2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) | 979-8-3315-4364-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/CVPR52734.2025.01078

HyperPose: Hypernetwork-Infused Camera Pose Localization and an Extended Cambridge Landmarks Dataset

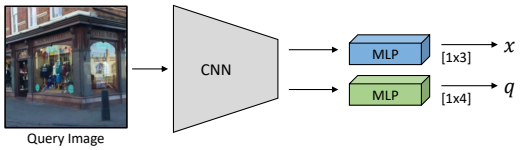
Ron Ferens Yosi Keller
Faculty of Engineering, Bar Ilan University, Ramat-Gan, Israel
{ronferens, yosi.keller}@gmail.com

Abstract

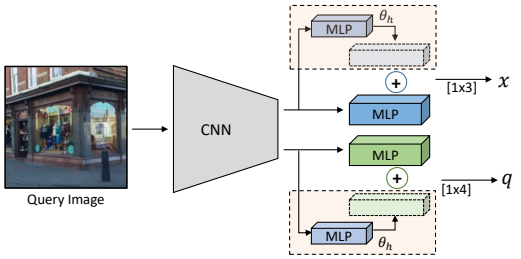
In this work, we propose HyperPose, which utilizes hypernetworks in absolute camera pose regressors. The inherent appearance variations in natural scenes, attributable to environmental conditions, perspective, and lighting, induce a significant domain disparity between the training and test datasets. This disparity degrades the precision of contemporary localization networks. To mitigate this, we advocate for incorporating hypernetworks into single-scene and multiscene camera pose regression models. During inference, the hypernetwork dynamically computes adaptive weights for the localization regression heads based on the particular input image, effectively narrowing the domain gap. Using indoor and outdoor datasets, we evaluate the HyperPose methodology across multiple established absolute pose regression architectures. We also introduce and share the Extended Cambridge Landmarks (ECL), a novel localization dataset, based on the Cambridge Landmarks dataset, showing it in multiple seasons with significantly varying appearance conditions. Our empirical experiments demonstrate that HyperPose yields notable performance enhancements for single- and multi-scene architectures. We have made our source code, pre-trained models [1], and the ECL dataset openly available [2].

1. Introduction

In many computer vision applications, such as indoor navigation, augmented reality, and autonomous driving, it is essential to determine the position and orientation of a camera using an image taken by that camera. Current camera localization techniques show various trade-offs between accuracy, runtime, and memory usage. Feature Matching (FM) and Scene Coordinate Regression (SCR) are considered the most accurate localization methods. FM-based methods [15, 27, 37–41, 51, 54] involve hierarchical localization pipelines, which commonly use a coarse-to-fine processing pipeline that begins with image retrieval to perform an initial localization based on images similar to the query image. The initial estimate is refined by local feature extraction and matching. To estimate the camera's 6-Degrees-Of-Freedom (6DoF) accurately, the resulting 2D-2D matches are mapped to 3D correspondences in the scene's point cloud and then used to determine the camera pose through Perspective-n-Point (PnP) and RANSAC. SCR-based methods [4–8, 47] directly regress correspondences between 2D pixels and 3D coordinates in the scene. Relative pose regression (RPR) schemes [2, 14, 17, 20, 26, 57, 62] directly estimate the relative pose between a query image and a set of reference images with well-defined ground truth poses, eliminating the need for conventional pipelines. However, an image retrieval scheme is essential to retrieve a set of nearest-neighbor images. Another approach, absolute pose



(a) Baseline architecture.



(b) Baseline architecture with a hypernetwork.

Figure 1. Figure 1a illustrates a baseline APR architecture. It consists of a CNN backbone that generates a feature vector encoding the input query image, followed by regression heads that estimate the translation and orientation. Figure 1b shows the extension of this baseline architecture with a hypernetwork that outputs the weights for the regression heads (translation and orientation) during inference.

[1] https://ronferens.github.io/hyperpose/
[2] https://ronferens.github.io/extcambridgelandmarks/

2575-7075/25/$31.00 ©2025 IEEE
DOI 10.1109/CVPR52734.2025.01078

regressors (APRs), can estimate the camera pose with a single forward pass using only the query image, resulting in significantly lower latency, although notably less accurate. Additionally, due to their low memory footprint, APRs can be deployed as a standalone application on edge devices with limited computational resources. A survey of visual camera pose localization is given by [43, 65].

PoseNet by Kendall et al. [25] was the first APR approach, which leveraged a convolutional backbone and a multilayer perceptron (MLP) head to regress the camera's position and orientation. This simple and computationally efficient approach, enabling execution at 5ms, led to numerous APR methods that improved accuracy by modifying the backbone and MLP architectures [11, 32, 33, 44, 49, 60, 61, 63]. In addition, alternative loss functions and optimization strategies were explored [23, 24, 43]. Indoor and outdoor environments exhibit dynamic variability in illumination, perspective, and object motion. APR models, despite their computational efficiency, often struggle to maintain accuracy under these diverse conditions and show limited generalization to novel scenes, necessitating scene-specific training. Consequently, while architecturally simple, a unified network might yield suboptimal performance when adapting to heterogeneous real-world visual inputs. Recent studies have proposed extending APR from single-scene to multiscene formulations. Blanton et al. [3] implemented a convolutional neural network (CNN) backbone to compute a global descriptor for the input image. Rather than employing a scene-specific multilayer perceptron (MLP), they optimized a set of fully connected (FC) layers, allocating one layer per scene based on a predicted scene identifier. Although introducing a framework to optimize a single model across multiple scenes, this methodology did not achieve state-of-the-art (SOTA) APR accuracy. An alternative approach was presented by [46] using dual-branch transformers [58]. Encoders focused on extracting pose-related features, while decoders transformed encoded scene identifiers into latent pose representations. The decoder outputs are leveraged for scene classification and selection of corresponding position and orientation embeddings, which are used to regress position and orientation vectors.

Hypernetworks [22] are a deep learning architecture in which an auxiliary network, the hypernetwork, generates the weights for the primary network (main network) based on the current input and downstream task. The main network and hypernetwork are trained end-to-end using backpropagation. The hypernetwork allows for dynamic and adaptive weight modulation of the main network, leading to improved flexibility in response to changing conditions [16, 18, 34, 55].

In this work, we propose *HyperPose*, a hypernetwork-based approach for absolute camera pose regression. Extending common APR methods [11, 23, 25, 44, 45, 60], the hypernetwork computes weights for the regression head. This enables adaptive analysis of image content, allowing the regression weights to focus on informative features, rather than relying on a single constant pretrained set of weights. We validated the proposed method by integrating hypernetworks into single- and multiscene APR architectures. For single-scene networks, we present HyperPose with an existing state-of-the-art APR model [61] as well as a simplistic baseline APR using an off-the-shelf backbone [53] and shallow regression heads, inspired by the principles outlined in [25]. For multiscene, we extend a leading multiscene APR model [46] by integrating a hypernetwork architecture. In both instances, hypernetwork-augmented models demonstrated superior performance compared to their original counterparts when evaluated on the benchmark datasets in this study. To evaluate localization robustness, we introduce an Extended Cambridge Landmarks (ECL) dataset - an augmentation of the original benchmark [25] with algorithmically generated environmental shifts. By combining evening, winter, and summer scenes, the ECL incorporates dynamics such as lighting shifts and seasonal transitions.

Thus, we propose the following contributions:

- We propose an approach to enhance any existing absolute camera pose regression models by integrating hypernetwork architectures.
- We comprehensively evaluated the proposed HyperPose on both single- and multiscene pose regressors, demonstrating improved accuracy achieved by the hypernetwork-enhanced models over the original versions.
- The proposed scheme achieves new SOTA accuracy on multiscene APR benchmarks spanning diverse outdoor and indoor localization tasks.
- We introduce the Extended Cambridge Landmarks (ECL) dataset, which builds upon the foundation of the original Cambridge Landmarks dataset [25]. Our ECL data set extends existing test scenes by incorporating various appearance conditions.

2. Related Work

Camera pose estimation techniques differ by their input during inference and algorithmic attributes.

3D-based or structure-based localization methods are considered the most accurate for camera pose estimation and have demonstrated state-of-the-art performance in leading benchmarks. These methods utilize correspondences between 2D features in an image and 3D world coordinates. [15, 27, 35, 37–41, 51, 54] introduced a two-phase approach for hierarchical pose estimation pipelines. Thus, each query image is first encoded using a CNN trained for image retrieval (IR), based on a pre-mapped image dataset. Tentative correspondences are then estimated by matching local im-

age features to 3D matches. The resulting matches are used by PnP-RANSAC to estimate the camera pose. DSAC [8] and its succeeding scheme, DSAC++ [4], utilize a CNN architecture to directly estimate 3D coordinates from 2D positions in an image and only require the query image during inference. The same as in Absolute Pose Regression (APR) methods, separate models must be trained for each scene, achieving state-of-the-art accuracy. Lately, [15] proposes a solution to mitigate this gap by introducing a scene-agnostic architecture.

Relative Pose Regression (RPR) algorithms estimate the relative pose between an input query image and a set of reference images based on their known location. Calculating the relative pose involves estimating the relative position and orientation between the query image and the reference images [1, 6, 68]. An Image Retrieval (IR) approach is utilized to determine the closest set of neighbor images, and the relative motion is then estimated between the query image and each of the retrieved images. The estimated relative poses are used, along with the known position of the reference images, to estimate the camera's absolute pose. Similar to 3D and structured-based localization methods, RPRs involve a multistep process and require access to a database of images with labeled poses during the inference phase.

Absolute Pose Regression (APR) directly estimates the camera pose given the input query image. APRs were first introduced by Kendall [25], using a modified version of the truncated GoogLeNet architecture, where the Softmax classification layer was replaced by a series of fully connected layers that output the pose estimate. Although less accurate than classical structure-based methods, APRs offer several advantages, such as faster inference times (milliseconds), reduced memory footprint (megabytes), and do not require feature engineering. Variations of the encoder and MLP architectures have been proposed to improve APR accuracy [11, 23, 33, 33, 44, 49]. Furthermore, modifications to the loss function have been suggested to improve the balance between orientation and position objectives [24] or to incorporate other types of information [9]. For an in-depth examination of the various architectural designs and loss functions utilized in camera pose regression, please refer to [43]. Although demonstrating faster inference times and low memory requirements, absolute pose regression (APR) models typically require dedicated per-scene training to achieve competitive accuracy compared to other methods. This limitation has raised concerns regarding the value of APRs versus significantly more accurate techniques such as [4, 6, 8, 36]. To address this issue, recent approaches suggested extending APRs to a multiscene paradigm. [3] introduced Multi-Scene PoseNet (MSPN), which utilizes a CNN to regress an image feature vector. The two branches then process this feature vector. The first branch classifies the related scene using a fully connected layer with a softmax operator. The estimated scene index selects scene-specific weights from a database. The second branch combines these weights with the feature vector to regress the pose. The model is trained end-to-end by applying both binary cross-entropy (for classification) and camera pose losses for each scene. Inspired by recent successful applications of Transformers [12, 19, 29] to computer vision tasks, [46] suggested a shared convolutional network backbone followed by two network branches consisting of a Transformer and a regression head. The first regresses the input query image's pose, and the other its orientation. The main incentive for utilizing Transformers encoders is to focus on pose-informative features and decoders to transform encoded scene identifiers into latent pose representations.

Hypernetworks were first presented by Ha et al. [22]. These are neural networks that have been designed to predict the weights of a primary network. The primary network's weights can be adjusted based on specific inputs, resulting in a more expressive and adaptive model. Hypernetworks have been used in various applications such as learning implicit neural representations [16], semantic segmentation [34], 3D scene representation and modeling [28, 48, 50] and continuous learning [59], to name a few.

3. APRs using Hypernetworks

HyperPose is a general approach, applicable to both single- and multi-scene scenarios, for enhancing APR schemes. It comprises two key components: the main network (APR) designed to estimate the camera's absolute position and the hypernetwork that computes the regression weights for the main network conditioned on the input query image. The main network performs a forward pass on the input image to localize the capture camera. The camera pose is represented as the tuple $< \mathbf{x}, \mathbf{q} >$, where $\mathbf{x} \in \mathbb{R}^3$ is the camera's position in world coordinates and $\mathbf{q} \in \mathbb{R}^4$ is the quaternion encoding of its $3D$ orientation. Following [13], we formulate the hypernetwork H as a parametric function $\theta_h = H(\theta, \mathbf{x})$, where θ represents the parameters of the hypernetwork, $\mathbf{x}$ the network's input, and θ_h the parameters of the regression heads layer for the input query image I_q.

3.1. Network Architecture

3.1.1. HyperPose for Single-Scene APRs

We apply the proposed HyperPose-based approach to single-scene APR using a baseline APR following [25], and the state-of-the-art *AtLoc* APR [61].

Baseline APR with Hypernetwork. As depicted in Fig. 1a, the baseline model's main network consists of a CNN backbone that generates a feature vector encoding the input query image. This vector is then processed by two regression heads. The first estimates the camera position, and the other estimates the orientation. Figure 1b illustrates the integration of a hypernetwork into this architecture. The input

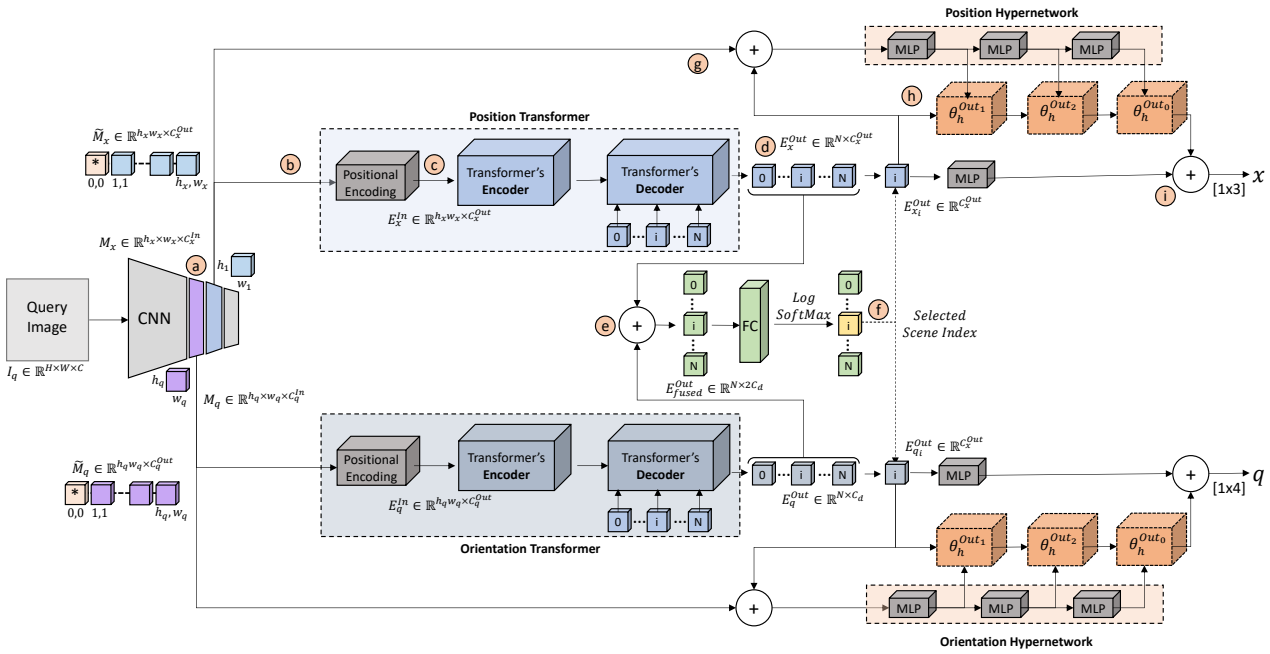


Figure 2. **MS-HyperPose** - The proposed multi-scene absolute pose regression architecture using a hypernetwork. The primary network employs position and orientation Transformers, in a dual-branch architecture, to extract activation maps from the underlying convolutional backbone. The hypernetwork generates the weights for the regression head in the primary network, using the input query image. These adaptive weights, combined with the latent vectors generated by the Transformers estimate the camera pose, consisting of the spatial position (x) and angular orientation (q).

to the hypernetwork is the feature vector generated by the CNN backbone of the main network. This vector is passed through a single fully-connected (FC) layer followed by a Swish activation [36] function. The processed vector serves as an input to a multilayer perceptron (MLP) that outputs the weights for a single layer in the regression heads of the main network. The camera pose is then estimated by the position and orientation regression heads of the main network.

AtLoc with Hypernetwork. [61] presents Attention-Guided Camera Localization (AtLoc), an absolute pose regression model based on self-attention. In this architecture, a visual encoder encodes an image into a latent vector. Conditioned on these extracted features, an attention module computes self-attention maps to reweight the representation into a new feature space. A pose regressor then maps the transformed features after the attention operators to estimate the camera pose. We incorporate a hypernetwork into the AtLoc, the same as for the baseline APR model. In this case, the input to the hypernetwork is the reweighted representation feature vector computed by the visual encoder.

3.1.2. HyperPose for Multi-Scene APRs

The suggested multi-scene model, named *MS-HyperPose* is an extension of the architecture introduced in [45]. As depicted in Fig. 2, the input query image I_q is processed by a convolutional backbone that computes intermediate activation maps from two different layers. These activation maps are then utilized as inputs to the main network and the hypernetwork. Within the main network, the positional and orientational Transformers process the activation maps and integrate sequential representations into single latent vectors. After converting the activation maps into representation vectors, these vectors, along with the latent vectors are passed through multilayer perceptron (MLP) layers in the hypernetwork to generate the weights θ_h for the translation and orientation regression heads. Using the weights θ_h produced by the hypernetwork H and the regression heads in the main network, the position, and orientation of the input query image are estimated.

The Main Network consists of a convolutional backbone and two separate branches to regress the camera's position and orientation. Each branch includes its particular Transformer and a multilayer perceptron (MLP) head. Studies have demonstrated that substituting the patchify stem, utilized in the early visual processing of a Vision Transformer (ViT), with a convolutional stem enhances optimization stability and elevates performance to a higher level [64]. Therefore, given an input query image $I_q \in \mathbb{R}^{H \times W \times C}$, a shared convolutional backbone generates activation maps. Two different activation maps are selected

from the CNN based on the particular learned task (Fig. 2a). The activation maps, represented as $M \in \mathbb{R}^{h \times w \times C^{In}}$, are processed and transformed into $\tilde{M} \in \mathbb{R}^{h \times w \times C^{Out}}$ using a 1×1 convolutional layer followed by flattening (Fig. 2b).

Following [45], we use the Transformer architecture from [12], modifying the standard Encoder and Decoder to add positional encoding at each attention layer. Each Transformer has L identical blocks, each with a multi-head attention (MHA) mechanism and a two-layer MLP with Swish activation [36]. Layer Normalization (LN) is applied before MHA and MLP processing, as in [58], with outputs merged via residual connections and dropout. To preserve spatial information, each activation map position receives a learned encoding. The processed sequence, serving as input to the Transformer is given by (Fig. 2c):

$$E^{In} = [T, \tilde{M}] \in \mathbb{R}^{(hw) \times C^{Out}}, T \in \mathbb{R}^{C_M^{Out}}. \quad (1)$$

where T is the positional encoding of the activation map.

Given a dataset with N scenes, the Transformer Decoders generate embeddings $E^{Out} =\in \mathbb{R}^{N \times C^{Out}}$ (Fig. 2d). For a query image, only one position matches its scene. The combined Transformer outputs (Fig. 2e) pass through a fully connected layer and Log SoftMax, selecting the highest-probability vectors (Fig. 2f).

The selected vectors $E_{x_i}^{Out}$ and $E_{q_i}^{Out}$ are processed through an MLP. Finally, the estimated camera position and orientation are obtained by summing the outputs of the main network and hypernetwork, using weights predicted by the hypernetwork (Fig. 2h and 2i).

The Hypernetworks consist of MLPs, each regressing the weights for a corresponding regression head (position or orientation) in the main network. Their input is the sum of task-specific encoded activation maps $\tilde{M}_x$ and $\tilde{M}_q$, along with global latent vectors $E_{x_i}^{Out}$ and $E_{q_i}^{Out}$ (Fig. 2g). To ensure size compatibility, the encoded activation map passes through a fully connected layer before summation. This vector is then processed by MLPs with Swish activation, generating inputs for another MLP set, each estimating weights for a specific layer in the main network's regression heads:

$$\theta_h^{Out_j} = H(\theta_h^{Out_j}, E_i^{Out}) \in \mathbb{R}^{C_{E_i}^{Out} \times C_H^{Out_j}}, \quad (2)$$

where j is the index of the regression head layer.

3.2. Optimization Criteria

The localization error is calculated by comparing the translation position and orientation of the ground truth pose $\mathbf{p}_{gt} =< \mathbf{x}_{gt}, \mathbf{q}_{gt} >$, where $\mathbf{x} \in \mathbb{R}^3$ represents the position of the camera in the world and $\mathbf{q} \in \mathbb{R}^4$ denotes its orientation encoded as a quaternion, with the estimated pose $\mathbf{p}_{est} =< \mathbf{x}_{est}, \mathbf{q}_{est} >$. The translation error, represented by L_x, is calculated by the Euclidean distance between the ground truth and the estimated camera positions. The orientation error, L_q, is defined as the Euclidean distance between the ground truth and the estimated unit quaternions [43]:

$$L_x = ||\mathbf{x}_0 - \mathbf{x}||_2, \quad (3)$$

$$L_q = \left\| \mathbf{q_0} - \frac{\mathbf{q}}{||\mathbf{q}||} \right\|_2 \quad (4)$$

where $\mathbf{q}$ is a normalized norm quaternion to ensure it is a valid orientation encoding.

As a multi-regression optimization problem, we adopt the method outlined in [24] by using different weighting schemes, based on the uncertainty of each task, to balance the individual loss functions defined for the separate objectives. The model optimizes the aggregated loss function, a combination of the weighted individual losses.

$$L_p = L_x \exp(-s_x) + s_x + L_q \exp(-s_q) + s_q, \quad (5)$$

where S_x and S_q are the learned parameters. As a further optimization, we adopt the approach detailed in [44] and follow a three-step training process. Firstly, the entire network is trained using the aggregated loss in Eq. 5. In the next two stages, each MLP head undergoes separate fine-tuning with its corresponding loss function.

3.3. Implementation Details

The shared backbone uses EfficientNet-B0 [31, 52]. We used the activation maps of two endpoints (reduction levels) as input for our position and orientation branches: $m_{rdct4} \in \mathbb{R}^{14 \times 14 \times 112}$ and $m_{rdct3} \in \mathbb{R}^{28 \times 28 \times 40}$, respectively. Following [45], we applied linear projections on each activation map to a common depth dimension of $C_M^{Out} = 256$ and learned the positional encoding of the same depth. The regression heads in the main network consist of two-layer MLPs with input and output layers. The input layer dimension is 1024 for both the position and orientation, while the output dimensions are 3 and 4, respectively.

Single-Scene APR. For both the baseline APR and the HyperPose extension of the AtLoc [61] model, the hypernetwork for the position branch generates weights for an MLP with a dimension of 256, while the orientation hypernetwork produces weights of 512 dimensions.

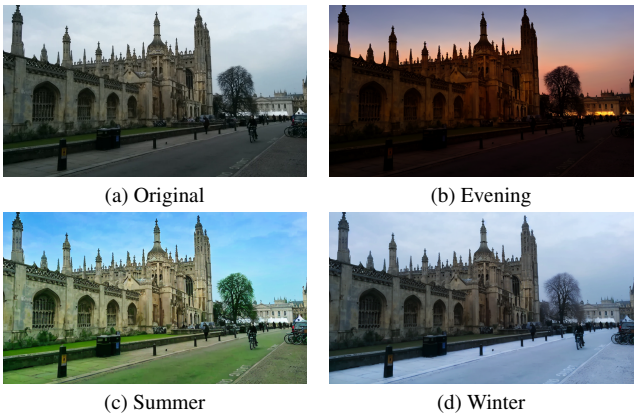
Multi-Scene APR. In contrast to [45], we incorporated four-layer (instead of six) Transformers in each branch (position and orientation) within the main network architecture. Each block consists of an MHA layer with four heads, followed by an MLP that preserves the input dimension. A dropout rate of $p = 0.1$ was applied to each MLP layer. The output dimension of each Transformer in the main network

Table 1. **Comparative analysis of APR architectures with and without the purposed hypernetwork addition using the Cambridge Landmarks dataset**. We report the median position/orientation error in meters/degrees. The best performance is marked in **bold**. (*) The authors of AtLoc did not report results on the Cambridge dataset, and we trained the AtLoc model on this dataset.

Method	K. College	Old Hospital	Shop Facade	St. Mary	Avg.
AtLoc* [61]	1.53,**3.21**	2.17,3.99	0.93,4.61	2.14,5.76	1.69,4.39
AtLoc w/ hypernet	**0.96**,3.43	**2.04**,**3.61**	**0.88**,**4.22**	**1.72**,**5.44**	**1.40**,**4.18**
Baseline APR	0.89,2.29	1.49,3.32	0.74,4.79	1.40,4.95	1.13,3.84
Baseline APR w/ hypernet	**0.77**,**2.07**	**1.47**,**3.29**	**0.72**,**4.01**	**1.37**,**4.90**	**1.08**,**3.57**

was $C_M^{Out} = 256$. The position and orientation branches of the hypernetwork generate the weights of three regression layers: input, hidden, and output. The number of weights in the position and orientation branches differ. The hypernetwork's position branch produces the weights for an MLP with dimensions 256, 128, and 3, while the orientation branch generates weights for an MLP with dimensions 512, 512, and 4. The weights generated by the hypernetwork are then put into the regression heads of the main network to estimate the position and orientation of the input query image. The input and hidden layers of the regression head utilize the Swish activation function.

4. Extended Cambridge Landmarks Dataset



(a) Original (b) Evening

(c) Summer (d) Winter

Figure 3. A sample from the King's College Scene in the proposed Extended Cambridge Landmarks (ECL) Dataset

To further assess the limitations of the HyperPose approach, we constructed an augmented version of the Cambridge Landmarks [25] benchmark called the Extended Cambridge Landmarks (ECL) dataset. Leveraging the original scenes, the ECL dataset introduces new localization challenges due to variations in appearance by creating synthetic evening, winter, and summer variants for each location. These scene permutations simulate real-world environmental variations by introducing shifts in lighting, weather patterns, foliage density, and more. Analyzing performance across ECL variations and their original Cambridge counterparts allows us to evaluate the robustness of models to real-world variations. These augmentations were conducted by utilizing the novel *InstructPix2Pix* work [10]. Visual samples of the ECL dataset can be found in the supplementary materials. Figure 3 shows sample from the King's College scene in the extended dataset. Additional samples from each scene in the ECL dataset can be found in the supplementary material. Quantitative pose regression analysis between these more challenging conditions and unmodified sequences helps scientifically assess model adaptability. We hypothesize that exposing limitations unseen in conventional test sets will motivate the development of more invariant solutions. As controllable expansions of established benchmarks, programmatically generated datasets can thus guide the development of robust models for real-world use.

5. Experimental Results

We conducted an experimental evaluation using the 7 Scenes [21] and Cambridge Landmarks [25] datasets for evaluating absolute pose regression. The Cambridge Landmarks dataset represents an outdoor urban environment ranging in size from 900 to 5500 square meters. We present results for four of its six scenes. We excluded the remaining two scenes due to limited coverage in prior work. To compare robustness, we also report the results on the Extended Cambridge Landmarks dataset (ECL) (Section 1 in the supplementary materials). The 7 Scenes dataset consists of seven small-scale indoor scenes with spatial extents ranging from approximately 1 to 10 square meters. These datasets present various localization challenges, including differences in scale, indoor versus outdoor settings, repetitive elements, textureless features, significant viewpoint changes, and variations in trajectory between training and test sets. To facilitate a broader analysis of our proposed method, we conduct an additional evaluation on the Oxford RobotCar Dataset [30]. Such dynamic environmental factors assess HyperPose's ability to improve localization accuracy amid diverse, challenging visual contexts. Our experimental procedures adhere to the same training and evaluation protocols as in [9, 61]. Additional information concerning these sequences, and a comprehensive description of the training methodology and hyperparameter selection for the models, are provided in the supplementary material.

We evaluate the impact of integrating hypernetworks in

Table 2. **Comparison to single-scene state-of-the-art methods - Cambridge Landmarks:** We report the median position/orientation error in meters/degrees. The most effective single and multi-scene APRs are marked in **bold**.

		K. College	Hospital	Shop Facade	St. Mary	Avg.
	DSAC* [6]	0.18,0.3°	0.21,0.4°	0.05,0.3°	0.15,0.5°	0.15,0.4°
Seq.	MapNet [9]	1.08,1.9°	1.94,3.9°	1.49,4.2°	2.00,4.5°	1.63,3.6°
	GL-Net [67]	0.59,0.7°	1.88,2.8°	0.50,2.9°	1.90,3.3°	1.22,2.4°
IR	VLAD [56]	2.80,5.7°	4.01,7.1°	1.11,7.6°	2.31,8.0°	2.56,7.1°
	VLAD+Inter [42]	1.48,4.5°	2.68,4.6°	0.90,4.3°	1.62,6.1°	1.67,4.9°
RPR	EssNet [68]	0.76,1.9°	1.39,2.8°	0.84,4.3°	1.32,4.7°	1.08,3.4°
	NC-EssNet [68]	0.61,1.6°	0.95,2.7°	0.7,3.4°	1.12,3.6°	0.85,2.8°
	RelocGNN[21]	0.48,1.0°	1.14,2.5°	0.48,2.5°	1.52,3.2°	0.91,2.3°
APR	PoseNet [25]	1.92,5.40°	2.31,5.38°	1.46,8.08°	2.65,8.48°	2.08,6.83°
	BayesianPN [23]	1.74,4.06°	2.57,5.14°	1.25,7.54°	2.11,8.38°	1.91,6.28°
	LSTM-PN [60]	0.99,3.65°	1.51,4.29°	1.18,7.44°	1.52,6.68°	1.30,5.57°
	SVS-Pose [33]	1.06,2.81°	1.50,4.03°	**0.63**,5.73°	2.11,8.11°	1.32,5.17°
	GPoseNet [11]	1.61,2.29°	2.62,3.89°	1.14,5.73°	2.93,6.46°	2.07,4.59°
	PoseNetLearn [24]	0.99,1.06°	2.17,**2.94**°	1.05,3.97°	1.49,3.43°	1.42,2.85°
	GeoPoseNet [24]	0.88,**1.04**°	3.20,3.29°	0.88,3.78°	1.57,**3.32**°	1.63,2.86°
	IRPNet [44]	1.18,2.19°	1.87,3.38°	0.72,**3.47**°	1.87,4.94°	1.41,3.50°
	Baseline APR w/ hypernet	**0.61**,1.84°	**1.44**,3.03°	0.70,3.62°	**1.37**,4.85°	**1.03**,**2.67**°
MS-APR	MSPN [3]	1.73,3.65°	2.55,4.05°	2.92,7.49°	2.67,6.18°	2.47,5.34°
	MS-Trans[46]	0.83,1.47°	**1.81**,2.39°	0.86,**3.07**°	1.62,3.99°	1.28,2.73°
	MS-HyperPose	**0.78**,**1.18**°	1.84,**2.11**°	**0.83**,3.13°	**1.61**,**3.22**°	**1.27**,**2.41**°

single-scene APRs. To this end, we compare the accuracy of two model variations, with and without hypernetworks, and report median position and orientation errors. Table 1 presents results on the Cambridge dataset. The hypernetwork improves the accuracy of the proposed baseline APR and AtLoc [61] models. The baseline APR model size is 26.7MB with an inference time of 9.32ms, increasing to 40.9MB and 9.45ms with a hypernetwork. The original published AtLoc model is 97.9MB with an inference time of 5.14ms. The hypernetwork-enhanced AtLoc version has a model size of 116.8MB and an inference time of 5.56ms. Note that the reported errors refer to the models' performance after the first stage of model training, without applying the two additional optimization steps described in the training details (See supplementary materials). Furthermore, Tables 2 and 3 compare the median position error, and the orientation error, of the proposed baseline APR with hypernetwork architecture, with the results of recent SOTA localization algorithms, using the Cambridge Landmarks and 7Scenes datasets, respectively. For the indoor and outdoor datasets, in addition to the results of leading APRs, we report the median position and orientation errors of structure-based, sequence-based, IR, and RPR methods. Although DSAC* [6] achieves SOTA accuracy, compared to all other APR methods, HyperPose obtains the most accurate average of $1.03m$, $2.67°$ on the Cambridge outdoor dataset, and $0.19m$, $7.48°$ on the 7Scenes indoor dataset. These localization errors show competitive performance compared to leading RPR techniques, with the hypernet-enhanced baseline ranking third and second/fourth (7Scenes/Cambridge) in positional and rotational accuracy, respectively. Next, we compare the proposed MS-HyperPose architecture with other multiscene APR methods. Tables 2 and 3 present median position and orientation errors, respectively, on the Cambridge Landmarks and 7 Scenes datasets. On both indoor and outdoor benchmarks, our MS-HyperPose approach achieves the most accurate average of 1.27m, 2.41° on Cambridge and 0.18m, 7.12° on 7Scenes compared to other single- and multi-scene APRs. Additionally, MS-HyperPose is the most accurate for both positional and rotational errors in 6 out of 8 parameters on Cambridge and 6 out of 14 on 7 Scenes. The runtime of MS-HyperPose is 30.93ms, similar to the reference architecture of [46]. Its model size is 571MB, notably larger than the 71.1MB of the original. We discuss storage considerations in the *Limitation and Future Work* section of the supplementary materials.

Table 4 reports the mean and median position and orientation errors of AtLoc [61] compared to our hypernetwork-enhanced variant using the Oxford RobotCar Dataset [30]. We also compare the enhanced model with [9], the SOTA APR for this dataset using image sequences. Visualizations of the localization results can be found in the supplementary materials.

Table 3. **Comparison to single-scene SOTA methods - 7Scenes:** We report the median position/orientation error in meters/degrees. The most effective single and multi-scene APRs are distinguished in **bold**.

		Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs	Avg.
SCR	DSAC* [6]	0.02,1.11°	0.02,1.24°	0.01,1.82°	0.03,1.18°	0.04,1.41°	0.03,1.70°	0.04,1.42°	0.03,1.41°
	Marepo[15]	0.02,1.24°	0.02,1.39°	0.02,2.03°	0.03,1.26°	0.04,1.48°	0.04,1.71°	0.06,1.67°	0.03,1.54°
Seq.	LsG [66]	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°	0.19,7.74°
	MapNet[9]	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[67]	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°
IR	VLAD [56]	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[42]	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 [26]	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[2]	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 [68]	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 [68]	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[21]	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 [17]	0.04,1.7°	0.03,1.7°	0.05,2.00°	0.04,1.60°	0.04,1.6°	0.04,1.6°	0.04,1.5°	0.04,1.70°
APR	PoseNet [25]	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°	0.44,10.4°
	BayesianPN [23]	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 [60]	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 [11]	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[24]	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[24]	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[44]	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°
	AtLoc[61]	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°
	TransBoNet[49]	0.11,4.48°	**0.25**,12.5°	0.18,14.0°	0.20,**5.08**°	0.19,4.77°	0.17,**5.35**°	0.30,13.0°	0.20,8.45°
	Baseline APR w/ hypernet	**0.09**,4.47°	0.28,**9.44**°	**0.15**,**11.4**°	**0.16**,6.29°	**0.19**,4.62°	**0.18**,6.95°	0.28,**9.15**°	**0.19**,**7.48**°
MS-APR	MSPN[3]	**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[46]	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°
	MS-HyperPose	0.11,**4.34**°	**0.23**,9.79°	**0.13**,**10.7**°	0.17,6.05°	**0.16**,5.24°	**0.17**,6.86°	0.27,**6.00**°	**0.18**,**7.00**°

Table 4. **Comparative analysis of different APR architectures with and without the proposed hypernetwork to the Oxford RobotCar dataset:** We report the mean and median position/orientation error in meters/degrees.

	MapNet[9]		**AtLoc**[61]		**AtLoc w/ hypernet**	
Sequence	Mean	Median	Mean	Median	Mean	Median
LOOP1	8.76, 3.46	5.79, 1.54	8.61, 4.58	5.68, 2.23	**8.42, 3.30**	**4.91, 1.33**
LOOP2	9.84, 3.96	4.91, 1.67	8.61, 4.58	5.68, 2.23	**8.10, 3.04**	**4.73, 1.27**
FULL1	41.4, 12.5	17.94, 6.68	29.6, 12.4	11.10, 5.28	**29.58, 6.77**	**7.86, 1.38**
FULL2	59.3, 14.8	20.04, 6.39	48.2, 11.1	12.2, 4.63	**45.96, 9.60**	**8.38, 1.45**

We also assess the performance of two configurations of the baseline APR model, with and without hypernetworks on the Extended Cambridge dataset (ECL). Across all scenes and variations, the hypernetwork-enhanced model outperforms the baseline, demonstrating better adaptation to the seasonal changes and diverse lighting conditions inherent in the scenes. The detailed analysis elucidating the robustness and resilience of HyperPose to scene changes is in the supplementary material.

6. Conclusions

In this work, we proposed a hypernetwork-based approach to improve APR models. Our experiments show that hypernetworks improve accuracy for single-scene APR models. We also present MS-HyperPose, a novel multiscene camera pose regression model enhanced with a hypernetwork, which achieves new SOTA accuracy in indoor and outdoor localization datasets. We introduce the Extended Cambridge Landmarks (ECL) dataset to facilitate research on robust localization methods.

References

[1] Eduardo Arnold, Jamie Wynn, Sara Vicente, Guillermo Garcia-Hernando, Aron Monszpart, Victor Prisacariu, Daniyar Turmukhambetov, and Eric Brachmann. Map-free visual relocalization: Metric pose relative to a single image. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 690–708. Springer, 2022. 3

[2] Vassileios Balntas, Shuda Li, and Victor Prisacariu. Relocnet: Continuous metric learning relocalisation using neural nets. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 1, 8

[3] Hunter Blanton, Connor Greenwell, Scott Workman, and Nathan Jacobs. Extending absolute pose regression to multiple scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 38–39, 2020. 2, 3, 7, 8

[4] E. Brachmann and C. Rother. Learning less is more - 6D camera localization via 3D surface regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4654–4662, 2018. 1, 3

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

[6] E. Brachmann and C. Rother. Visual Camera Relocalization From RGB and RGB-D Images Using DSAC. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(09):5847–5865, 2022. 3, 7, 8

[7] Eric Brachmann, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. Uncertainty-driven 6D pose estimation of objects and scenes from a single rgb image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3364–3372, 2016.

[8] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother. DSAC - differentiable RANSAC for camera localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2492–2500, Los Alamitos, CA, USA, 2017. 1, 3

[9] Samarth Brahmbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. Geometry-aware learning of maps for camera localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3, 6, 7, 8

[10] T. Brooks, A. Holynski, and A. A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18392–18402, Los Alamitos, CA, USA, 2023. IEEE Computer Society. 6

[11] Ming Cai, Chunhua Shen, and Ian Reid. A hybrid probabilistic model for camera relocalization. In *Proceedings of the British Machine Vision Conference*, page 238, 2018. 2, 3, 7, 8

[12] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 213–229, Cham, 2020. 3, 5

[13] Yoav Chai, Raja Giryes, and Lior Wolf. Supervised and unsupervised learning of parameterized color enhancement. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 992–1000, 2020. 3

[14] Kefan Chen, Noah Snavely, and Ameesh Makadia. Wide-baseline relative camera pose estimation with directional learning. 2021 ieee. In *CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3257–3267, 2021. 1

[15] Shuai Chen, Tommaso Cavallari, Victor Adrian Prisacariu, and Eric Brachmann. Map-relative pose regression for visual re-localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20665–20674, 2024. 1, 2, 3, 8

[16] Yinbo Chen and Xiaolong Wang. Transformers as meta-learners for implicit neural representations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 170–187. Springer, 2022. 2, 3

[17] Mingyu Ding, Zhe Wang, Jiankai Sun, Jianping Shi, and Ping Luo. CamNet: Coarse-to-fine retrieval for camera re-localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 1, 8

[18] Siyan Dong, Qingnan Fan, He Wang, Ji Shi, Li Yi, Thomas Funkhouser, Baoquan Chen, and Leonidas J Guibas. Robust neural routing through space partitions for camera relocalization in dynamic indoor environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8544–8554, 2021. 2

[19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. 3

[20] Sovann En, Alexis Lechervy, and Frédéric Jurie. Rpnet: An end-to-end network for relative camera pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 1

[21] B. Glocker, S. Izadi, J. Shotton, and A. Criminisi. Real-time rgb-d camera relocalization. In *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 173–179, 2013. 6, 7, 8

[22] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016. 2, 3

[23] Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4762–4769, 2016. 2, 3, 7, 8

[24] A. Kendall and R. Cipolla. Geometric loss functions for camera pose regression with deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6555–6564, 2017. 2, 3, 5, 7, 8

[25] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In

Proceedings of the IEEE International Conference on Computer Vision (ICCV), pages 2938–2946, 2015. 2, 3, 6, 7, 8

[26] Z. Laskar, I. Melekhov, S. Kalia, and J. Kannala. Camera relocalization by computing pairwise relative poses using convolutional neural network. In *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 920–929, 2017. 1, 8

[27] Xiaotian Li, Shuzhe Wang, Yi Zhao, Jakob Verbeek, and Juho Kannala. Hierarchical scene coordinate classification and regression for visual localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11983–11992, 2020. 1, 2

[28] Gidi Littwin and Lior Wolf. Deep meta functionals for shape representation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1824–1833, 2019. 3

[29] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 10012–10022, 2021. 3

[30] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The oxford robotcar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017. 6, 7

[31] Luke Melaskyriazi. Efficientnet-pytorch, 2019. 5

[32] Iaroslav Melekhov, Juha Ylioinas, Juho Kannala, and Esa Rahtu. Image-based localization using hourglass networks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 870–877, 2017. 2

[33] Tayyab Naseer and W. Burgard. Deep regression for monocular camera-based 6-dof global localization in outdoor environments. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1525–1530, 2017. 2, 3, 7

[34] Yuval Nirkin, Lior Wolf, and Tal Hassner. Hyperseg: Patch-wise hypernetwork for real-time semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4061–4070, 2021. 2, 3

[35] N. Radwan, A. Valada, and W. Burgard. Vlocnet++: Deep multitask learning for semantic visual localization and odometry. *IEEE Robotics and Automation Letters*, 3(4): 4407–4414, 2018. 2

[36] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017. 3, 4, 5

[37] P. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12708–12717, 2019. 1, 2

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

[39] Paul-Edouard Sarlin, Ajaykumar Unagar, Mans Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, et al. Back to the feature: Learning robust camera localization from pixels to pose. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3247–3257, 2021.

[40] Torsten Sattler, Michal Havlena, Filip Radenovic, Konrad Schindler, and Marc Pollefeys. Hyperpoints and fine vocabularies for large-scale location recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2102–2110, 2015.

[41] T. Sattler, B. Leibe, and L. Kobbelt. Efficient effective prioritized matching for large-scale image-based localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1744–1756, 2017. 1, 2

[42] T. Sattler, Q. Zhou, M. Pollefeys, and L. Leal-Taixé. Understanding the limitations of cnn-based absolute camera pose regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3297–3307, 2019. 7, 8

[43] Yoli Shavit and Ron Ferens. Introduction to camera pose estimation with deep learning. In *arXiv preprint arXiv:1907.05272*, 2019. 2, 3, 5

[44] Yoli Shavit and Ron Ferens. Do we really need scene-specific pose encoders? In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 3186–3192. IEEE, 2021. 2, 3, 5, 7, 8

[45] Yoli Shavit, Ron Ferens, and Yosi Keller. Learning multi-scene absolute pose regression with transformers. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2713–2722, 2021. 2, 4, 5

[46] Yoli Shavit, Ron Ferens, and Yosi Keller. Learning multi-scene absolute pose regression with transformers. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. 2, 3, 7, 8

[47] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2013. 1

[48] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems (NIPS)*, 33:7462–7473, 2020. 3

[49] Xiaogang Song, Hongjuan Li, Li Liang, Weiwei Shi, Guo Xie, Xiaofeng Lu, and Xinhong Hei. Transbonet: Learning camera localization with transformer bottleneck and attention. *Pattern Recognition*, 146:109975, 2024. 2, 3, 8

[50] Przemysław Spurek, Artur Kasymov, Marcin Mazur, Diana Janik, Slawomir K Tadeja, J Tabor, T Trzciński, et al. Hyperpocket: Generative point cloud completion. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6848–6853. IEEE, 2022. 3

[51] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Ak-

ihiko Torii. Inloc: Indoor visual localization with dense matching and view synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(4):1293–1307, 2021. 1, 2

[52] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. pages 6105–6114, Long Beach, California, USA, 2019. 5

[53] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 6105–6114. PMLR, 2019. 2

[54] Shitao Tang, Sicong Tang, Andrea Tagliasacchi, Ping Tan, and Yasutaka Furukawa. Neumap: Neural coordinate mapping by auto-transdecoder for camera localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 929–939, 2023. 1, 2

[55] Yi Tay, Zhe Zhao, Dara Bahri, Don Metzler, and Da-Cheng Juan. HyperGrid transformers: Towards a single model for multiple tasks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. 2

[56] Akihiko Torii, Relja Arandjelović, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1808–1817, 2015. 7, 8

[57] Mehmet Ozgur Turkoglu, Eric Brachmann, Konrad Schindler, Gabriel J Brostow, and Aron Monszpart. Visual camera re-localization using graph neural networks and relative pose supervision. In *2021 International Conference on 3D Vision (3DV)*, pages 145–155. IEEE, 2021. 1

[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS)*, pages 5998–6008, 2017. 2, 5

[59] Johannes Von Oswald, Christian Henning, João Sacramento, and Benjamin F Grewe. Continual learning with hypernetworks. *arXiv preprint arXiv:1906.00695*, 2019. 3

[60] F. Walch, C. Hazirbas, L. Leal-Taixé, T. Sattler, S. Hilsenbeck, and D. Cremers. Image-based localization using lstms for structured feature correlation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 627–637, 2017. 2, 7, 8

[61] Bing Wang, Changhao Chen, Chris Xiaoxuan Lu, Peijun Zhao, Niki Trigoni, and Andrew Markham. Atloc: Attention guided camera localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10393–10401, 2020. 2, 3, 4, 5, 6, 7, 8

[62] Dominik Winkelbauer, Maximilian Denninger, and Rudolph Triebel. Learning to localize in new environments from synthetic training data. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5840–5846. IEEE, 2021. 1

[63] J. Wu, L. Ma, and X. Hu. Delving deeper into convolutional neural networks for camera relocalization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 5644–5651, 2017. 2

[64] Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better. *Advances in Neural Information Processing Systems (NIPS)*, 34:30392–30400, 2021. 4

[65] Meng Xu, Youchen Wang, Bin Xu, Jun Zhang, Jian Ren, Stefan Poslad, and Pengfei Xu. A critical analysis of image-based camera pose estimation techniques. *arXiv preprint arXiv:2201.05816*, 2022. 2

[66] Fei Xue, Xin Wang, Zike Yan, Qiuyuan Wang, Junqiu Wang, and Hongbin Zha. Local supports global: Deep camera relocalization with sequence enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 8

[67] Fei Xue, Xin Wu, Shaojun Cai, and Junqiu Wang. Learning multi-view camera relocalization with graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 7, 8

[68] Qunjie Zhou, Torsten Sattler, Marc Pollefeys, and Laura Leal-Taixe. To learn or not to learn: Visual localization from essential matrices. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3319–3326. IEEE, 2020. 3, 7, 8