°	0.03,1.37°
Seq.	LsG [46]	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°	0.19,7.47°
	MapNet[31]	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°	0.21,7.78°
	GL-Net[43]	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°	0.19,7.50°
	VLocNet [29]	0.04,1.71°	0.04,5.34°	0.05,6.64°	0.04,1.95°	0.04,2.28°	0.04,2.20°	0.10,6.48°	0.05,3.80°
	VLocNet++ [30]	**0.02**,1.44°	**0.01**,1.39°	0.02,0.99°	**0.02**,1.14°	**0.02**,1.45°	**0.03**,2.27°	**0.02**,**1.08**°	**0.02**,1.39°
IR	VLAD [44]	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°	0.26,12.46°
	VLAD+Inter[6]	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°	0.24,11.71°
RPR	NN-Net [24]	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°	0.21,9.30°
	RelocNet[23]	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°	0.21,6.73°
	EssNet [45]	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°	0.22,8.03°
	NC-EssNet [45]	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°	0.21,7.50°
	RelocGNN[42]	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°	0.16,5.24°
	CamNet [25]	0.04,1.73°	0.03, 1.74°	0.05,1.98°	0.04,1.62°	0.04,1.64°	0.04,1.63°	0.04,1.51°	0.04,1.69°
SS-APR	PoseNet [7]	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°	0.44,10.44°
	BayesianPN [13]	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°	0.47,9.81°
	LSTM-PN [9]	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°	0.31,9.85°
	GPoseNet [12]	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°	0.31,9.95°
	PoseNetLearn[14]	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°	0.24,7.87°
	GeoPoseNet[14]	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°	0.23,8.12°
	IRPNet[10]	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°	0.23,8.49°
	AttLoc[11]	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°	0.20,7.56°
MS-APR	MSPN[16]	**0.09**/4.76°	0.29/10.5°	0.16/13.1°	**0.16**/6.80°	0.19/5.50°	0.21/6.61°	0.31/11.6°	0.20/7.56°
	MS-Trans[1]	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°	0.18,**7.28**°
	c2f-MS-Trans w/o init. (ours)	0.10,4.97°	0.23,9.46°	0.13,12.4°	0.17,5.71°	0.17,4.48	0.17,6.23°	0.25, 9.62°	0.18,7.55°
	c2f-MS-Trans (ours)	0.10,4.60°	**0.24**,9.88°	**0.12**,12.3°	**0.16**,5.64°	**0.16**,4.42°	**0.16**,6.39°	**0.25,7.76**°	**0.17/7.28**°

We report the average of median position/orientation errors in meters/degrees and the respective rankings. The most accurate global results are highlighted in underlined bold, while the most accurate APR results are highlighted in bold.

TABLE IV
ATTRIBUTES OF THE MAIN CLASSES OF LOCALIZATION SCHEMES

Family	Query-only	No 3D	Multi-Scene	Generalization	High Accuracy
SbP			✓	✓	✓
SCR	✓	*			✓
Seq		✓	✓	✓	✓
IR		✓	✓	✓	
RPR		✓	✓	✓	
Single-Scene APR	✓	✓			
Multi-Scene APR	✓	✓	✓		

We indicate that an approach has an attribute by a ✓ sign. The '*' sign indicates that the attribute is manifested by some schemes within the class.

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

Fig. 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.

activations corresponding to the OldHospital scene (Fig. 4(b)) are significantly stronger.

1) Visualization of Decoder Attention: In order to gain additional insights into the scene-specific features learned by our model, we further visualize the attentions $\{\mathbf{X}_i\}_1^N$ on the output of the positional decoder in two images from the *St. Mary's* scene. In addition, we measure and rank the decoder outputs by summing over the corresponding attention map. Indeed, the strongest response is obtained in the output corresponding to that scene (Fig. 5(d)). Interestingly, the activations related to the *ShopFacade* (Fig. 5(c)) scene are focused on the lower part of the input images, which typically includes the key features in images from this scene.

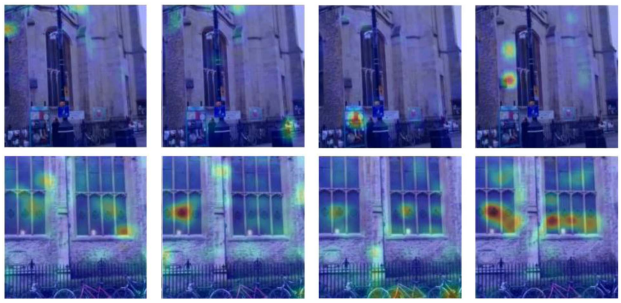
D. Ablation Study

To study the effect of different architectural design choices, we conducted multiple ablation experiments on the 7Scenes dataset (Tables VI–IX). In each experiment, we start with the

TABLE V
LOCALIZATION RESULTS WITH SINGLE-SCENE, MULTI-SCENE AND MULTI-DATASET LEARNING

APR Method	CambridgeLand. [m/deg]	7Scenes [m/deg]
Single-scene [14]	1.43/2.85	0.24/7.87
Multi-scene (Ours) [1]	1.28/2.73	0.18/7.28
Multi-dataset (Ours) [1]	1.50/**2.57**	0.22/6.78
c2f-Multi-scene (Ours)	**0.98**/3.10	**0.17**/7.28
c2f-Multi-dataset (Ours)	1.59/2.64	0.28/**6.44**

We report the average of median position/orientation errors in meters/degrees for the CambridgeLandmarks and 7Scenes datasets. Bold values indicate best performance.



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

Fig. 5. Translational Decoder attention visualization $\{\mathbf{X}_i\}_1^N$. Each activation relates to a different scene. The activations are due to two input images from the St Mary's scene. The activations of the corresponding scene are notably stronger.

TABLE VI
ABLATIONS OF THE CONVOLUTIONAL BACKBONE OF OUR MODEL, EVALUATED ON THE 7SCENES DATASET

Backbone	Position [meters]	Orientation [degrees]
Resnet50	0.19	8.60
EfficientNetB0	0.18	7.55
EfficientNetB1	0.17	7.26

We report the average of median position and orientation errors, across all scenes. The chosen model is highlighted in bold.

architecture used for our comparative analysis, without initialization, (Section IV-B) and modify a single algorithmic component / hyperparameter. Our ablation study focuses on three main aspects of our approach: (a) the derivation of activation maps (backbone and resolution) (b) the transformer architecture and (c) coarse-to-fine method (number of clusters). For the ablation of the number of clusters, we used the same backbone and Transformers configuration as in [1], to allow accurate comparisons with the proposed scheme. We computed the median position and orientation errors for each scene and reported the average between scenes.

Convolutional Backbone: We consider three convolutional encoders for our choice of backbone: ResNet50, EfficientNetB0 and EfficientNetB1. The results obtained with these backbones are shown in Table VI. The two EfficientNet variants achieve a better performance compared to the ResNet50 backbone, either due to overfitting (26 M parameters for ResNet50 compared to 5.3 M and 7.8 M for EfficientNetB0 and EfficientNetB1, respectively [47]) or a better learning capacity [47]. 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), at the expense of memory and runtime.

TABLE VII
ABLATIONS OF ACTIVATION MAPS EVALUATED ON THE 7SCENES DATASET

Resolution $\mathbf{A_x}/\mathbf{A_q}$	Position [meters]	Orientation [degrees]
$28 \times 28 \times 40/28 \times 28 \times 40$	0.22	7.42
$\mathbf{14 \times 14 \times 112/28 \times 28 \times 40}$	0.18	7.55
$7 \times 7 \times 320/28 \times 28 \times 40$	0.19	8.32
$14 \times 14 \times 112/14 \times 14 \times 112$	0.19	7.96

A_x and 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 chosen model is highlighted in bold.

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 the degree to which they are informative for position and orientation estimation. 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 or when segregating the sampling from two different resolutions. Table VII shows the results of different combinations. The best performance is obtained by providing a combination of coarse and fine activation maps for the position and orientation transformers, respectively. It follows that the chosen resolutions of $14 \times 14 \times 112/28 \times 28 \times 40$ are a sweet spot in this parameter space. In particular, for $\mathbf{A_x}$, the position estimation, the $14 \times 14 \times 112$ resolution is optimal as we tested for both the smaller ($7 \times 7 \times 320$) and larger ($28 \times 28 \times 40$) resolutions. As for $\mathbf{A_q}$, the orientation estimation, we tested for the smaller ($14 \times 14 \times 112$) activation map. We were unable to test for the higher $\mathbf{A_q}$ resolution ($56 \times 56 \times \cdot$) due to the associated memory constraints.

Transformer Architecture: The main hyperparameters of our Transformer architecture follow the standard choice. Thus, we further evaluate the sensitivity of our model performance to changing two main hyperparameters: the number of layers in the encoder and decoder components and the transformer dimension, C_d. The results are shown in Tables VIII and IX, respectively. All variants considered, in terms of layer number and transformer dimension, maintain the SOTA position and orientation accuracy, compared to other APR solutions (Table III). The accuracy improves with increasing the dimension of the Transformer, suggesting that larger models may achieve further improvement in localization accuracy. We note that regardless of the number of layers (Table VIII), our model outperforms other solutions (see Table III). We chose the standard 6-layer model for our operations and for comparing it with our previous

TABLE VIII
ABLATIONS OF THE NUMBER OF LAYERS IN THE TRANSFORMER ENCODERS AND DECODERS EVALUATED USING THE 7SCENES DATASET

Encoder/Decoder # Layers	Position [meters]	Orientation [degrees]
2	0.19	8.21
4	0.18	7.62
6	0.18	7.55
8	0.18	7.47

We report the average median position and orientation errors across all scenes. the chosen model is highlighted in bold.

TABLE IX
ABLATIONS OF THE TRANSFORMER'S LATENT DIMENSION, C_d, EVALUATED ON THE 7SCENES DATASET

Transformer Dimension	Position [meters]	Orientation [degrees]
64	0.19	8.32
128	0.18	7.32
256	0.18	7.55
512	0.18	7.21

We report the average of median position and orientation errors across all scenes. The chosen model is highlighted in bold.

TABLE X
ABLATIONS OF THE NUMBER OF POSITION AND ORIENTATION CLUSTERS, K_x AND K_q, RESPECTIVELY

K_x	K_q	Average [m/deg] Cambridge Landmarks	7Scenes
8	8	1.69/3.06	0.25/7.93
4	4	**1.27/3.01**	0.24/7.61
2	2	1.09/3.16	0.20/7.45
1	2	—	**0.18/7.55**

We report the average of the median position and orientation errors across all scenes for the 7Scenes and CambridgeLandmarks datasets. The chosen model is highlighted in bold, as we gave preference to the position error over the orientation error.

multi-scene architecture (to allow for a fair comparison), but also include a shallower model in runtime and memory analysis.

Number of Clusters: Our coarse-to-fine residual learning approach assumes the computation of k clusters (and centroids) for each scene, for orientation and position. To study the effect of this hyperparameter at small and large scene scales, we train our model with a varying number of clusters. Table X shows the results of this experiment for the CambridgeLandmarks and the 7Scenes datasets. We highlight the model chosen to report our main results by preferring the position over orientation errors. For mid- and large- scale scenes (CambridgeLandmarks), $k = 4$ yields the best results, with a similar performance obtained also for other choices (we report the performance of models trained for 400 epochs). When considering small-scale scenes (7scenes dataset), the choice of k has a significant effect on position estimation, with the best performance obtained using a single cluster. The results show the advantage of partitioning a scene into smaller subareas when learning to regress positions for mid- or large-scale scenes. When dealing with small-scale scenes, as in the 7Scenes dataset, this division does not improve position estimation. When considering orientation regression, the division improves performance, but, the results are less consistent and significant. We postulate that this can be improved by augmenting angle-based clustering with position information, where such an improvement is a direction for further research.

TABLE XI
RUNTIME (IN MS) AND MEMORY FOOTPRINT (IN MB) AS A FUNCTION OF THE NUMBER OF LEARNED SCENES

Num. Scenes / Num. Layers	Runtime [ms] 2	Runtime [ms] 6	Memory [MB] 2	Memory [MB] 6
1	18.8	34.6	40.8	74.6
4	18.8	35	40.8	74.6
7	19.2	35.2	40.8	74.6
10	19.2	35.2	40.8	74.6
100	19.6	35.4	41.0	74.8
500	21.0	41.0	41.8	75.6
1000	27.0	58.6	42.8	76.7

The results for two instances of our model, using two and six layers in all encoders and decoders.

Scalability: Runtime and memory footprint are two of the main advantages of single-scene APRs. However, in order to cover a site with N scenes, N models need to be stored and selected-from during inference time. Thus, encoding a site of 1000 scenes with PoseNet models [7] of size 30 MB each, requires $1000 \times 30\text{MB} = 3$ GB. For a given number of scenes, the required memory and run-time of the proposed scheme are invariant of the network's weights and depend only on its architecture. To evaluate the run-time and memory footprint, it is enough to instantiate models for a varying number of scenes and measure their model size and the runtime of their forward pass (without training). For our ablation study, we consider hypothetical datasets with an order of magnitude of $10^1 - 10^3$ scenes. The results are shown in Table XI, where we compare two variants of our model: the architecture used for the comparative analysis (Section IV-B), with six layers for each encoder and decoder, and a shallower model with two layers per encoder/decoder, for which we report similar performance as part of our ablation study (Section IV-C). The memory footprint of our model remains relatively constant, where increasing from 4 to 1000 scenes adds only 2 MB. In addition, both variants require less than 80 MB before any optimization. The runtime of the model remains constant in the range of 4-100 scenes and increases by $\sim \times 1.5$ per 1000 scenes. Assuming a constant scene selection time, our model is $\times 2 - \times 5$ slower than a single-scene APR. This can be expected due to the run-time complexity of the MHA operation, which is quadratic with respect to the sequence length (number of scenes). However, a significant acceleration can be achieved with recent linear-time MHA formulations [48] and other optimization methods [49]. We demonstrate this in the next section.

Runtime and Memory: We evaluate and compare the runtime and model size, and the total storage required, for our method and other localization schemes, in Tables XII and XIII, respectively. MS-Transformer and the proposed c2f-MS-Transformer were implemented using the EfficientNet-B0 backbone and have 6 layer encoders and decoders. The runtime of Duong et al. [50], Brachmann et al. [51] and DSAC [26] are cited from [50], while the DSAC* timing is cited from [42]. All timing measurements

TABLE XII
RUNTIME AND MODEL SIZE COMPARISON OF OUR APPROACH AND REPRESENTATIVE LOCALIZATION METHODS

Network	Time [ms]	Model Size [MB]	Device
DSAC*[42]	30	30	2080TI
DSAC*[42]	75	30	K80
DSAC[26]	1500	–	1080
Brachmann[51]	1000	–	1080
Duong[50]	50	–	1080
PoseNet[7]	7.8	26.7	1080
Ours (6 layers)	35	75	1080
Ours (2 layers)	19	42	1080
Ours-TRT (6 layers)	**6.1**	**48.2**	2080TI

The GeForce 1080 and tesla K80 GPUs are of similar computing power.
Bold values indicate the best configuration of our model in terms of runtime and model size at inference time.

TABLE XIII
THE ORDER OF MAGNITUDE OF THE STORAGE REQUIRED BY DIFFERENT LOCALIZATION SCHEMES FOR 10 AND 100 SCENES (ASSUMING 1000 IMAGES PER SCENE)

Method	Scenes# 10	100
SbP [3]	TB	TB
SCR[26]	GB	GB
SCR [52]	MB	GB
RPR ([23])	GB	GB
Single Scene APR [7]	MB	GB
Multi Scene APR (Ours)	**MB**	**MB**

For RPRs we assume single image encoding weighs of 5kb.
Bold values indicate the lowest requirements in terms of data size and model size required at inference time.

were made using GTX1080 and Tesla K80 GPUs that are of similar processing speed. We also applied NVIDIA's TensorRT[2] (TRT) to optimize our model size and runtime, using TRT 8.0.3.4 with CUDANN 8.2 and default settings. With this optimization, our model runs in 6.1ms ($\times$5.77 speedup) and has a model size of 48.2MB (27% reduction), demonstrating its potential for deployment in real-time applications on low-end devices. The optimized version is $\times$ 4.9 faster than the accelerated version of DSAC* [42]. The optimization process was performed with the GTX2080TI GPU.

V. CONCLUSION

In this work, we propose a novel transformer-based approach for multiscene absolute pose regression. Using two Transformer Encoders, self-attention is applied separately to positional and orientational image cues. Thus, aggregating the activation maps computed by the backbone CNN in a task-adaptive manner. Our formulation allows agglomerating non-scene-specific information in the backbone CNN and Transformer Encoders. Scene-specific information is encoded by a Transformer-Decoder and is queried per scene. Furthermore, the proposed mixed classification-regression architecture introduces a novel coarse-to-fine scheme by incorporating scene clustering. We demonstrate that our approach provides improved state-of-the-art localization accuracy for both single-scene and multiscene absolute regression approaches, across outdoor and indoor datasets.

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

REFERENCES

[1] Y. Shavit, R. Ferens, and Y. Keller, "Learning multi-scene absolute pose regression with transformers," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 2713–2722.
[2] T. Sattler, B. Leibe, and L. Kobbelt, "Efficient & effective prioritized matching for large-scale image-based localization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 9, pp. 1744–1756, Sep. 2017.
[3] H. Taira et al., "InLoc: Indoor visual localization with dense matching and view synthesis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 4, pp. 1293–1307, Apr. 2021.
[4] P. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12708–12717.
[5] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.
[6] T. Sattler, Q. Zhou, M. Pollefeys, and L. Leal-Taixé, "Understanding the limitations of CNN-based absolute camera pose regression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3297–3307.
[7] A. Kendall, M. Grimes, and R. Cipolla, "PoseNet: A convolutional network for real-time 6-DoF camera relocalization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2938–2946.
[8] T. Naseer and W. Burgard, "Deep regression for monocular camera-based 6-DoF global localization in outdoor environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 1525–1530.
[9] F. Walch, C. Hazirbas, L. Leal-Taixé, T. Sattler, S. Hilsenbeck, and D. Cremers, "Image-based localization using LSTMs for structured feature correlation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 627–637.
[10] Y. Shavit and R. Ferens, "Do we really need scene-specific pose encoders?," in *Proc. 25th Int. Conf. Pattern Recognit.*, 2021, pp. 3186–3192.
[11] B. Wang, C. Chen, C. X. Lu, P. Zhao, N. Trigoni, and A. Markham, "AtLoc: Attention guided camera localization," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, pp. 10393–10401.
[12] M. Cai, C. Shen, and I. Reid, "A hybrid probabilistic model for camera relocalization," in *Proc. Brit. Mach. Vis. Conf.*, 2018, pp. 1–12.
[13] A. Kendall and R. Cipolla, "Modelling uncertainty in deep learning for camera relocalization," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 4762–4769.
[14] A. Kendall and R. Cipolla, "Geometric loss functions for camera pose regression with deep learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6555–6564.
[15] Y. Shavit and R. Ferens, "Introduction to camera pose estimation with deep learning," 2019, *arXiv:1907.05272*.
[16] H. Blanton, C. Greenwell, S. Workman, and N. Jacobs, "Extending absolute pose regression to multiple scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 38–39.
[17] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 213–229.
[18] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Representations*, 2021.
[19] L. Li and H.-T. Lin, "Ordinal regression by extended binary classification," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2007, vol. 19, pp. 865–872.
[20] W. Cao, V. Mirjalili, and S. Raschka, "Rank consistent ordinal regression for neural networks with application to age estimation," *Pattern Recognit. Lett.*, vol. 140, pp. 325–331, 2020.
[21] G. Jenkinson, K. Khezeli, G. R. Oliver, J. Kalantari, and E. W. Klee, "Universally rank consistent ordinal regression in neural networks," 2021, *arXiv:2110.07470*.
[22] M. Dusmanu et al., "D2-Net: A trainable CNN for joint description and detection of local features," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8084–8093.

[23] V. Balntas, S. Li, and V. Prisacariu, "RelocNet: Continuous metric learning relocalisation using neural nets," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 751–767.
[24] Z. Laskar, I. Melekhov, S. Kalia, and J. Kannala, "Camera relocalization by computing pairwise relative poses using convolutional neural network," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2017, pp. 920–929.
[25] M. Ding, Z. Wang, J. Sun, J. Shi, and P. Luo, "CamNet: Coarse-to-fine retrieval for camera re-localization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 2871–2880.
[26] E. Brachmann et al., "DSAC—Differentiable RANSAC for camera localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2492–2500.
[27] E. Brachmann and C. Rother, "Learning less is more - 6D camera localization via 3D surface regression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4654–4662.
[28] S. Saha, G. Varma, and C. Jawahar, "Improved visual relocalization by discovering anchor points," in *Proc. Brit. Mach. Vis. Conf.*, 2018.
[29] A. Valada, N. Radwan, and W. Burgard, "Deep auxiliary learning for visual localization and odometry," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 6939–6946.
[30] N. Radwan, A. Valada, and W. Burgard, "VLocNet++: Deep multitask learning for semantic visual localization and odometry," *IEEE Robot. Automat. Lett.*, vol. 3, no. 4, pp. 4407–4414, Oct. 2018.
[31] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz, "Geometry-aware learning of maps for camera localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2616–2625.
[32] E. Brachmann and C. Rother, "Expert sample consensus applied to camera re-localization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7525–7534.
[33] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learn. Representations*, 2015.
[34] A. Vaswani et al., "Attention is all you need," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30, pp. 5998–6008.
[35] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2019, pp. 4171–4186.
[36] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, 2019, Art. no. 9.
[37] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.
[38] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982.
[39] A. Paszke, "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2019, vol. 32, pp. 8026–8037.
[40] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn. Res.*, 2019, vol. 97, pp. 6105–6114.
[41] B. Glocker, S. Izadi, J. Shotton, and A. Criminisi, "Real-time RGB-D camera relocalization," in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, 2013, pp. 173–179.
[42] E. Brachmann and C. Rother, "Visual camera re-localization from RGB and RGB-D images using DSAC," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5847–5865, Sep. 2022.
[43] F. Xue, X. Wu, S. Cai, and J. Wang, "Learning multi-view camera relocalization with graph neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11372–11381.
[44] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla, "24/7 place recognition by view synthesis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1808–1817.
[45] Q. Zhou, T. Sattler, M. Pollefeys, and L. Leal-Taixe, "To learn or not to learn: Visual localization from essential matrices," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 3319–3326.
[46] F. Xue, X. Wang, Z. Yan, Q. Wang, J. Wang, and H. Zha, "Local supports global: Deep camera relocalization with sequence enhancement," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2841–2850.
[47] M. Tan and Q. Le, " EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.
[48] K. M. Choromanski et al., "Rethinking attention with performers," in *Proc. Int. Conf. Learn. Representations*, 2020.
[49] H. Vanholder, "Efficient inference with tensorrt," in *Proc. GPU Technol. Conf.*, 2016, vol. 1, no. 2.
[50] N.-D. Duong, A. Kacete, C. Soladie, P.-Y. Richard, and J. Royan, "Accurate sparse feature regression forest learning for real-time camera relocalization," in *Proc. Int. Conf. 3D Vis.*, 2018, pp. 643–652.
[51] E. Brachmann, F. Michel, A. Krull, M. Y. Yang, S. Gumhold, and C. Rother, "Uncertainty-driven 6D pose estimation of objects and scenes from a single RGB image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3364–3372.
[52] M. O. Turkoglu, E. Brachmann, K. Schindler, G. J. Brostow, and A. Monszpart, "Visual camera re-localization using graph neural networks and relative pose supervision," in *Proc. Int. Conf. 3D Vis.*, 2021, pp. 145–155.

Yoli Shavit received the BSc degree in computer science and in life science from Tel Aviv University, Tel Aviv, Israel, the MSc degree in bioinformatics and systems biology from Imperial College London, London, U.K., and the PhD degree in computer science from the University of Cambridge, Cambridge, U.K. She is currently a postdoctoral researcher with Bar-Ilan University, Ramat Gan, Israel. Her current research focuses on deep learning methods for camera localization with recent publications at CVPR, ICCV, ECCV, and NIPS. She was the recipient of the Cambridge International Scholarship, and her thesis was nominated for the best thesis award in the U.K.

Ron Ferens received the undergraduation degree in electrical engineering from the Ben-Gurion University of the Negev, Beersheba, Israel, in 2006, and the master's degree in electrical engineering from Tel Aviv University, Tel Aviv, Israel, in 2010. He is currently working toward the PhD degree in electrical and electronics engineering with Bar-Ilan University. He is a proficient expert in the fields of computer vision and AI. He has effectively led research teams with renowned institutions including Huawei, Magic Leap, and Intel, with a focused exploration into domains such as 3D vision, eye-tracking, pose regression, and augmented reality. His dedication extends to both his academic pursuits and professional endeavors, firmly centered on the realms of computer vision and deep learning.

Yosi Keller received the undergraduation degree in electrical engineering from the Technion-Israel Institute of Technology, Haifa, Israel, in 1994, and the master's and Doctoral (*summa cum laude*) degrees in electrical engineering from Tel Aviv University, Tel Aviv, Israel, in 1998 and 2003, respectively. Following his PhD degree, he was a Gibbs Assistant professor with Yale University, New Haven, CT, USA, from 2003 to 2006. He is currently an associate professor with the Faculty of Engineering, Bar Ilan University, Ramat-Gan, Israel. His academic research interests include computer vision and deep learning.