Leveraging Equivariant Features for Absolute Pose Regression

Mohamed Adel Musallam  
mohamed.ali@uni.lu

Vincent Gaudillièere  
v Vincent.gaudilliere@uni.lu

Miguel Ortiz del Castillo  
miguel.ortizdelcastillo@uni.lu

Kassem Al Ismaeil  
kassem.alismaeil@gmail.com

Djamila Aouada  
djamila.aouada@uni.lu

Interdisciplinary Center for Security, Reliability and Trust (SnT)  
University of Luxembourg, Luxembourg

Abstract

While end-to-end approaches have achieved state-of-the-art performance in many perception tasks, they are not yet able to compete with 3D geometry-based methods in pose estimation. Moreover, absolute pose regression has been shown to be more related to image retrieval. As a result, we hypothesize that the statistical features learned by classical Convolutional Neural Networks do not carry enough geometric information to reliably solve this inherently geometric task. In this paper, we demonstrate how a translation and rotation equivariant Convolutional Neural Network directly induces representations of camera motions into the feature space. We then show that this geometric property allows for implicitly augmenting the training data under a whole group of image plane-preserving transformations. Therefore, we argue that directly learning equivariant features is preferable than learning data-intensive intermediate representations. Comprehensive experimental validation demonstrates that our lightweight model outperforms existing ones on standard datasets.¹

1. Introduction

In computer vision, camera pose estimation, and its reference frame inverse, i.e., object pose estimation, have been extensively studied over the last decades [38, 42, 54].

Traditionally, pose estimation has been addressed using 3D geometry. In practice, a set of 2D-3D feature correspondences is generated, then statistically leveraged to recover the camera pose [18, 39, 49, 64]. More recently, direct Absolute Pose Regression (APR) approaches have been introduced, drawing upon early successes of deep learning [1].

¹This work was funded by the Luxembourg National Research Fund (FNR), under the project reference BRIDGES2020/IS/14755859/MEET-A/Aouada, and by LMO (https://www.lmo.space).

Figure 1. Illustration of our approach - Our method adopts a translation and rotation-equivariant convolutional neural network to extract geometry-aware features that directly encode camera planar motions $R, t$. While camera moves, equivariance of the proposed feature extractor $F$ leads to explicit image $(\phi^{(i)}_{R,t})$ and feature $(\phi^{(F)}_{R,t})$ changes. This property is leveraged to propose a more efficient solution to the absolute pose regression problem.

These methods consist in directly mapping an image to its pose using a suitably trained Convolutional Neural Network (CNN). Therefore, end-to-end trainable methods have the advantage of providing fully differentiable results, enabling the optimization of all parameters in a comprehensive manner. Moreover, predictions are achieved at a steady speed and power consumption, whereas RANdom SAmple Consensus (RANSAC)-based methods [18] are less predictable...
and likely to suffer from an efficiency drop when the inlier rate is low. However, state-of-the-art APR methods have been proven theoretically and shown experimentally to have a lower accuracy compared to 3D structure-based approaches [50]. Indeed, the former are more closely related to image retrieval than to 3D structure [50]. The questions we ask in this work are: Why do current APR methods fall short in accuracy? How can they reach their full potential? Our hypothesis is that there is a lack of exploitation of the geometric properties of data. This happens typically at the level of the feature extraction layers commonly used in classical deep learning approaches. Specifically, we posit that in the case of APR and pose estimation, having a representation which is equivariant to the group of rigid motions, i.e., rotations and translations in 3D, may be an effective way to boost the network performance. This should play the role of an implicit data augmentation by means of group equivariance, and in turn alleviate the need for an explicit data augmentation for training.

Indeed, recently, there has been a growing interest in designing more geometric models that are equivariant to groups of such transformations. These approaches leverage theoretical contributions which from group theory, representation theory, harmonic analysis and fundamental deep learning [8–10, 12, 21, 22, 61]. More specifically, group-equivariant neural networks, or Group-equivariant CNNs (G-CNNs), are part of the broader and promising field of geometric deep learning [4], that aims to exploit any underlying geometric relationship that can exist within the data. In particular, the special Euclidean groups in 2 and 3 dimensions, denoted as $SE(2)$ and $SE(3)$ and encompassing respective rigid motions, are of particular interest in 3D computer vision [13, 61].

Despite the conceptual advances they represent, to the best of our knowledge, the use of deep equivariant features in the APR context is still considerably unexplored. This paper proposes, for the first time, to investigate and justify the use of deep equivariant features for solving APR (see Figure 1).

**Contributions.** Our contributions are summarized below:

1. A formulation of how an equivariant CNN induces representations of planar camera motions, lying in $SE(2)$, directly into the feature space. (Section 4.1)
2. An intuitive explanation is provided as to how $SE(2)$-equivariant features can be leveraged to recover any camera motion lying in $SE(3)$. (Section 4.2)
3. A lightweight equivariant pose regression model, referred to as E-PoseNet, is introduced. (Section 5)
4. Extensive experimental evaluation of E-PoseNet showing its competitive performance as compared to existing APR methods on standard datasets. (Section 6)

**Paper organization.** An overview of state-of-the-art APR methods and current exploitations of deep equivariant features is given in Section 2. Section 3 presents the formal definition of equivariance along with the formulation of APR. The theoretical justification as to how $SE(2)$-equivariant features can explicitly encode planar camera motions is presented in Section 4, whereas the full pose regression pipeline is introduced in Section 5. An extensive experimental validation is given in Section 6 along with a discussion of limitations. Section 7 concludes the paper.

## 2. Related Work

The goal of this paper is to exploit the power of equivariant features in the context of APR. Therefore, we split this section into: (1) a review of the relevant literature on APR, and (2) an overview of recent deep equivariant feature extraction methods applied to computer vision problems.

**Absolute Pose Regression.** Since the rise of deep learning and CNNs in the early 2010’s, many works have explored the application of CNNs for APR. This began with the introduction of PoseNet by Kendal et al. [35], who used the GoogLeNet model [55] as a feature extraction backbone coupled with a regression head to estimate the translation and rotation vectors. Most of the subsequent improvements lie in changes in the feature extraction architecture [3, 35, 62], modified objective functions [32, 43, 57], and additional intermediate representations [25, 27].

In [50], Sattler et al. provide an in-depth analysis of existing works on APR [6, 34, 44, 62, 63]. In particular, they show that structure-based and image retrieval methods are more accurate than APR. Moreover, they demonstrate that APR algorithms do not explicitly leverage knowledge about projective geometry. Instead, they learn a mapping between image content and camera poses directly from the data, and in the form of a set of base poses such that all training samples can be expressed as a linear combination of these reference entities. Wang et al. [60] proposed an approach to integrate dense correspondence-based intermediate geometric representations within an end-to-end trainable pipeline. However, this method still relies on classical (non-equivariant) features, and thereby requires a significant amount of data for generalization. Furthermore, methods such as [25, 27, 60] propose to learn intermediate representations that are indirectly equivariant, such as segmentation masks, object detections, and depth or normal maps. However, this comes at the cost of parameter redundancy. This core observation suggests that directly learning equivariant features may be a valuable direction to improve the accuracy of pose estimation while reducing the number of model parameters.

**Deep Equivariant Features.** There is a rich history in computer vision on the design of hand-crafted equivariant features (e.g., Scale-Invariant Feature Transform (SIFT) [41], Oriented filters [65], Steerable filters [19],
Rotation-equivariant Fields of Experts (R-FoE) [51], Lie groups-based filters [17,46]. In the deep learning literature, convolutional layers [37] have been proven to be equivariant to image shifting, while max-pooling layers are only invariant to small shifts of the input image [20]. Although convolutional layers are inherently equivariant to translation, there is a significant amount of spatial information regarding the inputs that is not encoded by CNNs in a precise fashion [28,31]. More specifically, local and global information regarding the inputs that is not encoded by CNNs in a precise fashion [28, 31].

A recent investigation shows that many neurons in CNNs poolings, if added to CNNs, render translation information precise fashion [28, 31]. More specifically, local and global information regarding the inputs that is not encoded by CNNs in a precise fashion [28, 31].

A recent investigation shows that many neurons in CNNs learn slightly transformed (e.g., rotated) versions of the same basic feature [47]. These are especially common in early vision, e.g., in curve detectors, high-low frequency detectors, and line detectors.

There have been attempts to extend the G-CNNs to wider groups of transformations. In [5,48], Mallat et al. extended CNNs to be equivariant to SE(2) using scattering transform with predefined wavelets. In [2,29], Bekkers et al. also extended CNNs to be equivariant to the SE(2) group via B-splines. In [9], Cohen et al. proposed group convolutions network equivariant to the p4m discrete group via 90° rotations and flips, where they demonstrated the effectiveness of group convolutions for classification task.

More recently, the use of equivariant features has been investigated for solving various computer vision tasks such as 3D point cloud analysis [8], aerial object detection [22] and 2D tracking [21]. In [14], Esteves et al. proposed to use projection and embedding from 2D images into a spherical CNN latent space to estimate the relative orientations of the object. Similarly, Zhang et al. proposed to use spherical CNN for learning camera pose estimation in omnidirectional localization [66]. However, to the best of our knowledge, equivariant features have not yet been explicitly leveraged in the context of APR for single 2D input image, which is the very focus of this paper.

3. Preliminaries

This section provides the necessary mathematical background. First, we introduce the notions of invariant and equivariant features. Then, we present the general framework for APR, and finally show the added value of relying on equivariant features in this context.

Notation. The following notation will be adopted: vectors and column images are denoted by boldface lowercase letters x, matrices by uppercase letters X, scalars by italic letters x or X, functions as X and spaces as X. The special orthogonal group, the Euclidean group, and the special Euclidean group, of dimension n, are denoted as SO(n), E(n) and SE(n), respectively.

Invariant and Equivariant Features.

Given an image x, captured by a camera, an APR method \( \mathcal{P} \) predicts the 6-Degrees-of-Freedom (6-DoF) pose, i.e., position and orientation, of the camera with respect to its environment.

Let us denote by \( I \subset \mathbb{R}^m \) the linear space of vectorized m-dimensional images (or image regions), and by \( F \subset \mathbb{R}^n \) the latent space of features – with dimension n. Considering a CNN-based feature extraction function \( F \), we write:

\[
F : I \rightarrow F
\]

\[
x \mapsto F(x).
\]

Given \( G \), a generic group of transformations and \( g \), an element of \( G \), we denote by \( \phi_g^{(i)} \) and \( \phi_g^{(p)} \) the actions of \( g \) into the image and feature spaces, respectively.

**Definition 1** \( F \) is invariant to \( G \) if and only if

\[
\forall g \in G, \forall x \in I, \quad F(\phi_g^{(i)} x) = F(x).
\]

**Definition 2** \( F \) is equivariant to \( G \) if and only if

\[
\forall g \in G, \forall x \in I, \quad F(\phi_g^{(i)} x) = \phi_g^{(p)} F(x).
\]

Note that invariance can be seen as a special case of equivariance where \( \phi_g^{(p)} = I \), the identity mapping, \( \forall g \in G \).

**Equivariant Features for APR.** Sattler et al. proposed the following formulation for the pose function \( \mathcal{P} \) [50]:

\[
\mathcal{P}(x) = b + P \cdot \mathcal{E}(F(x)),
\]

where the feature extractor \( F \) is first applied to the image x followed by a non-linear embedding of the features \( \mathcal{E} \) lifting them to a higher-dimensional space. Then, a linear projection into the space of camera poses, represented by a matrix \( P \), is applied. Finally, a bias term \( b \) is added.

As presented in Section 2, the work in [50] demonstrated that classical APR is more closely related to pose approximation via image retrieval than to accurate pose estimation leveraging the 3D structure, thus the accuracy gap.

Our hypothesis is that this is likely due the lack of geometric information carried by classical CNN features. Indeed, the perceptive power of classical CNNs can most often be considered as a statistical phenomenon, whereas pose estimation is a geometrical problem.

In this work, we thus propose to replace the classical convolutional layers of the feature extractor \( F \) by their group-equivariant counterparts [9], then to assess how this affects both the accuracy and data efficiency of the model. Therefore, assuming that \( F \) is equivariant to \( G \), i.e., verifies **Definition 2**, and applying the transformation \( \phi_g^{(i)} \) to the image x, the pose regression function \( \mathcal{P} \) in (3) becomes:

\[
\mathcal{P}(\phi_g^{(i)} x) = b + P \cdot \mathcal{E} \left( \phi_g^{(p)} v \right),
\]
Similarly, translating the camera center along \( \mathbf{u} \) as infinite define image intensity values \( \mathbf{Z} = (X, Y, Z = Z_0) \). Rays of lights are then projected to the camera center. Intersections of projected rays with the image plane (considered as infinite) define image intensity values \( l(x, y) \). Camera motions are restricted to \( \text{SE}(2) \), i.e. translation along \( \mathbf{u}_x, \mathbf{u}_y \) and rotation around \( \mathbf{u}_z \) (characterized by roll angle \( \theta \)).

\[
\mathbf{v} = \mathcal{F}(\mathbf{x})
\]

This suggests that any action of \( \mathfrak{G} \) on the image has a direct effect in the latent space and that, in particular, camera motion transformations of images, i.e. changes in camera pose, explicitly induce actions on the feature vector \( \mathbf{v} \), and implicitly on the regressed pose. Considering \( \mathfrak{G} \) as \( \text{SE}(2) \) or \( \text{SE}(3) \), we posit that such equivariant features will help improve the performance of APR.

4. Pose from \text{SE}(2)-Equivariant Features

We consider a piecewise planar scene, where the scene planes are parallel to the image plane. We then consider camera motions that locally preserve the latter.

4.1. \text{SE}(2)-Equivariant Features

We herein restrict camera motions to those of the \( \text{SE}(2) \) group, i.e., planar translations and rotations within the image plane (Figure 2). With that, we analyse the effects of planar camera motions on the image and feature spaces, assuming that the feature extractor \( \mathcal{F} \) is equivariant to \( \text{SE}(2) \).

Effects of Camera Planar Motions on Images. Following the notations introduced in Figure 2, rotating the camera with a roll angle \( \theta \) is equivalent to rotating the scene around \( \mathbf{u}_z \) (camera viewing direction) with angle \( -\theta \) \cite{30}.

Similarly, translating the camera center along \( \mathbf{u}_x \) and \( \mathbf{u}_y \) is equivalent to translating the scene in the opposite direction. Let us denote any rigid motion of the camera along its image plane (i.e., in \( \text{SE}(2) \)) by \( \mathbf{R}, \mathbf{t} \), where \( \mathbf{t} \) is a planar translation and \( \mathbf{R} \) a planar rotation. The effect of this motion on any point \( \mathbf{p} \) of the scene is obtained by applying \( -\mathbf{t} \) then \( \mathbf{R}^\top \) such that \( \mathbf{p}' = \mathbf{R}^\top(\mathbf{p} - \mathbf{t}) \).

In 3D, considering \( \mathbf{t} = (T_X, T_Y, 0)^\top \), we have

\[
\mathbf{p}' = \begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X - T_X \\ Y - T_Y \\ Z \end{pmatrix}.
\]

Following classical projection rules \cite{23}, image coordinates \( (x, y) \) are then given by \( x = f \frac{X}{Z_0} \) and \( y = f \frac{Y}{Z_0} \), where \( f \) is the distance from the camera center to the image plane, and \( Z_0 \) is the distance to the scene plane.

Multiplying (5) by \( \frac{f}{Z_0} \), then restricting the coordinates to the first two ones gives:

\[
\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} x - t_x \\ y - t_y \end{pmatrix},
\]

where \( t_x = f \frac{T_X}{Z_0} \) and \( t_y = f \frac{T_Y}{Z_0} \). By denoting \( \mathbf{R}^{(2)} \) the 2D rotation matrix of angle \( \theta \) and \( \mathbf{t}^{(2)} = (t_x, t_y)^\top \), the effect of any planar camera motion \( \mathbf{R}, \mathbf{t} \) on any point \( \mathbf{p}^{(2)} \) of the image is thus given by \( \mathbf{p}^{(2)}' = (\mathbf{R}^{(2)\top} (\mathbf{p}^{(2)} - \mathbf{t}^{(2)})) \), where \( \mathbf{p}^{(2)} \) is the image of \( \mathbf{p}^{(2)} \) under the transformation.

We finally denote \( \phi_{\mathbf{R}, \mathbf{t}}^{(2)} \) the effect of the camera motion on an image \( \mathbf{x}_1 \), resulting in another image \( \mathbf{x}_2 \), i.e., \( \mathbf{x}_2 = \phi_{\mathbf{R}, \mathbf{t}}^{(2)}(\mathbf{x}_1) \).

In what follows, we prove that the image transformation due to planar motions of the camera commute with the projection operator \( \mathcal{P} \) (Figure 3). Indeed, applying a planar motion \( \mathbf{R}_1, \mathbf{t}_1 \) followed by a second one \( \mathbf{R}_2, \mathbf{t}_2 \) to the camera, has the following effect\footnote{Please refer to the supplementary material for further details}:

\[
\mathbf{p}' = \mathbf{R}_3^\top(\mathbf{p} - \mathbf{t}_3),
\]

where \( \mathbf{R}_3 = \mathbf{R}_2 \mathbf{R}_1 \) and \( \mathbf{t}_3 = \mathbf{t}_1 + \mathbf{R}_2 \mathbf{t}_2 \).

Similarly, one can easily observe that combining two camera motions has a similar effect on any point \( \mathbf{p}^{(2)} \) of the image such that \( \mathbf{p}^{(2)}' = \mathbf{R}_3^{(2)\top} (\mathbf{p}^{(2)} - \mathbf{t}_3^{(2)}) \) where \( \mathbf{R}_3^{(2)} = \mathbf{R}_2^{(2)} \mathbf{R}_1^{(2)} \).
By replacing the translation group equivariance of classical CNNs by equivariance to the rotation group, we herein discuss how these features, which are equivariant to planar camera motions, i.e., in SE(2), are leveraged for general pose regression in SE(3).

Indeed, SE(2)-equivariant features extracted with \( \mathcal{F} \), are now to be mapped to camera poses in SE(3) via \( \gamma : \mathcal{P} \times \mathcal{E} \). This is the last step towards finding \( \mathcal{P}(x) \), as defined in equation (3).

The SE(3) group may be written as a semidirect product \( \mathbb{R}^3 \ltimes \mathbb{S}(3) \), and similarly for \( \mathbb{R}(2) \). We rely on the observation that the quotient space \( \mathbb{S}(3) / \mathbb{S}(2) \cong \mathbb{S}(2) \) is the sphere in 3D, where \( \cong \) denotes a homeomorphism.

For every point on \( \mathbb{S}(2) \), it is possible to move to another point via a rotation. \( \mathbb{S}(2) \) is consequently a homogeneous space for \( \mathbb{S}(3) \), and \( \mathbb{S}(3) \) can be seen as a bundle of elements of \( \mathbb{S}(1) \), i.e., planar circles on \( \mathbb{S}(2) \), for which a continuous mapping \( \gamma^* \) exists [11]. These mappings, \( \gamma^* \), directly relate to \( \gamma \) by composing with translations. Figure 4 illustrates how the planar rotations around a fixed axis (Figure 4(a)) can be viewed as local patches on the sphere in 3D (Figure 4(b)); thus, relating to rotations in 3D, and finally how this may be generalized to full rigid motions by translation as shown in Figure 4(c). The mapping \( \gamma \) is learned as part of the end-to-end APR.

This section gives an overview of our proposed equivariant pose regression model, E-PoseNet. To be able to assess how explicitly encoding pose information into the feature space can result in a more accurate

\[ R_2^{(2)} R_1^{(2)} \text{, and } t_3^{(2)} = t_1^{(2)} + R_1^{(2)} t_2^{(2)}. \]
To decrease the computational cost, we discretize the rotation, and non-linearities, to build an equivariant ResNet18. Specifically, we use the `e2cnn` extractor $SE(2)$-Equivariant Backbone.

Connected MLP head.

To regress the position and orientation of the camera, each branch consists of an independent fully-connected Multilayer Perceptrons (MLP) for lifting the features to a higher dimensional space, followed by two branches for separately regressing the position and orientation of the camera. Each branch consists of an independent fully-connected MLP head.

**Network Architecture.** *E-PoseNet* is composed of a roto-translation equivariant ResNet18 backbone, two fully connected Multilayer Perceptrons (MLP) for lifting the features to a higher dimensional space, followed by two branches for separately regressing the position and orientation of the camera. Each branch consists of an independent fully-connected MLP head.

**SE(2)-Equivariant Backbone.** Our backbone, i.e., feature extractor $F$, is an SE(2) roto-translation equivariant version of ResNet. Specifically, we use the `e2cnn` implementation for $E(2)$-equivariant convolution, pooling, normalization, and non-linearities, to build an equivariant ResNet18. To decrease the computational cost, we discretize the SE(2) group making the model only equivariant to the $(\mathbb{R}^2,+)$ × $C_N$ group, meaning all translations in $\mathbb{R}^2$ and rotations by angles multiple of $\frac{\pi}{N}$. Extracted features are now rotation-equivariant feature maps $V$ with the size $(K \times N \times H \times W)$, where $K$ is the number of channels, $N$ the number of feature orientations (for our model we used $N = 8$), and $H, W$ respectively, the height and width.

In addition to classical translation information, obtained features thus encode rotation information that can enhance the pose regression. Furthermore, equivariance to broader transformations constrains the network in a way that can aid generalization, especially due to the weights shared under image rotations [9]. Finally, this rotation-equivariant ResNet shows a significant reduction in model size, about $1/N$ parameters compared to the regular ResNet architecture, to obtain the same feature size. Indeed, the size of classical feature maps is in the form $(K \times H \times W)$.

**Loss Function.** To regress camera poses, we use the loss function introduced in [34], and defined as:

$$L_P = L_t \exp (-s_t) + s_t + L_R \exp (-s_R) + s_R,$$

where the position loss $L_t = \| t_0 - t \|_2$, and the orientation loss $L_R = \| q_0 - q \|_2$, are computed from predicted $(q, t)$ and groundtruth $(q_0, t_0)$ camera poses, considering the quaternion representation for orientations. $s_t, s_R$ are learned parameters.

**6. Experiments and Analysis**

The proposed method aims to improve APR accuracy by utilizing an equivariant feature extraction backbone able to learn geometry-aware feature maps. We first show the effect of SE(2)-equivariant models on rotated feature maps using samples from the T-Less dataset [26]. Then, we benchmark our proposed *E-PoseNet* on two datasets for both indoor and outdoor camera localization.

**Equivariance analysis on T-Less.** In this study, we use a sequence of ‘object 5’ from the T-Less training dataset [26]. With only one textureless symmetric object present in the scene and undergoing continuous rotations, this sequence represents an ideal case for testing the impact of the different rotation parametrization and channeling. To assess the effect of equivariance, our backbone is made of 10 convolution layers with kernel size equal to 2, ELU non-linearity and Max Pooling downsampling every two layers with kernel size equal to 2. Different degrees of equivariance were tested, namely, Classical CNN “translation equivariant”, Equivariant 90° ($N=4$), Equivariant 45° ($N=8$), Equivariant 18° ($N=20$), Equivariant 10° ($N=36$), and finally the Equivariant $SO(2)$. Equivariant models are generated based on `e2cnn` implementation [61]. We trained the model on one sequence only, without any data augmentation and for 100 epochs. Number of parameters, optimizer, learning rate and random seeds were fixed.
Figure 6. **Extracted feature maps** - difference between: (b) equivariant CNN and (c) classical CNN. The used samples (a) are from the T-LESS Dataset [26].

Figure 7. **Equivariant models comparison** - On sequence of ‘object 5’ from the T-less train dataset [26], we can see the increase in the model accuracy when increasing the equivariant group.

Figure 7 reports the proportion of samples for which the predicted pose error is below 10cm, 10°. We observed that increasing the level of equivariance, *i.e.*, decreasing the discrete sampling angle, leads to increasing the performance of the pose estimation model. Furthermore, the best reported performance has been achieved by the SO(2) continuous rotation equivariance. The metric used here does not follow the standard T-LESS metric since it is only used for model variants comparison.

The difference between rotation-equivariant and classical CNN features is highlighted in Figure 6. By using images with different orientations (Figure 6(a)) as input, the same transformation links extracted feature maps from different stages of the model (b). In contrast, this is not the case with the classical CNN where the obtained feature maps are not rotated versions of each other (c).

**Datasets.**

**Cambridge Landmarks** – We use this dataset [35] to evaluate the performance of *E-PoseNet* in outdoor camera re-localization. It is a large scale dataset taken around Cambridge University, containing original videos labelled with 6-DoF camera poses and a visual reconstruction of the scene (spatial extent of $\sim 900 - 5500m^2$). We train and evaluate *E-PoseNet* on four scenes (see Table 1). Furthermore, a few samples are used to visualize the obtained *E-PoseNet* feature fields. Figure 8 shows that they directly support representations of SE(2) and are therefore enriched with some notion of orientation, visualized in a vector field form. On the contrary classical CNNs do not encode geometric information directly into their feature space.

**7-Scenes** – For indoor camera localization, we use the 7-Scenes dataset [53] which is a collection of tracked RGB-D camera frames for indoor scenes with a spatial extent of $\sim 1 - 10m^2$. Only RGB images are used in our experiments.

Finally, the two datasets present various challenges, *i.e.*, occlusion, reflections, motion blur, lighting conditions, repetitive textures, and variations in viewpoint and trajectory.

**Comparative Analysis of Camera Pose Regressors.** We compare the performance of *E-PoseNet* with state-of-the-art APR methods for camera localization in both outdoor and indoor scenes. First, we tested the performance on the Cambridge Landmarks dataset, for which we provide the median position and orientation errors in Table 1. We also compare the performance of *E-PoseNet* with respect to the state-of-the-art monocular pose regressors reporting on the 7-Scenes dataset. Table 2 contains the results. From the results on both datasets, and in comparison with APR methods, we
Table 1. Comparative analysis of pose regressors on Cambridge Landmarks dataset (outdoor localization) [35] - We report the median position/orientation error in meters/degrees for each method. Best results are highlighted in bold.

<table>
<thead>
<tr>
<th>Method</th>
<th>Chess</th>
<th>Fire</th>
<th>Heads</th>
<th>Office</th>
<th>Pumpkin</th>
<th>Kitchen</th>
<th>Stairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>DenseVLAD + Inter. [56]</td>
<td>0.18/10.0</td>
<td>0.47/12.4</td>
<td>0.15/14.3</td>
<td>0.25/10.1</td>
<td>0.26/9.42</td>
<td>0.27/11.1</td>
<td><strong>0.24/14.7</strong></td>
</tr>
<tr>
<td>PoseNet (PN) [35]</td>
<td>0.32/8.12</td>
<td>0.47/14.4</td>
<td>0.29/12.0</td>
<td>0.48/7.68</td>
<td>0.47/8.42</td>
<td>0.59/8.64</td>
<td>0.47/13.8</td>
</tr>
<tr>
<td>PN learned weights [34]</td>
<td>0.14/4.50</td>
<td>0.27/11.8</td>
<td>0.18/12.1</td>
<td>0.20/5.77</td>
<td>0.25/4.82</td>
<td>0.24/5.52</td>
<td>0.37/10.6</td>
</tr>
<tr>
<td>BayesianPN [33]</td>
<td>0.37/7.24</td>
<td>0.43/13.7</td>
<td>0.31/12.0</td>
<td>0.48/8.04</td>
<td>0.61/7.08</td>
<td>0.58/11.5</td>
<td>0.48/13.1</td>
</tr>
<tr>
<td>LSTM-PN [58]</td>
<td>0.24/5.77</td>
<td>0.34/11.9</td>
<td>0.21/13.7</td>
<td>0.30/8.08</td>
<td>0.33/7.0</td>
<td>0.37/8.83</td>
<td>0.40/13.7</td>
</tr>
<tr>
<td>GPoseNet [7]</td>
<td>0.20/7.11</td>
<td>0.38/12.3</td>
<td>0.21/13.8</td>
<td>0.28/8.83</td>
<td>0.37/9.74</td>
<td>0.35/8.15</td>
<td>0.37/12.5</td>
</tr>
<tr>
<td>MapNet [3]</td>
<td>0.13/4.48</td>
<td>0.27/11.3</td>
<td>0.17/13.0</td>
<td>0.19/5.55</td>
<td>0.26/4.75</td>
<td>0.23/5.35</td>
<td>0.35/12.4</td>
</tr>
<tr>
<td>GeoPoseNet [34]</td>
<td><strong>0.08/3.25</strong></td>
<td>0.27/11.7</td>
<td>0.18/13.3</td>
<td><strong>0.17/5.15</strong></td>
<td>0.22/4.02</td>
<td>0.23/4.93</td>
<td>0.30/12.1</td>
</tr>
<tr>
<td>IRPNet [52]</td>
<td>0.13/5.64</td>
<td>0.25/9.67</td>
<td>0.15/13.1</td>
<td>0.24/6.33</td>
<td>0.22/5.78</td>
<td>0.30/7.29</td>
<td>0.34/11.6</td>
</tr>
<tr>
<td>AttLoc [59]</td>
<td>0.10/4.07</td>
<td>0.25/11.4</td>
<td>0.16/11.8</td>
<td>0.17/5.34</td>
<td>0.21/4.37</td>
<td>0.23/5.42</td>
<td>0.26/10.5</td>
</tr>
<tr>
<td>MS-Transformer [15]</td>
<td>0.11/4.66</td>
<td>0.24/9.60</td>
<td>0.14/12.2</td>
<td>0.17/5.66</td>
<td>0.18/4.44</td>
<td><strong>0.17/5.94</strong></td>
<td>0.26/8.45</td>
</tr>
<tr>
<td>TransPoseNet [16]</td>
<td><strong>0.08/5.68</strong></td>
<td>0.24/10.6</td>
<td><strong>0.13/12.7</strong></td>
<td>0.17/6.34</td>
<td>0.17/5.60</td>
<td>0.19/6.75</td>
<td><strong>0.30/7.02</strong></td>
</tr>
<tr>
<td>E-PoseNet (Ours)</td>
<td><strong>0.08/2.57</strong></td>
<td><strong>0.21/11.0</strong></td>
<td><strong>0.16/10.3</strong></td>
<td><strong>0.15/6.80</strong></td>
<td><strong>0.16/3.82</strong></td>
<td><strong>0.20/6.81</strong></td>
<td><strong>0.24/9.92</strong></td>
</tr>
</tbody>
</table>

conclude that the proposed E-PoseNet achieves the lowest location error across all the outdoor and indoor scenes, and the lowest orientation error across the majority of them. It also competes with most recent transformer-based architectures [15, 16] on these datasets.

Implementation Details. We tested different architectures for the equivariant backbone, with ResNet18 being the most suitable model in our experiments, from both model size and performance perspectives. We trained our model for 5 – 10k epochs using Adam optimizer [36], with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-5}$ and a batch size of 256. During the training phase, we rescaled the image so that its smaller length is 256 pixels followed by a random $224 \times 224$ crop. No further data augmentation was used.

Limitations. While we focus on introducing equivariant operations for the feature extraction part of the APR pipeline, the following stages (i.e. embedding, regression) do not have the same property, resulting in breaking the equivariance of the overall pipeline. Another limitation of the proposed APR model is the longer time required for equivariant CNN models as compared to classical CNN ones. Note that this is only during training, while the inference time is similar for both types of models.

7. Conclusions
This paper presents a new direction for the problem of camera pose regression leveraging equivariant features to encode more geometric information about the input image. By using an SE(2)-equivariant feature extractor, our model is able to outperform existing methods on both outdoor and indoor benchmarks. Furthermore, we conclude that the equivariant properties of deep learning models that are used for geometric reasoning offer a promising direction for reaching the potential of absolute pose regression.
References


