International Journal of Computer Vision (2024) 132:2530–2550
https://doi.org/10.1007/s11263-023-01982-9

HSCNet++: Hierarchical Scene Coordinate Classification and Regression for Visual Localization with Transformer

Shuzhe Wang[1] · Zakaria Laskar[2] · Iaroslav Melekhov[1] · Xiaotian Li[1] · Yi Zhao[1] · Giorgos Tolias[2] · Juho Kannala[1]

Received: 10 February 2023 / Accepted: 25 December 2023 / Published online: 6 February 2024

Abstract
Visual localization is critical to many applications in computer vision and robotics. To address single-image RGB localization, state-of-the-art feature-based methods match local descriptors between a query image and a pre-built 3D model. Recently, deep neural networks have been exploited to regress the mapping between raw pixels and 3D coordinates in the scene, and thus the matching is implicitly performed by the forward pass through the network. However, in a large and ambiguous environment, learning such a regression task directly can be difficult for a single network. In this work, we present a new hierarchical scene coordinate network to predict pixel scene coordinates in a coarse-to-fine manner from a single RGB image. The proposed method, which is an extension of HSCNet, allows us to train compact models which scale robustly to large environments. It sets a new state-of-the-art for single-image localization on the 7-Scenes, 12-Scenes, Cambridge Landmarks datasets, and the combined indoor scenes.

Keywords Scene coordinate regression · Hierarchical classification · Visual localization · Transformers

1 Introduction

Estimating the six degrees-of-freedom (6-DoF) camera pose from a given RGB image is a key component in many computer vision systems such as augmented reality, autonomous driving, and robotics. Classical methods (Sattler et al., 2011, 2012, 2016a; Taira et al., 2018; Sarlin et al., 2019) establish 2D-2D(-3D) correspondences between query and database local descriptors, followed by PnP-based camera pose estimation. Although powerful, these methods are memory and computationally inefficient requiring to keep an immense amount of local image descriptors and to perform hierarchical descriptor matching in a RANSAC loop to infer camera pose.

On the other hand, end-to-end pose regression methods that directly regress the camera pose parameters are much faster and scalable (Kendall et al., 2015; Balntas et al., 2018; Chen et al., 2021; Shavit & Keller, 2022). However, such methods are significantly less accurate than the ones based on local descriptors. A better trade-off between accuracy and computational efficiency is offered by structured localization approaches (Brachmann et al., 2017; Brachmann & Rother, 2018, 2021; Shotton et al., 2013; Li et al., 2020; Wang et al., 2021). Structured methods are trained to learn an implicit representation of the 3D environment by directly regressing 3D scene coordinates corresponding to a 2D pixel location in a given input image. This directly provides 2D-3D correspondences and avoids storing and explicitly matching database local descriptors with the query. For small-scale scenes, the scene-coordinate regres-

Communicated by Xiaowei Zhou.

✉ Shuzhe Wang
Shuzhe.Wang@aalto.fi

Zakaria Laskar
laskazak@fel.cvut.cz

Iaroslav Melekhov
Iaroslav.Melekhov@aalto.fi

Xiaotian Li
Xiaotian.Li@aalto.fi

Yi Zhao
Yi.Zhao@aalto.fi

Giorgos Tolias
toliageo@fel.cvut.cz

Juho Kannala
Juho.Kannala@aalto.fi

[1] Aalto University, Espoo, Finland

[2] Visual Recognition Group, Faculty of Electrical Engineering, Czech Technical University in Prague, Prague 6, Czechia

sion methods work on par (Brachmann et al., 2021) or outperform (Brachmann & Rother, 2018, 2021) local image descriptors-based approaches. Nevertheless, the storage and computational benefits of structured-based methods are superior to their classical counterparts.

Existing scene-coordinate regression approaches (Brachmann et al., 2017; Brachmann & Rother, 2018, 2021) are designed to predict scene coordinates from a small local image patch that provides robustness to viewpoint changes. However, such methods are limited in applicability to larger scenes where ambiguity from visually similar local image patches cannot be resolved with a limited receptive field. Using larger receptive field sizes, up to the full image, to regress the coordinates can mitigate the issues from ambiguities by encoding more context. This, however, has been shown to be prone to overfitting the larger input patterns in the case of limited training data, even if data augmentation alleviates this problem to some extent (Li et al., 2018; Brachmann & Rother, 2021).

Increasing context by enlarging the receptive field while maintaining the distinctiveness of local descriptors or not overfitting is a challenging problem. We address this using a special network architecture, called HSCNet (Li et al., 2020), which hierarchically encodes scene context using a series of classification layers before making the final coordinate prediction. The overall pipeline is illustrated in Fig. 1. Specifically, the network predicts scene coordinates progressively in a coarse-to-fine manner, where predictions correspond to a region in the scene at the coarse level and coordinate residuals at the finest level. The predictions at each level are conditioned on both descriptors and predictions from the preceding level which is the key component in large scenes as we experimentally demonstrate in this work. This conditioning leverages FiLM (Perez et al., 2018) layers that allow to gradually increase the receptive field. The HSCNet approach utilizes CNNs to encode the descriptors and predictions. In this work, we extend this idea and propose the transformer-based (Vaswani et al., 2017) conditioning mechanism, named HSCNet++, which is more efficient in capturing global context into local representations through attention and does not require heavy conventional layers to enlarge the receptive field. The architecture manages to improve coordinate prediction at all levels, both coarse and fine. We integrate dynamic position information in the form of predicted coarse positional encoding, without the need to learn or construct explicitly position embeddings and show promising results on several camera relocalization benchmarks.

We further extend HSCNet++ by removing the dependency on dense ground truth scene coordinates. Dense coordinates limit the applicability of HSCNet to outdoor scenes. Similar to Brachmann and Rother (2018), HSCNet addressed the issue of sparse data on Cambridge dataset (Kendall et al., 2015) by using MVS-based densification (Schönberger et al., 2016). However, these methods either introduce additional noise and are costly to obtain. Directly training HSCNet with sparse supervision leads to a significant performance drop. In HSCNet++, we propose a simple yet effective pseudo-labelling method, where ground-truth labels at each pixel location are propagated to a fixed spatial neighbourhood. This is based on the assumption that nearby pixels share similar statistics. To provide robustness to pseudo-label noise, symmetric objective functions based on cross-entropy and re-projection loss are proposed. While the symmetric cross-entropy cost function provides robustness to the classification layers of HSCNet, the re-projection loss rectifies the noise in pseudo-labelled 3D scene coordinates.

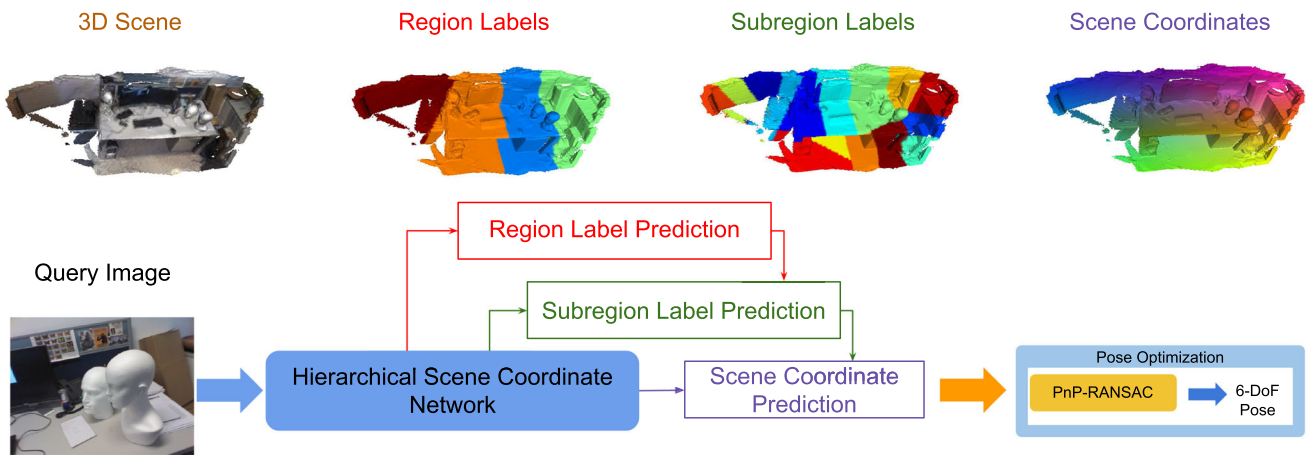


Fig. 1 HSCNet architecture. The ground-truth scene 3D coordinates are hierarchically quantized into regions and sub-regions. Direct branches of the network sequentially predict discrete regions and sub-regions, and continuous 3D coordinates, with the processing of each branch being conditioned on the result of the previous one. Given an input image, HSCNet predicts 3D coordinates for 2D image pixels, which then form the input to PnP-RANSAC for 6DoF pose estimation

This work is a summary and extension of HSCNet. We validate our approach on three datasets used in previous works: 7-Scenes (Shotton et al., 2013), 12-Scenes (Valentin et al., 2016), and Cambridge Landmarks (Kendall et al., 2015). Our approach demonstrates consistently better performance and achieves state-of-the-art results for single-image camera relocalization. In addition, by compiling the 7-Scenes and 12-Scenes datasets into single large scenes we show that our approach scales more robustly to larger environments. In summary, our contributions are as follows:

1. Compared to HSCNet, we utilize an improved transformer based conditioning mechanism that efficiently and effectively encodes global spatial information to scene coordinate prediction pipeline, resulting in a significant performance improvement from 84.8% to 88.7% on indoor localization while requiring only 57% of the memory footprint;
2. We extend HSCNet to optionally leverage the sparse ground truth only in the training procedure by introducing pseudo ground truth labels and angle-based re-projection errors. When using sparse supervision for training, the proposed approach achieves better performance on the Cambridge outdoor camera relocalization dataset compared to the MVS-based densified training data;
3. We show that the classical pixel-based positional encoding in our conditioning mechanism suffers from a significant performance drop, especially in scenes exhibiting substantial repetitive patterns. Our spatial positional encoding inspired by the FiLM layer eliminates this problem and achieves SoTA performance on several image-based localization benchmarks.

2 Related Work

Existing methods for visual localization are reviewed depending on the category they belong to.

Classical visual localization methods assume that a scene is represented by a 3D model, which is a result of processing a set of database images. Each 3D point of the model is associated with one or several database local descriptors. Given a query image, a sparse set of keypoints and their local descriptors are obtained using traditional (Calonder et al., 2010; Lowe, 2004; Rublee et al., 2011; Bay et al., 2006) or learned CNN-based (DeTone et al., 2018; Revaud et al., 2019; Dusmanu et al., 2019; Melekhov et al., 2021, 2020; Luo et al., 2019; Wang et al., 2020; Tian et al., 2017; Balntas et al., 2016; Zagoruyko & Komodakis, 2015; Han et al., 2015; Melekhov et al., 2017; Simo-Serra et al., 2015; Mishchuk et al., 2017) approaches. The query local descriptors are then matched with local descriptors extracted from database images to establish tentative 2D-3D matches. These tentative matches are then geometrically verified using RANSAC (Fischler & Bolles, 1981) and the camera pose is estimated via PnP. Although these methods produce a very accurate pose estimate, the computational cost of sparse keypoint matching becomes a limitation, especially for large-scale environments. The large computational cost is addressed by image retrieval-based methods (Arandjelović et al., 2016; Radenović et al., 2016) restricting matching query descriptors to local descriptors extracted from top-ranked database images only. Moreover, despite the recent advancements of learned keypoint detectors and descriptors (Wang et al., 2020; Dusmanu et al., 2019; Melekhov et al., 2020, 2021; Sun et al., 2021; Zhou et al., 2021; Revaud et al., 2019; Tyszkiewicz et al., 2020), extracting discriminative local descriptors which are robust to different viewpoint and illumination changes is still an open problem.

Absolute camera pose regression (APR) methods aim to alleviate the limitations of structure-based methods by using a neural network that directly regresses the camera pose of a query image (Kendall et al., 2015; Brahmbhatt et al., 2018; Kendall & Cipolla, 2016, 2017; Melekhov et al., 2017; Walch et al., 2017; Chen et al., 2021, 2022) that is given as input to the network. The network is trained on database images with ground-truth poses by optimizing a weighted combination of orientation and translation L2 losses (Kendall et al., 2015; Melekhov et al., 2017), leveraging uncertainty (Kendall et al., 2018), utilizing temporal consistency of the sequential images (Walch et al., 2017; Radwan et al., 2018; Valada et al., 2018; Xue et al., 2019) or using GNNs (Xue et al., 2020) and Transformers (Shavit et al., 2021). The APR methods are scalable, fast, and memory efficient since they do not require storing a 3D model. However, their accuracy is an order of magnitude lower compared to the one obtained by structure-based localization approaches and comparable with image retrieval methods (Sattler et al., 2019). Moreover, the APR approaches require a different network to be trained and evaluated per scene when the scenes are registered to different coordinate frames.

Relative camera pose regression (RPR) methods, in contrast to APR, train a network to predict relative pose between the query image and each of the top-ranked database images (Ding et al., 2019; Laskar et al., 2017; Balntas et al., 2018), obtained by image retrieval (Arandjelović et al., 2016; Radenović et al., 2016). The camera location is then obtained via triangulation from two relative translation estimations verified by RANSAC. This leads to better generalization performance without using scene-specific training. However, the RPR methods suffer from low localization accuracy similarly to APR.

Scene coordinate regression (SCR) methods learn the first stage of the pipeline in the structure-based approaches. Namely, either a random forest (Brachmann et al., 2016; Cavallari et al., 2020, 2017; Guzmán-Rivera et al., 2014;

Massiceti et al., 2017; Meng et al., 2017, 2018; Shotton et al., 2013; Valentin et al., 2015) or a neural network (Brachmann et al., 2017; Brachmann & Rother, 2018, 2019a, c, 2021; Budvytis et al., 2019; Bui et al., 2018; Cavallari et al., 2019; Li et al., 2018; Massiceti et al., 2017) is trained to directly predict 3D scene coordinates for the pixels and thus the 2D-3D correspondences are established. These methods do not explicitly rely on feature detection, description, and matching, and are able to provide correspondences densely. They are more accurate than traditional feature-based methods at small and medium scales, but usually do not scale well to larger scenes (Brachmann & Rother, 2018, 2019a). In order to generalize well to novel viewpoints, these methods typically rely on only local image patches to produce the scene coordinate predictions. However, this may introduce ambiguities due to similar local appearances, especially when the scale of the scene is large. To resolve local appearance ambiguities, we introduce element-wise conditioning layers to modulate the intermediate feature maps of the network using coarse discrete location information. We show this leads to better localization performance, and we can robustly scale to larger environments.

Joint classification-regression frameworks have been proven effective in solving various vision tasks. For example, Rogez et al. (2017, 2019) proposed a classification-regression approach for human pose estimation from single images. In Brachmann et al. (2016), a joint classification-regression forest is trained to predict scene identifiers and scene coordinates. In Weinzaepfel et al. (2019), a CNN is used to detect and segment a predefined set of planar Objects-of-Interest (OOIs), and then, to regress dense matches to their reference images. In Budvytis et al. (2019), scene coordinate regression is formulated as two separate tasks of object instance recognition and local coordinate regression. In Brachmann and Rother (2019a), multiple scene coordinate regression networks are trained as a mixture of experts along with a gating network which assesses the relevance of each expert for a given input, and the final pose estimate is obtained using a novel RANSAC framework, *i.e.* Expert Sample Consensus (ESAC). In contrast to existing approaches, in our work, we use spatially dense discrete location labels defined for all pixels, and propose FiLM-like (Perez et al., 2018) conditioning layers to propagate information in the hierarchy. We show that our novel framework allows us to achieve high localization accuracy with one single compact model.

Transformer has already shown a positive impact on the problem of visual localization. Shavit et al. (2021) show that multi-headed transformer architectures can be used to improve end-to-end absolute camera pose localization in multiple scenes with a single trained model. Similarly, SuperGlue, LoFTR and COTR (Sarlin et al., 2020; Sun et al., 2021; Jiang et al., 2021) demonstrate the usefulness of transformer architectures in learning local descriptor models. Inspired by the above success, the paper proposes methods to extend transformer architecture to the structured localization method.

3 Problem Formulation and Notation

The goal of camera pose estimation is to predict the 6-DoF pose $p(x) \in \mathbb{R}^6$ for an RGB image x. We adopt a standard two-step approach. As a first step, 3D coordinates are predicted for each pixel, or some of the pixels, of an image. Those are the coordinates from a known 3D scene. Such predictions result in a set of 2D-3D correspondences. As a second and final step, these correspondences are fed into the PnP algorithm that estimates the camera pose. In this work, we focus on the 3D coordinate prediction task.

We rely on a function $f : [0, 1]^{W \times H \times 3} \rightarrow \mathbb{R}^{w \times h \times 3}$, $w = W/8$ and $h = H/8$[1] that provides such coordinate predictions given an input image x of resolution equal to $W \times H$ pixels; the predicted coordinates for image x are given by $\hat{y}(x) = f(x)$.

The known 3D environment is represented by a set of training images, with known ground-truth labels per pixel in the form of 3D coordinates. The training set comprises pairs of the form $(x, y(x))$ for image x and ground-truth 3D coordinates $y(x)$. In case ground-truth is available only sparsely, *i.e.* on small part of the image pixels, a corresponding binary mask $m(x) \in \{0, 1\}^{w \times h}$ denotes which are the valid pixels. The value of ground-truth or prediction at a particular pixel is denoted by subscript i, *e.g.* $y(x)_i$ for the ground-truth coordinate of pixel i.

4 HSCNet++: Hierarchical Scene Coordinate Prediction with Transformers

4.1 Overview

A baseline *conventional* approach for this task is to use a fully convolutional network (FCN) that maps input images to 3D coordinate predictions and is trained with a regression loss. The proposed architecture extends this scheme by constructing a hierarchy of labels, from coarse-level to fine-level, and by adding extra layers to predict those labels. Hierarchical discrete labels are defined by partitioning the ground-truth 3D points of the scene with hierarchical k-means. The number of levels in the hierarchy is fixed to 2 in this work. In this way, in addition to the ground-truth 3D

[1] The spatial resolution of the prediction is smaller, by a factor of 8, than that of the input image. The coordinate predictions are provided for a down-sampled version of the image, which is aligned with the use of deep CNNs that inherently perform such down-sampling.

scene coordinates, each pixel in a training image is also associated with two discrete labels, namely region and sub-region labels, obtained at different levels of the clustering hierarchy. Region and sub-region labels are denoted by one-hot encodings $y_r(x) \in \{0,1\}^{w\times h\times k_1}$ and $y_s(x) \in \{0,1\}^{w\times h\times k_2}$, respectively. The fine-level information is given by the residual between the ground-truth 3D point and the corresponding sub-region center, which we denote by $y_{3D}(x) \in \mathbb{R}^{w\times h\times 3}$. Ground-truth 3D pixel coordinates $y(x)$ are replaced by $y_r(x)$, $y_s(x)$, and $y_{3D}(x)$. Sub-region centers and residuals, when combined by addition, compose the pixel 3D coordinates, *i.e.* $y(x) = c(y_r(x) \times k_2 + y_s(x)) + y_{3D}(x)$, where c is a function providing the sub-region center.

The proposed architecture includes two classification branches for regions and sub-regions, which provide the label predictions in the form of the k-dimensional probability distributions, and a regression branch for the residual prediction. Regions, sub-region and residual predictions are denoted by $\hat{y}_r(x)$, $\hat{y}_s(x)$, and $\hat{y}_{3D}(x)$, respectively. A key ingredient is to propagate coarse region information to inform the predictions at finer levels, which is achieved by conditioning layers before the classification/regression layers.

4.2 Preliminaries

We describe FiLM layers and transformer blocks, which we use in the proposed architecture.

The FiLM Perez et al. (2018) *conditioning layer* represents a block whose processing is conditioned on an auxiliary input. Conditioning relies on parameter generators γ, β to generate a set of scaling and shifting parameters $\gamma(l)$ and $\beta(l)$, the auxiliary input $l \in R^{w\times h\times d}$ is the (sub-)region label encoding. The conditioning is processed by

$$\phi(F, l) = \gamma(l) \odot F + \beta(l), \tag{1}$$

where $\odot$ is the Hadamard product, $F \in \mathbb{R}^{w\times h\times d}$ is the main input. Therefore, the parameters of the FiLM layer are conditioned on the auxiliary. The FiLM-based processing is a way to jointly encode the main and the auxiliary input. In the following, it is used to encode the predicted (sub-)regions information together with the image features.

Transformer We view a 3D activation tensor of size $w \times h \times d$ as a set of $w \times h$ vectors/tokens and provide them as input to transformer blocks. The vanilla transformer has the computational complexity that is quadratic in the cardinality $n = w \times h$ of input set, which is computationally unaffordable in our case. Inspired by prior work (Sun et al., 2021), we apply the linear transformer (Katharopoulos et al., 2020) that reduces the complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$ by using the associativity property of matrix products and replacing the exponential similarity kernel with a linear dot-product kernel.

Consequently, the transformer modules that are part of our architecture do not have a significant impact on run time.

4.3 HSCNet++ Architecture

This section presents the model architecture for HSCNet++ and discusses the difference compared to the original HSCNet architecture.

Overview of the model architecture The overall architecture of HSCNet++ is summarized in Fig. 2. We first present the model as it operates during inference and then clarify the differences between training and inference. An FCN backbone is used for dense feature encoding and is denoted by $\mathcal{F}(x) \in \mathbb{R}^{w\times h\times d}$. This is a mapping of the input image to a dense feature tensor which represents the appearance of the input image.

Prediction of region labels is performed first. A module $g_r : \mathbb{R}^{w\times h\times d} \rightarrow \mathbb{R}^{w\times h\times d}$ is used that consists of convolutional layers and a transformer block. Its input is feature map $\mathcal{F}(x)$ and the output is given by $\mathbf{x}_r = g_r(\mathcal{F}(x))$. Feature map processing is performed within the local context of the receptive field with convolutions and within a global context with the transformer. The region predictor $h_r : \mathbb{R}^{w\times h\times d} \rightarrow \mathbb{R}^{w\times h\times k_1}$ comprises a 1×1 convolutional layer and is used to obtain the region prediction denoted by $\hat{y}_r(x) = h_r(\mathbf{x}_r)$.

Then, sub-region prediction is performed. A module $g_s : \mathbb{R}^{w\times h\times d} \times \mathbb{R}^{w\times h\times k_2} \rightarrow \mathbb{R}^{w\times h\times d}$ is used, which consists of convolutional layers and transformer blocks, but also FiLM layers, therefore the two inputs. The main input is the feature map $\mathcal{F}(x)$, while the auxiliary input is the region prediction $\hat{y}_r(x)$ from the earlier stage. In practice, $\hat{y}_r(x)$ is passed through a series of convolutional layers before inputted to the FiLM layer as shown in Fig. 3 (c, middle block). Conditioning on region predictions is a way to jointly encode appearance and geometry which comes in the form of region prediction. Therefore, conditioning on region predictions is used to improve sub-region predictions. Then, $\mathbf{x}_s = g_s(\mathcal{F}(x), \hat{y}_r(x))$ is fed into the sub-region predictor $h_s : \mathbb{R}^{w\times h\times d} \rightarrow \mathbb{R}^{w\times h\times k_2}$ comprising a 1×1 convolution layer, whose output is denoted by $\hat{y}_s(x) = h_s(\mathbf{x}_s)$ and constitutes the sub-region prediction.

Now, residual prediction is performed. Similar to the earlier stage, feature map $\mathcal{F}(x)$ is processed by conditioning on the concatenation of region and sub-region predictions, *i.e.* $\hat{y}_r(x)$ and $\hat{y}_s(x)$. This is denoted by module $g_{3D} : \mathbb{R}^{w\times h\times d} \times \mathbb{R}^{w\times h\times (k_1+k_2)} \rightarrow \mathbb{R}^{w\times h\times d}$ and consists of convolutional and FiLM layers and transformer blocks. Similarly, as before, concatenated region and sub-region predictions are passed through a series of convolutional layers before being inputted to the FiLM layer as an auxiliary input (*c.f.*Fig. 3 (c, right block)). Then, $\mathbf{x}_{3D} = g_{3D}(\mathcal{F}(x), \hat{y}_s(x))$ is fed into the residual predictor to obtain $\hat{y}_{3D}(x) = h_{3D}(\mathbf{x}_{3D})$, where

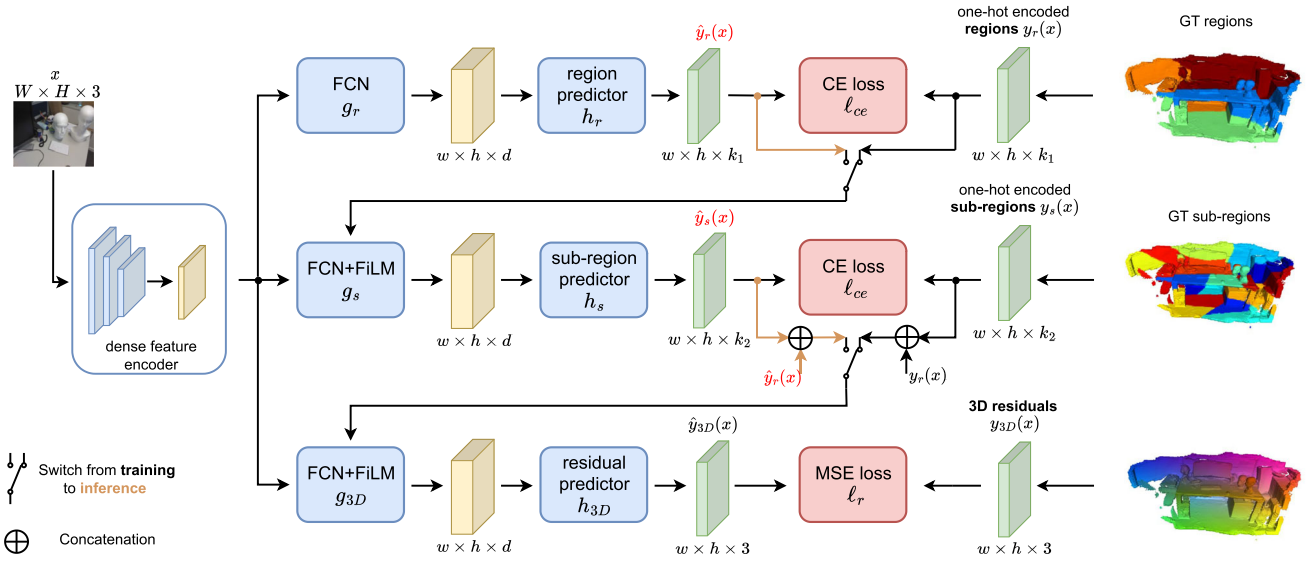


Fig. 2 An overview of the proposed HSCNet++. The figure shows the network architecture of the proposed HSCNet++. The depicted losses correspond to the case of learning with dense ground-truth. Note that the switch is applied during inference when the predicted labels are encoded instead of the ground-truth labels

$h_{3D} : \mathbb{R}^{w \times h \times d} \rightarrow \mathbb{R}^{w \times h \times 3}$ consists of a 1×1 convolution. The detailed architecture for HSCNet++ and the different modules is shown in Fig. 3.

Synergy between FiLM and transformers Modules g_s and g_{3D} include the use of FiLM layers followed by transformer blocks. Transformers typically rely on the use of 2D positional encodings (Vaswani et al., 2017) in order to take the position of activations into account. Discarding those positions is not an appropriate choice for our task. Nevertheless, our architecture design dispenses with the need for those classical positional encodings. This is due to the fact that FiLM layers jointly encode appearance with 3D coordinate predictions, instead of the 2D positions within the image. To the best of our knowledge, such form of geometry encoding for transformers has not appeared in the computer vision or machine literature before. We experimentally show that this is an effective design choice.

Compared to HSCNet, the proposed HSCNet++ incorporates the use of transformers. The design choice of placing them right after FiLM layers supports their synergy due to the mentioned case of encoding positions.

4.4 Training

When training with dense supervision, the following losses are adopted. Classification loss ℓ_c is applied to the output of the two classification branches,

$$\ell_c = \ell_{ce}(\hat{y}_r(x), y_r(x)) + \ell_{ce}(\hat{y}_s(x), y_s(x)) \tag{2}$$

Where ℓ_{ce} is the cross-entropy loss. Additionally, regression loss ℓ_r, in particular mean squared error, is applied on $\hat{y}_{3D}(x)$ and $y_{3D}(x)$. The total loss $\mathcal{L}$ is a weighted sum of the two classification losses and the regression loss.

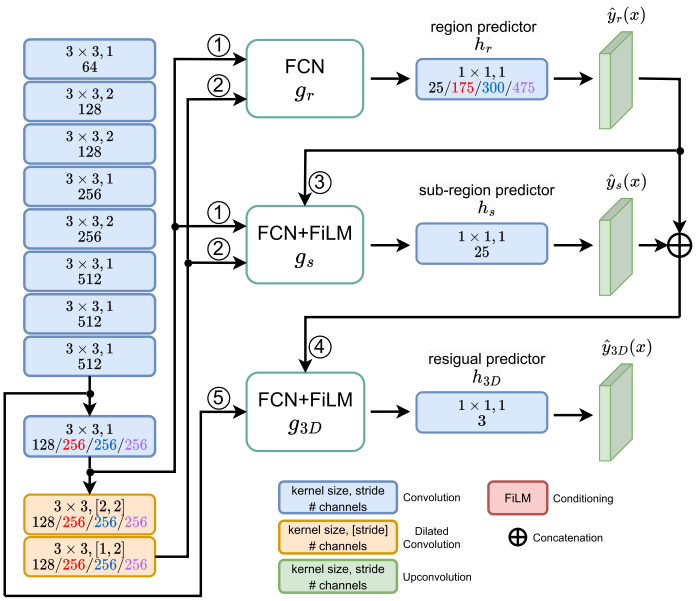
$$\mathcal{L} = \lambda_1 \ell_c + \lambda_2 \ell_r \tag{3}$$

Where λ_1 and λ_2 are the weights for each term. We observe that the regression prediction is more sensitive to localization performance. Thus, a larger weight is assigned to the ℓ_r.

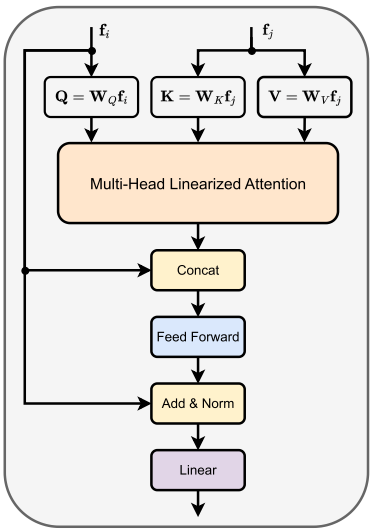
4.5 Inference

During inference, the predicted 3D coordinates $\hat{y}(x)$ and their corresponding 2D pixels are fed into the PnP-RANSAC loop to estimate the 6-DoF camera pose. These predicted 3D coordinates are obtained by simply summing the center of predicted sub-regions $c(y_r(x) \times k_2 + y_s(x))$ and predicted residuals $\hat{y}_{3D}(x)$.

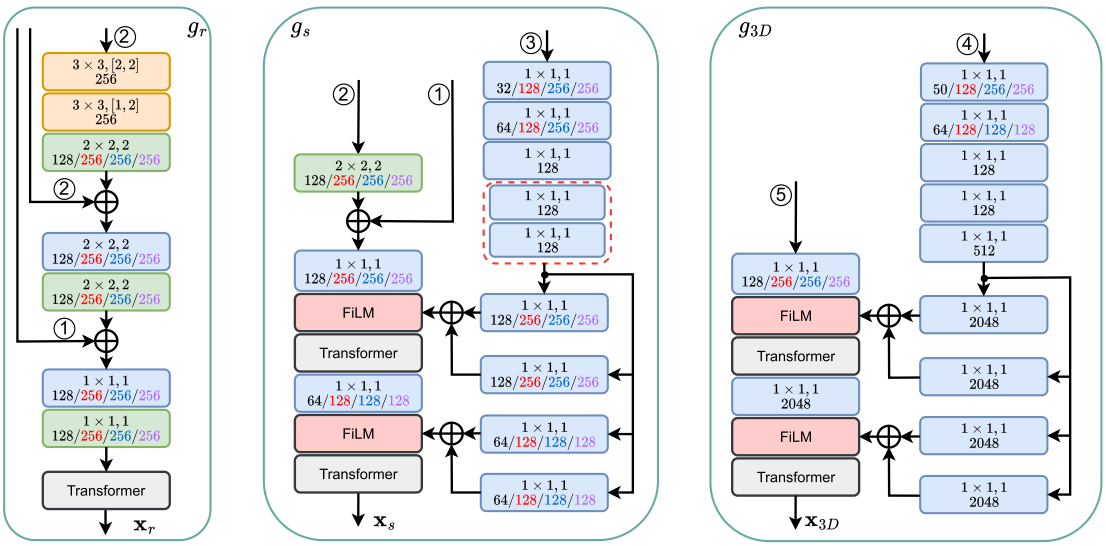
We differentiate on how conditioning is conducted during training and inference as shown in Fig. 2. At training time, conditioning is performed using the ground truth (sub-)region labels, *i.e.* $y_r(x)$ and $y_s(x)$ are the second inputs of the conditioning blocks. At test time, conditioning is implemented using predicted (sub-)region labels. Specifically, the one-hot encodings of the argmax operation of $\hat{y}_r(x)$ and $\hat{y}_s(x)$ are the second inputs of the conditioning blocks.



(a) Main network: overview



(b) Transformer



(c) Main network: detailed

Fig. 3 HSCNet++ detailed architecture. The figure shows the detailed network architecture of the main pipeline and the FiLM conditioning network. For experiments on the combined scenes we added two more layers in the first conditioning generator, g_s that are marked in (dotted) red. We also roughly doubled the channel counts that are highlighted in red, cyan and violet for i7-Scenes, i12-Scenes and i19-Scenes, respectively (Color figure online)

4.6 Training with Sparse Supervision

When only sparse ground truth of 3D coordinates, indicated by mask $m(x)$ for image x, is available, the straightforward approach is to apply the loss only on pixels where the mask value is 1, which we refer to as *valid pixel*. Instead, we propose to perform propagation of the available labels to nearby pixels and use two additional losses that are appropriately handling the scarcity of the labels. We refer to the HSCNet++ model trained with such sparse supervision as HSCNet++(S).

Label propagation (LP) We rely on a smoothness assumption: labels do not change much in a small pixel neighborhood. Consequently, we propagate the labels in a local neighborhood around each pixel. The neighborhood is defined by a square area of size $z \times z$. All neighbors of a valid pixel are marked as valid too and ground-truth maps, namely $y_r(x)$,

$y_s(x)$, and $y_{3D}(x)$, are updated by replicating the label of the original pixel to the neighboring pixels. Then, the classification and regression losses are applied to the newly obtained valid pixels after propagation. This is seen as some form of pseudo-labeling that increases the density of the available labels.

Symmetric cross-entropy loss (SCE) Pseudo-labels are expected to include noise. This noise will typically be larger if propagation reaches background pixels starting from a foreground-object valid pixel. We quantitatively analyze the percentage of noisy labels with the increasing of neighbor radius in Sect. 5.4. Thus, we face a challenging task which is learning correct classification with noisy labels. The traditional cross-entropy loss is not reliable in such a scenario as it exhibits overfitting to noisy labels on some "easy" classes and suffers from under learning on some “hard” classes (Wang et al., 2019).

Following (Wang et al., 2019), we increase the robustness of the classification with minimal cost by introducing the symmetric cross-entropy loss. The additional reverse cross entropy loss in SCE is a noise-tolerant term that exhibits the property of overestimating and underestimating the target value resulting in the same loss. This property makes it more adaptive to noisy labels and allows the model to cope better with label noise. The SCE loss is defined as a weighted summation of the following terms:

$$l_{sce} = \lambda_{ce} l_{ce} + \lambda_{sce} l_{rce} \tag{4}$$

Where l_{rce} is the reverse cross-entropy loss. For a valid pixel $i \in I$, the l_{rce} is:

$$\ell_{rce}(x, i) = \hat{y}_r(x)_i \log y_r(x)_i, \tag{5}$$

compared to the conventional one defined as follows:

$$\ell_{ce}(x, i) = y_r(x)_i \log \hat{y}_r(x)_i, \tag{6}$$

Re-projection error loss (Rep) Besides the SCE Loss to predict the correct labels from noise, we also adopt the re-projection loss as a semi-supervised term to further enhance both the labels and distance residual prediction. The loss term is especially efficient in scenes with a large amount of textureless or repeating patterns. However, the vanilla re-projection loss requires careful initialization to avoid the impact of unstable gradients from degenerate 3D predictions (*e.g.* too far or behind the camera). Training with vanilla re-projection loss requires extra geometric constraints and a long convergence time (Brachmann & Rother, 2021). Inspired by Li et al. (2018), we employ the angle-based re-projection loss which aims to minimize the angle θ between two rays that share the camera center. This strategy forces predictions to lie in front of the camera, ensuring smoother gradients during training. Consequently, it eliminates the need for a time-consuming initialization step and mitigates the burden of related geometric constraints.

Given ground-truth camera pose P, the loss for pixel i of image x, whose 2D coordinates in the image are denoted by p_i, is given by

$$\ell_{rep}(x, i) = ||\gamma_i P^{-1} \hat{y}_i(x) - f C^{-1} p_i||, \tag{7}$$

where $\gamma_i = ||f C^{-1} p_i|| / ||P^{-1} \hat{y}(x)_i||$, f is the focal length, and C is the intrinsic matrix. The angle-based re-projection loss is computed in the camera coordinate system between two points on a 3D sphere centered at the camera center and touching the image plane at the ground-truth pixel location, *i.e.* radius of the sphere is $||f C^{-1} p_i||$. The two points on the sphere correspond to the locations where the vector from the camera center to the predicted 3D point and ground-truth pixel location (both in camera coordinate system) intersect the 3D sphere represented by first and second terms in Eq. 7 respectively.

Note that the re-projection loss is not added to the total loss in the beginning epochs for a fast training convergence. Similar to our dense setting, the total loss for sparse supervision is the weighted summation of regression loss, symmetric classification loss, and re-projection loss, $\ell_{sparse} = \ell_{sce} + \lambda_2 \ell_r + \lambda_3 \ell_{rep}$.

5 Experiments

In this section, we discuss the experimental setup and employed datasets, present our results, and compare our approach to state-of-the-art localization methods.

5.1 Experimental Setup

Datasets We use three standard benchmarks for the evaluation; namely, 7-Scenes (Shotton et al., 2013), 12-Scenes (Valentin et al., 2016), and Cambridge Landmarks (Kendall et al., 2015), The 7-Scenes dataset covers a volume of $\sim 6m^3$ for each individual scene. The 3D models and ground truth poses are included in the dataset. 12-Scenes is another indoor RGB-D dataset that contains 4 large scenes with a total of 12 rooms, the volume ranges 14–$79m^3$ for each room. The union of these two datasets forms the 19-Scenes dataset. Cambridge Landmarks dataset is a standard benchmark for evaluating scene coordinate methods in outdoor scenes. It is a small-scale outdoor dataset consisting of 6 individual scenes, and the ground truth pose is provided by structure-from-motion.

Following prior work (Brachmann & Rother, 2019a), we conduct experiments per scene, *i.e.* the individual scenes setting, but also by training a single model on all scenes of a corresponding dataset, *i.e.* the combined scenes setting. The

Table 1 Indoor localization: individual scene setting (7-Scenes)

Method	7-Scenes														Accuracy
	Chess		Fire		Heads		Office		Pumpkin		Red Kitchen		Stairs		
	t, cm	r, °	t, cm	r, °	t, cm	r, °	t, cm	r, °	t, cm	r, °	t, cm	r, °	t, cm	r, °	
MapNet (Brahmbhatt et al., 2018)	8.0	3.30	27.0	11.70	18.0	13.30	17.0	5.20	22.0	4.00	23.0	4.90	30.0	12.10	–
Geometric PoseNet (Kendall & Cipolla, 2017)	13.0	4.50	27.0	11.30	17.0	13.00	19.0	5.60	26.0	4.80	23.0	5.40	35.0	12.40	–
AttTxf (Shavit et al., 2021)	11.0	4.66	24.0	9.60	14.0	12.19	17.0	5.66	18.0	4.44	17.0	5.94	26.0	8.45	–
LSTM-Pose (Walch et al., 2017)	24.0	5.80	34.0	11.90	21.0	13.70	30.0	8.10	33.0	7.00	37.0	8.80	40.0	13.70	–
AnchorNet(Saha et al., 2018)	6.0	3.90	16.0	11.10	9.0	11.20	11.0	5.40	14.0	3.60	13.0	5.30	21.0	11.90	–
LENS (Moreau et al., 2021)	3.0	1.30	10.0	3.70	7.0	5.80	7.0	1.90	8.0	2.20	9.0	2.20	14.0	3.60	–
AS (Sattler et al., 2016a)	3.0	0.87	2.0	1.01	**1.0**	0.82	4.0	1.15	7.0	1.69	5.0	1.72	4.0	1.01	68.7
HLoc (Sarlin et al., 2019)	2.0	0.85	2.0	0.94	**1.0**	0.75	3.0	0.92	5.0	1.30	4.0	1.40	5.0	1.47	73.1
PixLoc (Sarlin et al., 2021)	2.0	0.80	2.0	**0.73**	**1.0**	0.82	3.0	0.82	4.0	1.21	**3.0**	1.20	5.0	1.30	75.7
VS-Net (Huang et al., 2021)	**1.5**	**0.50**	**1.9**	0.80	1.2	**0.70**	2.1	**0.60**	3.7	1.00	3.6	1.10	**2.8**	**0.80**	–
SFT-CR (Guan et al., 2021)	2.1	0.70	2.0	0.78	1.1	0.81	2.4	0.66	3.4	0.98	3.4	**1.06**	3.5	0.97	–
DSAC++ (Brachmann & Rother, 2018)	2.0	**0.50**	2.0	0.90	**1.0**	0.80	3.0	0.70	4.0	1.10	4.0	1.10	9.0	2.60	76.1
DSAC*(3D) (Brachmann & Rother, 2021)	2.0	1.10	2.0	1.24	**1.0**	1.82	3.0	1.15	4.0	1.34	4.0	1.68	3.0	1.16	85.2
Reg-only (Li et al., 2020)	2.0	0.70	2.0	0.90	**1.0**	0.80	3.0	0.90	4.0	1.10	5.0	1.40	4.0	1.00	74.7
HSCNet (Li et al., 2020)	2.0	0.70	2.0	0.90	**1.0**	0.90	3.0	0.80	4.0	1.00	4.0	1.20	3.0	**0.80**	84.8
HSCNet++	2.0	0.63	2.0	0.79	**1.0**	0.80	**2.0**	0.65	**3.0**	**0.85**	**3.0**	1.09	3.0	0.83	**88.7**

For each scene of 7-Scenes dataset we report the median translation (t, cm) and orientation (r, °) error. The best and second best results are in **bold** and underlined. Note that except VS-Net (Huang et al., 2021) and SFT-CR (Guan et al., 2021), the rest results are reported in centimeter precision for translation error

Table 2 Indoor localization: individual scene setting (12-Scenes)

Scenes	Methods											
	Reg-only Li et al. (2020)			DSAC*(3D) Brachmann and Rother (2021)			HSCNet Li et al. (2020)			HSCNet++		
	t, cm	r, °	Acc	t, cm	r, °	Acc	t, cm	r, °	Acc	t, cm	r, °	Acc
Kitchen-1	0.8	**0.4**	**100**	–	–	–	0.8	**0.4**	**100**	**0.7**	**0.4**	**100**
Living-1	1.1	**0.4**	**100**	–	–	–	1.1	**0.4**	**100**	**1.0**	**0.4**	**100**
Bed	1.3	0.6	**100**	–	–	–	**0.9**	**0.4**	**100**	1.0	**0.4**	**100**
Kitchen-2	0.8	0.4	**100**	–	–	–	**0.7**	**0.3**	**100**	0.8	0.4	**100**
Living-2	1.4	0.6	**100**	–	–	–	**1.0**	**0.4**	**100**	**1.0**	**0.4**	**100**
Luke	2.0	0.9	93.8	–	–	–	**1.2**	**0.5**	96.3	1.3	0.6	**98.1**
Gate362	1.1	0.5	**100**	–	–	–	**1.0**	**0.4**	**100**	**1.0**	0.5	**100**
Gate381	1.6	0.7	98.8	–	–	–	1.2	0.6	**99.1**	**1.1**	**0.5**	98.6
Lounge	1.5	0.5	99.4	–	–	–	1.4	0.5	**100**	**1.3**	**0.4**	**100**
Manolis	1.4	0.7	97.2	–	–	–	**1.1**	**0.5**	**100**	1.2	**0.5**	**100**
Floor. 5a	1.6	0.7	97.0	–	–	–	**1.2**	**0.5**	**98.8**	1.3	**0.5**	96
Floor. 5b	1.9	0.6	93.3	–	–	–	1.5	0.5	97.3	**1.4**	**0.4**	**99.5**
Accuracy		96.4			99.1			99.1			**99.4**	

Similar to the 7-Scenes localization benchmark, we provide the median translation (t, cm), orientation (r, °) error, and accuracy with the error threshold of 5 *cm* and 5°. The best accuracy results are in **bold**

combined settings of the given indoor localization benchmarks are denoted by i7-Scenes, i12-Scenes, and i19-Scenes, respectively.

Competing methods In this work, we compare the proposed approach with the following methods: *(1)* pose regression methods that directly regress absolute or relative camera pose parameters: MapNet (Brahmbhatt et al., 2018), Geometric PoseNet (Kendall & Cipolla, 2017), AttTxf (Shavit et al., 2021), LSTM-Pose (Walch et al., 2017), AnchorNet (Saha et al., 2018) and LENS (Moreau et al., 2021); *(2)* local feature based pipelines based on SIFT such as Active Search (AS) (Sattler et al., 2016a) and HLoc (Sarlin et al., 2019) based on CNN descriptors; *(3)* DSAC*(3D) (Brachmann & Rother, 2021): the latest scene coordinate regression approach with 3D model; *(4)* VS-Net (Huang et al., 2021): scene-specific segmentation and voting; *(5)* PixLoc (Sarlin et al., 2021): scene-agnostic network; *(6)* SFT-CR (Guan et al., 2021): scene coordinate regression with global context-guidance. In addition, we also compare with *(7)* ESAC (Brachmann & Rother, 2019a) on the combined scenes. We also consider a baseline called *Reg-only* without the hierarchical classification layers.

Evaluation metrics We report the median translation and orientation error (cm,°) as well as the accuracy of test images under the threshold of ($5cm$, 5°) on indoor scenes. On Outdoor Cambridge Landmarks (Kendall et al., 2015), we report only the median pose error as in previous methods (Brachmann & Rother, 2021; Brachmann et al., 2017; Li et al., 2020).

Training details We generate the region labels by hierarchical K-means. For 7-Scenes, 12-Scenes, and Cambridge landmarks, we adopt 2-level ground truth labels with a branching factor of 25 for all the levels. Furthermore, for combined scenes, i7-Scenes, i12-Scenes, and i19-Scenes, the first level branching factor is set to 7×25, 12×25, and 19×25, respectively. For the individual scene setting, training is performed for 300K iterations with Adam optimizer. For the combined scenes the number of iterations is set to 900K. Throughout all experiments, we use a batch size of 1 with the initial learning rate of 10^{-4}.

The classification loss weights λ_1 is set to 1 for all datasets, while regression loss weight λ_2 is 10 for single scenes and 10^5 for combined scenes. In the sparse supervision setting, λ_{ce} and λ_{rce} are set to 0.1 and 1, respectively, while λ_2 follows the dense setting, and λ_3 is increased from 0 to 0.1 after first 10 epochs. We initialize the network by training with l_r using pseudo-label coordinates and later also add l_{rep} after 10 epochs. When training with sparse supervision, we select the neighborhood size $z = 11$ to propagate labels, and use the cluster centers obtained from dense scene coordinates for a direct comparison.

Data augmentation is also effective in increasing the final accuracy. Thus, similar to HSCNet (Li et al., 2020), we randomly augment training images using translation, rotation, scaling and shearing by uniform sampling from [−20%, 20%], [−30°, 30°], [0.7, 1.5], [−10°, 10°] respectively. In addition, images are augmented with additive brightness uniformly sampled from [−20, 20].

Table 3 Indoor localization: combined scene setting

Method	Localization Accuracy (%)		
	i7-Scenes	i12-Scenes	i19-Scenes
Reg-only (Li et al., 2020)	37.9	5.0	5.7
ESAC (Brachmann & Rother, 2019a)	70.3	97.1	88.1
HSCNet (Li et al., 2020)	83.3	99.3	92.5
HSCNet++	**88.3**	**99.5**	**93.6**

The table presents average localization accuracy under $5cm/5°$ of baseline models and proposed methods on i7-Scenes, i12-Scenes, and i19-Scenes datasets

Pose estimation We follow the same PnP-RANSAC pipeline and parameters setting as in Brachmann and Rother (2018). The inlier threshold and the softness factor are set to $\tau = 10$ and $\beta = 0.5$, respectively. We randomly select 4 correspondences to formulate a minimal set for a PnP algorithm to generate a camera pose hypothesis, and a set of 256 initial hypotheses are sampled. Similar to Brachmann and Rother (2018, 2021), a pose refinement process is performed until convergence for a maximum of 100 iterations.

Architecture details The detailed architecture of HSCNet++ is shown in Fig. 3; we also visualize the block details of the FiLM conditioning network and the transformer modules. By removing the transformer layers, we derive the architecture of HSCNet. Additionally, the number of channels in the last branch, g_{3D} of HSCNet is 4096, while it is 2048 for HSCNet++ that reduces memory cost (*c.f.*Sect. 5.6). For experiments on the combined scenes we added two more layers in the first conditioning generator, g_s that are marked in (dotted) red. We also roughly doubled the channel counts that are highlighted in red, cyan and violet for i7-Scenes, i12-Scenes and i19-Scenes, respectively. For individual scenes, we add 2 multi-head attention layers (MHA) to both classification and regression conditioning blocks, while in the combined setting, the number of MHA is set to 5.

5.2 Results for HSCNet and HSCNet++

Individual scenes setting. We present results on 7-Scenes and 12-Scenes in Table 1 and Table 2, accordingly. All models are trained and evaluated individually on each scene of the corresponding dataset. Results show that HSCNet is still competitive with respect to methods published later. With the addition of transformers, HSCNet++ further boosts the average performance by 4% on 7-Scenes and obtains the best accuracy on 7-Scenes among the competitors.

Combined scenes setting To test the scalability of scene-coordinate regression methods, we go beyond small-scale environments such as individual scenes in 7-Scenes and 12-Scenes and use the combined scenes, *i.e.* i7-Scenes, i12-Scenes, and i19-Scenes by combining the former datasets.

Results on the combined scenes setting presented in Table 3 including comparison with the regression-only baseline and ESAC. Results show that our method scales well with increase in number of scenes compared to *Reg-only* baseline. It is to be noted that ESAC requires training and storing multiple networks specializing in local parts of the environment, whereas our approach requires only a single model. Results show that our approach outperforms ESAC on i7-Scenes and i12-Scenes, while performing comparably on i19-Scenes (87.9% *vs*.88.1%). ESAC and our approach could be combined for very large-scale scenes, but we do not explore this option in this work. HSCNet++ advances the state-of-the-art on all datasets, demonstrating the utility of transformers for this task.

Cambridge Landmarks Table 4 reports the results of three types of visual localization methods on Cambridge landmarks. AS (Sattler et al., 2016a) and HLoc (Sarlin et al., 2019) estimate the camera poses with sparse SfM ground truth. DSAC++, DSAC* and our approaches train a scene-coordinate regression model with MVS-densified depth maps, VS-Net leverages the hybrid of the two. Both HSCNet and HSCNet++ perform better than other scene coordinate methods DSAC++ and DSAC*. The performance is comparable to more recent approaches. However, we observe that the models trained with MVS-densified pseudo ground truth show a slightly worse performance compared to the approaches that use the sparse SfM 3D map. HSCNet++ shows even worse performance by adding the transformer modules. Such results motivated us to extend the HSCNet++ to train with sparse supervision and our hypothesis is that the MVS densification introduces more noise to the dense supervision. The performance of HSCNet++(S) trained with sparse supervision on Cambridge landmarks in Sect. 5.5 verified our hypothesis.

5.3 Ablations: HSCNet

Data augmentation Using geometric and color data augmentation provides robustness to lighting and viewpoint changes (DeTone et al., 2018; Melekhov et al., 2021). We investigate the impact of data augmentation and summarize the obtained results in Table 5a. Applying data augmentations leads to better localization accuracy. Note that without data augmentation, the proposed approach still provides compara-

Table 4 Outdoor localization: individual scene setting (Cambridge)

Method	Cambridge									
	Kings College		Great Court		Old Hospital		Shop Facade		St Mary Church	
	t, cm	r, °	t, cm	r, °	t, cm	r, °	t, cm	r, °	t, cm	r, °
AS (Sattler et al., 2016a)	24	0.13	13	0.22	20	0.36	**4**	0.21	8	0.25
HLoc (Sarlin et al., 2019)	16	**0.11**	**12**	0.20	**15**	**0.30**	**4**	**0.20**	**7**	**0.21**
PixLoc (Sarlin et al., 2021)	14	0.24	30	0.14	16	0.32	5	0.23	10	0.34
VS-Net (Huang et al., 2021)	16	0.20	22	**0.10**	16	**0.30**	6	0.30	8	0.30
DSAC++ (Brachmann & Rother, 2018)	**13**	0.40	40	0.20	20	0.30	6	0.30	13	0.40
DSAC*(3D) (Brachmann & Rother, 2021)	15	0.30	49	0.30	21	0.40	5	0.30	13	0.40
HSCNet (Li et al., 2020)	18	0.30	28	0.20	19	**0.30**	6	0.30	9	0.30
HSCNet++	19	0.34	39	0.23	20	0.31	6	0.24	9	0.27

For each scene of the dataset we report the median translation (t, cm) and orientation (r, °) error. The best results are in **bold**

ble results to state of the art methods (*c.f.*ESAC (Brachmann & Rother, 2019a) in Table 3 *vs*.row 3 of Table 5a).

Conditioning mechanism The two key components of HSCNet are the coarse-to-fine joint classification-regression module and its combination with the conditioning mechanism. Their impact is evaluated and results are shown in Table 5a. We train a variant of our network without the conditioning mechanism, *i.e.* we remove all the conditioning generators and layers. The network still estimates scene coordinates in a coarse-to-fine manner by using the predicted location labels, but there is no coarse location information that is fed to influence the network activations at the finer levels. Results indicate the importance of the conditioning mechanism for accurate scene coordinate prediction. Compared to single scene setting in Tables 1 and 3, the performance of regression only baseline drops significantly in the combined scene setting as shown in Table 5a.

Hierarchy and partition granularity The robustness of HSCNet to the label hierarchy hyperparameter by varying depth and width are reported in Table 5. The results show that the performance of our approach is robust *w.r.t.* the choice of these hyperparameters, with a significant drop in performance observed only for the smallest 2-level label hierarchy. Increasing the number of classification layers from 2 is not always beneficial and only brings marginal improvement in 7-Scenes, while increasing the computational costs. We observe the best trade-off for the partition of 25×25 for both 7-Scenes and 175×25 for i7-Scenes ($175 = 7 \times 25$ due to 7 scenes combined).

5.4 Ablations: HSCNet++

Impact of internal transformer encoder layers. In this ablation, we remove transformers encoders t_r and t_s, while only t_{3D} remains. This variant is denoted by HSCNet++† and Table 6a shows a small to noticeable drop in all cases.

To factor out the impact of multi-headed attention (MHA) layers, we report results in Table 6a, which shows that increasing the number of MHA layers in HSCNet++† does not lead to performance improvement. It is worth mentioning that HSCNet++† with 8 MHA layers has 2 million more parameters than HSCNet++. Our intuition is that this happens due to the improvement of predictions at coarse levels of the network. To test the above hypothesis, we compute the accuracy of the sub-region predictions. For each valid pixel in a query image, this metric evaluates whether the valid pixel is correctly classified. Results in Table 6c show that adding transformers at classification branches helps to improve the label classification accuracy. However, the sub-region prediction accuracy does not always correlate with the localization performance. This can be attributed to RANSAC-based filtering of final 3D scene coordinates for camera pose estimation. That is, incorrect 3D scene predictions due to erroneous sub-region predictions can be detected as outliers by RANSAC.

Impact of positional encoding. We compare the proposed way of providing region (position) information to the transformer blocks with the classical positional encoding used in transformers. As label encoding is an inherent part of HSCNet, for a direct comparison with positional encoding, we additionally add the positional encoding right before the transformer block and perform experiments on i7-Scenes. Results presented in Table 6b show that with the additional position encoding the results noticeably drop.

5.5 Results for HSCNet++(S)

We now present results for HSCNet++(S) with sparse supervision and study the pseudo-labeling and loss functions in detail. For indoor scenes, we synthetically sparsify dense coordinates using sparse SIFT-based SfM reconstruction. That is, we select the subset of dense 3D coordinates whose 2D re-projections (pixel locations) are also registered in the

Table 5 Ablation for HSCNet

(a) Data augmentation and conditioning mechanism Method	Localization Accuracy (%)		
	i7-Scenes	i12-Scenes	i19-Scenes
HSCNet (Li et al., 2020)	83.3	99.3	92.5
w/o conditioning	70.3	97.1	88.1
w/o augmentation	71.5	98.7	87.9
Reg-only	37.9	5.0	5.7

(b) Label hierarchy: 7-Scenes dataset Label hierarchy	Accuracy, %
9×9	82.9
49×49	85.0
10×100×100	85.9
10×100×100×100	85.5
625	85.3
25×25	84.8

(c) Label hierarchy: i7-Scenes dataset Label hierarchy	Accuracy, %
63×9	80.6
343×49	83.7
70×100×100	83.0
70×100×100×100	82.1
7×25×25	83.0
175×25	83.3

Average pose accuracy obtained with different hierarchy settings. The models with 4-level label hierarchy are classification-only, *i.e.* the final regression layer is omitted

Table 6 Ablations for HSCNet++. We analyze the influence of different design choices of the proposed approach on i7-Scenes

Method	#MHA	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs	Average
HSCNet++†	5	95.3	96.0	98.4	88.6	63.7	71.8	80.4	84.9
HSCNet++†	8	94.9	95.3	98.4	88.3	63.7	70.0	79.5	84.3
HSCNet++	5	**98.2**	**96.6**	**99.6**	**90.8**	**72.1**	**76.8**	**83.7**	**88.3**

(a) The impact of increasing the number of MHA layers #MHA. Without intermediate transformers at the classification branches (only t_{3D} is used), adding additional #MHA layers to HSCNet++† does not improve performance.

Method	Encoding	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs	Average
HSCNet++	w/ PE	87.4	60.1	80.3	79.0	66.9	67.4	17.7	65.5
HSCNet++		**98.2**	**96.6**	**99.6**	**90.8**	**72.1**	**76.8**	**83.7**	**88.3**

(b) Positional encoding. PE: conventional positional encoding with sine and cosine functions.

Method	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs	Average
HSCNet	75.1	65.6	63.0	77.2	66.1	72.4	51.5	67.3
HSCNet++†	76.8	67.4	61.5	78.5	67.2	73.5	56.0	68.7
HSCNet++	**77.1**	**69.6**	**67.1**	**79.5**	**70.1**	**75.5**	**56.1**	**70.7**

(c) Sub-region prediction accuracy (%). Results show that HSCNet++ improves classification accuracy at the sub-region level.

Bold values highlight the architecture that gives the best results

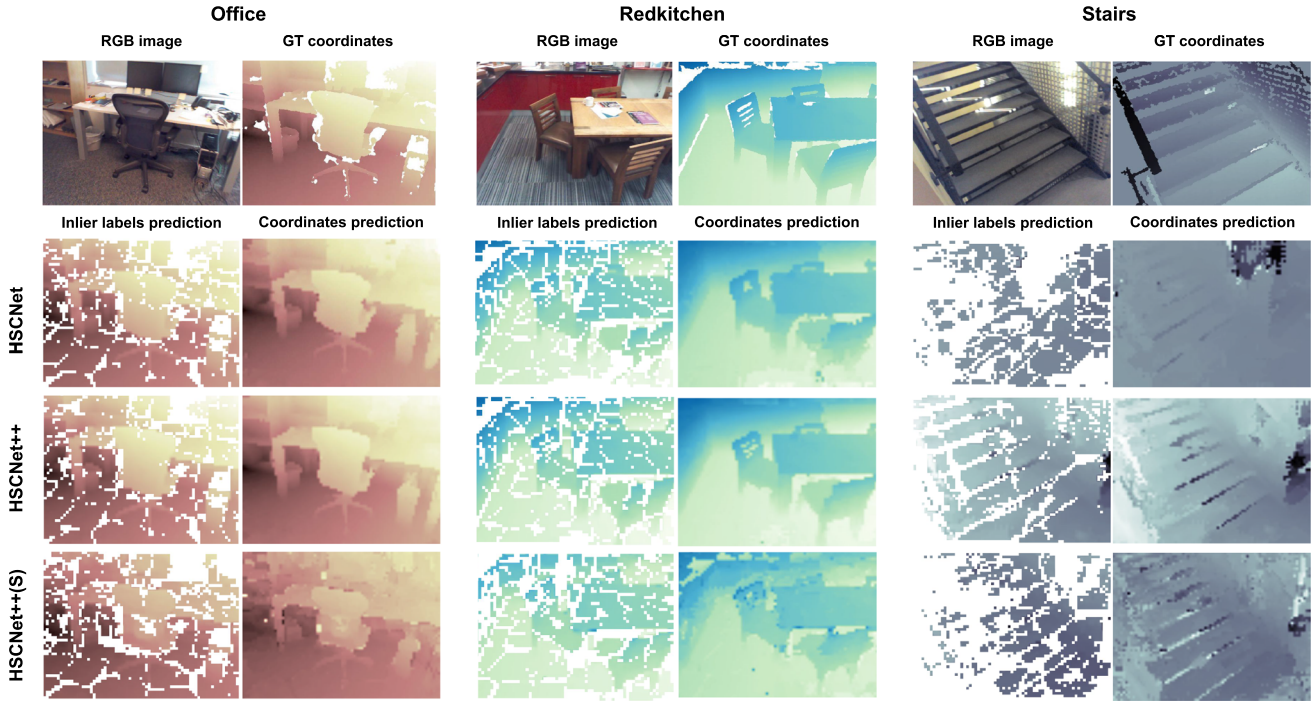


Fig. 4 Scene coordiantes visualization on i7-Scenes. We visualize the scene coordinate predictions for three test images with HSCNet, HSCNet++, and HSCNet++(S) on i7-Scenes. The XYZ coordinates are mapped to the heatmap, and the ground truth scene coordinates are computed from the depth maps. For each image, the left column is the correct predicted label and the right column is the predicted scene coordinates

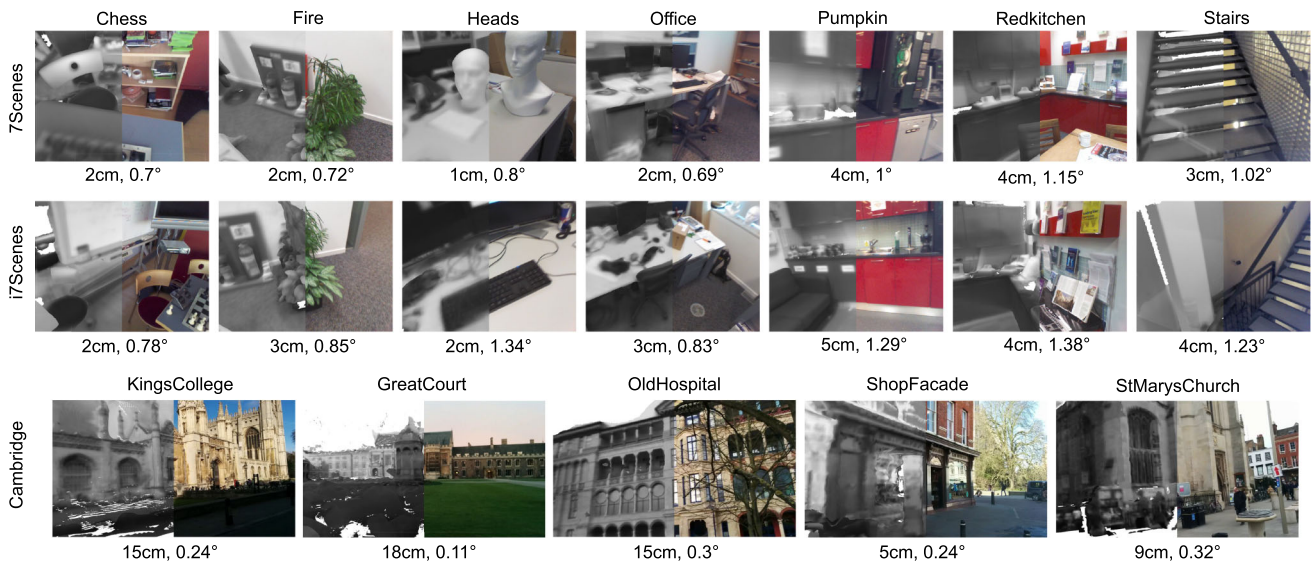


Fig. 5 Median Error for HSCNet++(S). We show the frames with median pose estimation error in each scene and visualize the accuracy by overlaying the query image (right) with a rendered image (left, grayscale) using the estimated pose and the ground truth 3D model

Table 7 HSCNet++(S) results

Method	Localization		
	Accuracy (%) ↑		Error (cm/°) ↓
	7-Scenes	i7-Scenes	Cambridge
HSCNet	84.8	83.3	16.0 / 0.28
HSCNet++	**88.7**	**88.3**	18.6 / 0.28
HSCNet++(S)	85.2	78.5	**12.4 / 0.24**

The table presents average localization accuracy (%) under $5cm/5°$ and average median pose error (cm/∘) of HSCNet++(S) and dense counterparts on 7-Scenes, i7-Scenes and Cambridge

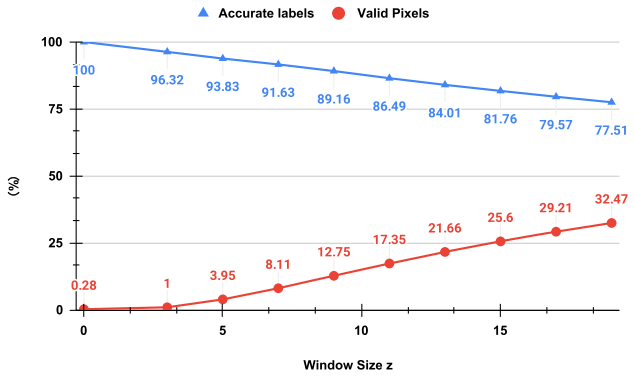


Fig. 6 Impact of neighborhood size z. The percentage of accurate labels and valid pixels change with the increasing of neighborhood window size z

SfM reconstruction. For the outdoor Cambridge dataset, we directly obtain the keypoints of training images from the provided SfM models.

The localization performance on 7-Scenes, i7-Scenes, and Cambridge datasets is provided in Fig. 5 and Table 7. Results show that even with sparse coordinate supervision, HSCNet++(S) achieves competitive results on 7-Scenes with respect to the dense counterpart, even outperforming HSCNet. On the more challenging combined scene setup of i7-Scenes, HSCNet++(S) lacks by 10% indicating a further requirement for future research in this direction. However, on the outdoor dataset Cambridge Landmarks, where only sparse coordinate data is available in most cases, HSCNet++(S) outperforms HSCNet and HSCNet++, which are trained on MVS-densified (Brachmann & Rother, 2018; Schönberger et al., 2016; Li et al., 2020) data, by a large margin. It demonstrates the effectiveness of our label propagation and supports our hypothesis that noisy dense ground truth from MVS harms the training process. The largest improvement is observed on Kings College, Great Court and Old Hospital with median pose errors ($cm/\circ$) of 15/0.24, 18/0.11 and 15/0.30 respectively (*c.f.*Table 4). On average median pose error, HSCNet++ (S) outperforms PixLoc (15/0.25), VSNet (13.6/0.24) and DSAC* (20.6/0.34).

Table 9 Impact of z on pose estimation

Methods	Scenes				
	Red Kitchen			GreatCourt	
	t, cm	r, °	Accuracy, %	t, cm	r, °
$z = 0$	6	1.37	65.5	32	0.28
$z = 7$	4	1.14	70.3	18	0.11
$z = 11$	4	1.15	72.9	18	0.11
$z = 15$	4	1.12	71.7	21	0.14
$z = 19$	3	1.12	73.0	35	0.20

We report the pose estimation results (median errors and accuracy) on Red Kitchen and Great Court with different neighborhood window size

Component ablations We formulate ablations on 7-Scenes to examine the components in the proposed HSCNet++(S). We first train the model without the proposed label propagation, *i.e.* only with sparse keypoint pixels only as the baseline. Then, for the HSCNet++(S), we present three variants by removing each component - transformers, symmetric cross-entropy and re-projection loss in HSCNet++(S) as shown

Table 8 Ablations for HSCNet++(S)

	Method	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs	Average
Error	HSCNet++(S)	2 / 0.70	2 / 0.72	1 / 0.8	2 / 0.69	4 / 1.00	4 / 1.15	3 / 1.02	–
	w/o LP	3 / 0.86	3 / 0.91	3 / 1.47	5 / 1.15	6 / 1.37	5 / 1.39	7 / 1.91	–
	w/o Txf	2 / 0.70	2 / 0.94	1 / 0.76	3 / 0.78	4 / 1.12	4 / 1.2	3 / 1.01	–
	w/o SCE	2 / 0.75	2 / 0.77	1 / 0.85	3 / 0.71	4 / 1.04	4 / 1.14	4 / 1.03	–
	w/o Rep	2 / 0.70	2 / 0.80	1 / 0.93	3 / 0.81	4 / 1.09	4 / 1.35	5 / 1.32	–
Accuracy	HSCNet++(S)	98.1	97.0	98.8	88.2	65.1	72.9	76.6	85.2
	w/o LP	86.0	81.1	85.4	56.0	39.4	49.6	36.2	62.0
	w/o Txf	97.3	98.8	99.6	85.6	59.4	64.4	80.5	83.7
	w/o SCE	97.6	96.2	96.5	84.2	64.1	70.1	73.2	83.1
	w/o Rep	97.5	98.2	96.8	80.2	62.8	64.8	55.0	79.3

The results of HSCNet++(S) and various variants are presented, the table shows the median translation and rotation errors (Error) and localization accuracy (Accuracy) under 5 cm/5°

Table 10 Comparison of the model capacity and runtime

Dataset	7-Scenes		i7-Scenes	
	HSCNet	HSCNet++	HSCNet	HSCNet++
Model Size, Mb	147.9	84.5	163	113.5
Training time, ms/iter	∼125	∼89	∼135	∼133
Inference time, ms/query	∼85–130	∼85-130	∼85-130	∼85-130

We compare the statistics of the model of HSCNet and HSCNet++, we provide the results on the same software and hardware setting

in Table 8. The baseline achieves only 62.0% on average accuracy which is significantly worse than our result (85.2%). Variants without the use of transformer layers (*w/o* **Txf**), **SCE** and **Rep** models show worse performance compared to HSCNet++(S) on average. Results demonstrate that the synergy of individual components leads the superior results.

Impact of LP neighborhood size. In this section, we analyze the impact of the LP neighborhood window size, z. We vary the neighborhood size z range from $0 \rightarrow 19$ on RedKitchen as ablation, and the results are reported in Fig. 6 and Table 9. Figure 6 shows that increasing the size of z, also increases pseudo-label noise shown by a decrease in the percentage of accurate labels. For *e.g.* when $z = 11$ the fraction of noisy labels is 15%. Results in Table 9 shows that there is a trade-off between increasing z, and camera localization accuracy. This effect is more pronounced in the outdoor scene, Great Court from the Cambridge dataset, where increasing z from $0 \rightarrow 11$ reduces median pose error (**t/r**) from $32/0.28 \rightarrow 18/0.11$. But increasing z further from $11 \rightarrow 19$ increases median pose error from $18/0.11 \rightarrow 35/0.2$. Limiting the spatial proximity of pseudo-labels to initial sparse labels seems a suitable choice.

5.6 Model Capacity and Efficiency

Model capacity As mentioned in Sect. 4.3, we prune some heavy convolution layers compared to HSCNet. To demonstrate the efficiency of this setting, Table 10 reports the model size of HSCNet and HSCNet++ on 7-Scenes and i7-Scenes. Our method has a memory footprint reduction of 43% compared to HSCNet on the individual scene training and 30% reduction on the combined scenes.

Runtime For a fair comparison of the running time, we run all the experiments on NVIDIA GeForce RTX 2080 Ti GPU and AMD Ryzen Threadripper 2950x CPU. It takes ∼7.4 h for 300k iterations on individual scene training for HSCNet++ and ∼10.4 h on HSCNet with the same setting. We show the approximate training time for one iteration in Table 10. It is clear that HSCNet++ has a smaller memory footprint and faster training time while offering higher accuracy. We also notice that the training time grows with the number of multi-head attention layers increases.

We have not observed a clear difference between the two methods in the inference running time. The running time varies from around 85 ms to 130 ms to localize one image. This is mainly dependent on the accuracy of predicted 2D-3D correspondences fed into the RANSAC-PnP loop.

6 Conclusion

We have proposed a novel hierarchical coarse-to-fine approach for scene coordinate prediction. The network benefits from FiLM-like conditioning of coarse region predictions for better scene coordinate prediction. Experimentally we demonstrate that both hierarchical and prediction conditioning are required for improvement. The method is extended to handle sparse labels using the proposed pseudo-labeling approach. Adaptation of symmetric cross-entropy and re-projection losses provides robustness to pseudo-label noise. We also show that the synergy of each component proposed in this work is needed for the best performance.

Results show that the proposed hierarchical scene coordinate network is more accurate than previous regression only approaches for single-image RGB localization. The proposed method is also more scalable as shown by results on three indoor datasets. In addition, the proposed method is extended to handle sparse labels using less costly methods than existing methods and obtaining better results on outdoor scenes.

Acknowledgements This work was supported by the Academy of Finland (grant No. 327911, 353138), Junior Star GACR (Grant No. GM 21-28830M) and Programme Johannes Amos Comenius CZ.02.01.01/00/22_010/0003405. We acknowledge the computational resources provided by the Aalto Science-IT project, CSC-IT Center for Science, Finland, and OP VVV funded project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics". We thank Dr. Jakob Verbeek for contributing to the HSCNet publication.

Funding Open Access funding provided by Aalto University.

Data Availibility The datasets generated during and/or analysed during the current study are available in the RGB-D Dataset 7-Scenes, RGB Camera Relocalization 12-Scenes, and PoseNet project repositories.

Declarations

Conflict of interest The authors have no competing interests to declare that are relevant to the content of this article.

Appendix

We detail our experiments on Aachen dataset (Sattler et al., 2018). Following the earlier HSCNet work (Li et al., 2020) we modify the architecture for this experiment, and therefore present it separately in this appendix.

HSCNet++ Architecture for Aachen

For large-scale datasets such as Aachen Day-Night, the scene coordinate network is underperforming due to the challenge of extracting reliable features in the end-to-end training procedure. Thus, instead of training a feature extractor from scratch as in the HSCNet dense setting, we leverage the pre-trained SuperPoint network (DeTone et al., 2018) to extract more reliable image features as input. We modify our network to consider the SuperPoint features as input. Therefore, the dense set of local features is replaced by a sparse set of features. As a consequence, in the follow-up processing we are using convolutional layers with 1×1 convolutions. FiLM conditioning layers together with transformer modules are integrated in a similar way.

Due to the large scale of the scene, a retrieval process is used during inference to provide contextual evidence. Predictions are conditioned on the retrieved image id. During training, the image id of each training image is used as additional input, in the same spirit as the region labels. It is seen as the coarsest piece of localization information within the large scale scene; next coarsest is the discretized region labels. During inference, the image id of the retrieved image is provided as additional input. The retrieval method used is NetVLAD (Arandjelović et al., 2016) and the search is performed with the test image as query and the training images as database. We use multiple top retrieved images, perform the process for each of them, and maintain the predicted camera pose associated with the largest number of inliers. The detailed architecture of HSCNet++ for Aachen is shown in Fig. 7 and denoted by HSCNet++(A). This variant only relies on classification branches and no regression branch is used, which means that the final predictions are quantized 3D coordinates. There are four classification branches in total. K-means with a branching factor of 100 is used, which results in 685k valid

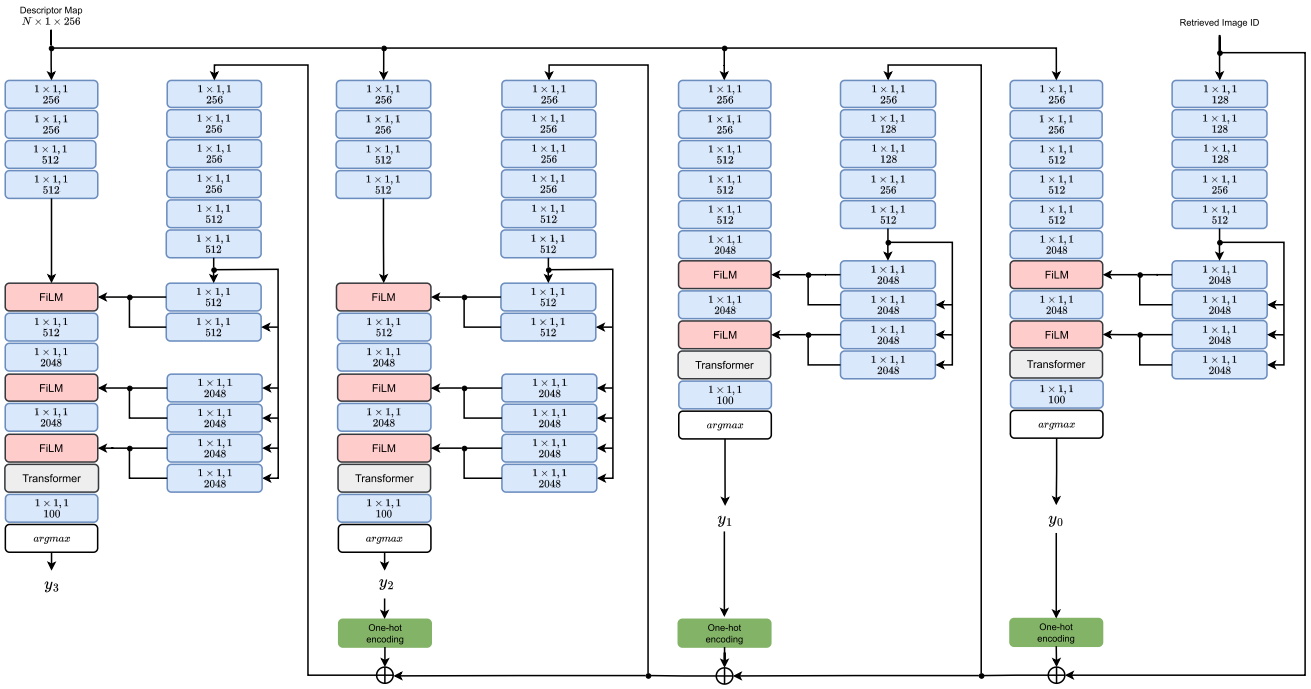


Fig. 7 An overview of the proposed HSCNet++(A) on Aachen. The figure shows the network architecture of the modified HSCNet++ for large-scale Aachen dataset. Here, y_0, y_1, y_2, y_3 are coarse-to-fine label predictions

Table 11 Accuracy on the Aachen dataset

Method	Aachen day 0.25m, 2° / 0.5m, 5° / 5 m, 10°	Aachen night 0.5m, 2° / 1 m, 5° / 5 m, 10°
AS (Sattler et al., 2016b)	57.3 / 83.7 / 96.6	28.6 / 37.8 / 51.0
HLoc (Sarlin et al., 2020)	**89.6 / 95.4 / 98.8**	**86.7 / 93.9 / 100.0**
PixLoc (Sarlin et al., 2021)	64.3 / 69.3 / 77.4	51.0 / 55.1 / 67.3
ESAC (50 experts) (Brachmann & Rother, 2019b)	42.6 / 59.6 / 75.5	6.1 / 10.2 / 18.4
HSCNet++(A)	72.7 / 81.6 / 91.4	43.9 / 57.1 / 76.5
HSCNet(A) top-10	71.1 / 81.9 / 91.7	40.8 / 56.1 / 76.5
HSCNet(A) top-1	64.0 / 76.1 / 85.4	28.6 / 38.8 / 59.2
HSCNet(A) top-1 (regression)	47.8 / 61.8 / 79.9	11.2 / 17.3 / 39.8
HSCNet(A) w/o retrieval	50.6 / 56.3 / 70.1	12.2 / 12.2 / 22.4

We report localization performance as a percentage (%) of correctly localized query images for 3 different thresholds. The best results are highlighted in bold

clusters at the finest level. Removing transformer modules from this architecture results in the HSCNet(A) architecture.

The network is trained for 900K iterations with a batch size of 1 and a learning rate of 10^{-4}. We use Adam (Kingma & Ba, 2014) optimizer and halve the learning rate every 50K iterations for the last 200K iterations. During training, only those Superpoint keypoints are kept that are triangulated in the sparse 3D model. At test time, top 2K Superpoint keypoints are kept per image based on keypoint scores after non-maximum suppression (NMS).

Results are presented in Table 11. Using more neighbors provides a good performance boost, while not conditioning on the image ids, therefore not using retrieval at all during inference, results in a large drop in performance. Changing the large branch into regression instead of classification compromises performance as well. The transformer modules noticeably boost the performance in this experiment as well. We compare with ESAC (Brachmann & Rother, 2019a), PixLoc (Sarlin et al., 2021) and local feature-based methods AS (Sattler et al., 2016b) and HLoc (Sarlin et al., 2020). The results indicate HSCNet++ surpasses end-to-end methods, ESAC and PixLoc across most thresholds. The proposed approach, alongside other end-to-end methods, falls short compared to local feature-based methods such as HLoc (Sarlin et al., 2020). The performance gap becomes more evident in night-time settings showing the limited robustness of end-to-end methods to illumination variations. However, HLoc's reliance on maintaining 3D maps can be quite challenging for large-scale environments, especially on mobile devices constrained by storage and communication bandwidth limitations. Therefore, the consideration of the memory-accuracy trade-off is imperative. While our model only requires 0.24GB, local feature methods like HLoc demand 7.8GB for their local descriptor database. Nevertheless, the accuracy of the proposed method is susceptible to a notable decline when faced with a substantial increase in scene scale for a fixed model size. This limitation could be addressed by deploying different models for distinct parts of a large scene by maintaining the memory-accuracy trade-off.

References

Arandjelović, R., Gronat, P., Torii, A., Pajdla, T. & Sivic, J. (2016). NetVLAD: CNN architecture for weakly supervised place recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 5297–5307).

Balntas, V., Li, S. & Prisacariu, V. (2018). RelocNet: Continuous metric learning relocalisation using neural nets. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 751–767). Springer International Publishing.

Balntas, V., Riba, E., Ponsa, D. & Mikolajczyk, K. (2016). Learning local feature descriptors with triplets and shallow convolutional neural networks. In *Proceedings of the British machine vision conference (BMVC)*

Bay, H., Tuytelaars, T. & Van Gool, L. (2006). SURF: Speeded up robust features. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 404–417). Springer International Publishing.

Brachmann, E., Humenberger, M., Rother, C. & Sattler, T. (2021). On the limits of pseudo ground truth in visual camera re-localisation. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)* (pp. 6218–6228).

Brachmann, E., Krull, A., Nowozin, S., Shotton, J., Michel, F., Gumhold, S. & Rother, C. (2017). DSAC - Differentiable RANSAC for camera localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 6684–6692).

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

Brachmann, E. & Rother, C. (2018). Learning less is more - 6D camera localization via 3D surface regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 4654–4662)

Brachmann, E. & Rother, C. (2019). Expert sample consensus applied to camera re-localization. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)* (pp. 7524–7533)

Brachmann, E. & Rother, C. (2019). Expert sample consensus applied to camera re-localization. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)* (pp. 7525–7534).

Brachmann, E. & Rother, C. (2019). Neural-guided RANSAC: Learning where to sample model hypotheses. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 4322–4331).

Brachmann, E., & Rother, C. (2021). Visual camera re-localization from RGB and RGB-D images using DSAC. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 44*(9), 5847–5865.

Brahmbhatt, S., Gu, J., Kim, K., Hays, J. & Kautz, J. (2018). Geometry-aware learning of maps for camera localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 2616–2625).

Budvytis, I., Teichmann, M., Vojir, T. & Cipolla, R. (2019). Large scale joint semantic re-localisation and scene understanding via globally unique instance coordinate regression. In *Proceedings of the British machine vision conference (BMVC)*

Bui, M., Albarqouni, S., Ilic, S. & Navab, N. (2018). Scene coordinate and correspondence learning for image-based localization. In *Proceedings of the British machine vision conference (BMVC)*

Calonder, M., Lepetit, V., Strecha, C. & Fua, P. (2010). BRIEF: Binary robust independent elementary features. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 778–792). Springer Berlin Heidelberg

Cavallari, T., Bertinetto, L., Mukhoti, J., Torr, P. & Golodetz, S. (2019). Let's take this online: Adapting scene coordinate regression network predictions for online RGB-D camera relocalisation. In: *International conference on 3D vision (3DV)* (pp. 564–573).

Cavallari, T., Golodetz, S., Lord, N., Valentin, J., Prisacariu, V., Di Stefano, L., & Torr, P. H. (2020). Real-time RGB-D camera pose estimation in novel scenes using a relocalisation cascade. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 42*(10), 2465–2477.

Cavallari, T., Golodetz, S., Lord, N.A., Valentin, J., Di Stefano, L. & Torr, P.H. (2017). On-the-fly adaptation of regression forests for online camera relocalisation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 4457–4466).

Chen, S., Li, X., Wang, Z. & Prisacariu, V. (2022). Dfnet: Enhance absolute pose regression with direct feature matching. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 1–17). Springer Nature Switzerland.

Chen, S., Wang, Z. & Prisacariu, V. (2021). Direct-posenet: Absolute pose regression with photometric consistency. In *International conference on 3D vision (3DV)* (pp. 1175–1185).

DeTone, D., Malisiewicz, T. & Rabinovich, A. (2018). Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 224–236).

Ding, M., Wang, Z., Sun, J., Shi, J. & Luo, P. (2019). CamNet: Coarse-to-fine retrieval for camera re-localization. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)* (pp. 2871–2880).

Dusmanu, M., Rocco, I., Pajdla, T., Pollefeys, M., Sivic, J., Torii, A. & Sattler, T. (2019). D2-Net: A trainable CNN for joint detection and description of local features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 8092–8101).

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM, 24*(6), 381–395.

Guan, P., Cao, Z., Yu, J., Zhou, C., & Tan, M. (2021). Scene coordinate regression network with global context-guided spatial feature transformation for visual relocalization. *IEEE Robotics and Automation Letters, 6*(3), 5737–5744.

Guzmán-Rivera, A., Kohli, P., Glocker, B., Shotton, J., Sharp, T., Fitzgibbon, A.W. & Izadi, S. (2014). Multi-output learning for camera relocalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 1114–1121).

Han, X., Leung, T., Jia, Y., Sukthankar, R. & Berg, A.C. (2015). Matchnet: Unifying feature and metric learning for patch-based matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 3279–3286).

Huang, Z., Zhou, H., Li, Y., Yang, B., Xu, Y., Zhou, X., Bao, H., Zhang, G. & Li, H. (2021). VS-Net: Voting with segmentation for visual localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 6101–6111).

Jiang, W., Trulls, E., Hosang, J., Tagliasacchi, A. & Yi, K.M. (2021). COTR: Correspondence transformer for matching across images. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)* (pp. 6207–6217).

Katharopoulos, A., Vyas, A., Pappas, N. & Fleuret, F. (2020). Transformers are rnns: Fast autoregressive transformers with linear attention. In *Proceedings of the 37th international conference on machine learning (ICML)* (pp. 5156–5165). JMLR

Kendall, A. & Cipolla, R. (2016). Modelling uncertainty in deep learning for camera relocalization. In *Proceedings of the IEEE international conference on robotics and automation (ICRA)* (pp. 4762–4769).

Kendall, A., Cipolla, R. (2017). Geometric loss functions for camera pose regression with deep learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 5974–5983).

Kendall, A., Gal, Y. & Cipolla, R. (2018). Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp 7482–7491).

Kendall, A., Grimes, M. & Cipolla, R. (2015). PoseNet: A convolutional network for real-time 6-DoF camera relocalization. In: *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)* (pp 2938–2946).

Kingma, D.P. & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv:1412.6980

Laskar, Z., Melekhov, I., Kalia, S. & Kannala, J. (2017). Camera relocalization by computing pairwise relative poses using convolutional neural network. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV) workshops* (pp. 929–938).

Li, X., Wang, S., Zhao, Y., Verbeek, J. & Kannala, J. (2020). Hierarchical scene coordinate classification and regression for visual localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 11,983–11,992).

Li, X., Ylioinas, J. & Kannala, J. (2018). Full-frame scene coordinate regression for image-based localization. In *Proceedings of robotics: science and systems (RSS)*

Li, X., Ylioinas, J., Verbeek, J. & Kannala, J. (2018). Scene coordinate regression with angle-based reprojection loss for camera relocalization. In *Proceedings of the European conference on computer vision (ECCV) workshops* (pp 229–245). Springer International Publishing.

Li, X., Ylioinas, J., Verbeek, J. & Kannala, J. (2018). Scene coordinate regression with angle-based reprojection loss for camera relocalization. In *Proceedings of the European conference on computer vision (ECCV) workshops* (pp. 0–0).

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision, 60*(2), 91–110.

Luo, Z., Shen, T., Zhou, L., Zhang, J., Yao, Y., Li, S., Fang, T. & Quan, L. (2019). Contextdesc: Local descriptor augmentation with cross-

modality context. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp 2527–2536).
Massiceti, D., Krull, A., Brachmann, E., Rother, C., & Torr, P.H. (2017). Random forests versus neural networks—What's best for camera localization? In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (pp. 5118–5125).
Melekhov, I., Brostow, G.J., Kannala, J. & Turmukhambetov, D. (2020). Image stylization for robust features. ArXiv preprint arXiv:2008.06959.
Melekhov, I., Kannala, J. & Rahtu, E. (2017). Image patch matching using convolutional descriptors with euclidean distance. In *Proceedings of the Asian conference on computer vision (ACCV) workshops* (pp. 638–653). Springer.
Melekhov, I., Laskar, Z., Li, X., Wang, S. & Juho, K. (2021). Digging into self-supervised learning of feature descriptors. In: *International conference on 3D vision (3DV)*(pp. 1144–1155).
Melekhov, I., Ylioinas, J., Kannala, J. & Rahtu, E. (2017). Image-based localization using hourglass networks. In: *Proceedings of the IEEE/CVF international conference on computer vision (ICCV) Workshops* (pp. 879–886).
Meng, L., Chen, J., Tung, F., Little, J.J., Valentin, J. & de Silva, C.W. (2017). Backtracking regression forests for accurate camera relocalization. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 6886–6893).
Meng, L., Tung, F., Little, J.J., Valentin, J. & de Silva, C.W. (2018). Exploiting points and lines in regression forests for RGB-D camera relocalization. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 6827–6834).
Mishchuk, A., Mishkin, D., Radenovic, F. & Matas, J. (2017). Working hard to know your neighbor's margins: Local descriptor learning loss. In *Advances in Neural Information Processing Systems (NIPS)*(vol. 30, pp. 4826–4837). Curran Associates, Inc.
Moreau, A., Piasco, N., Tsishkou, D., Stanciulescu, B. & de La Fortelle, A. (2021). LENS: Localization enhanced by neRF synthesis. In *Annual conference on robot learning*
Perez, E., Strub, F., De Vries, H., Dumoulin, V., & Courville, A. (2018). Film: Visual reasoning with a general conditioning layer. *Proceedings of the AAAI Conference on Artificial Intelligence, 32*(1), 3942–3951.
Radenović, F., Tolias, G.& Chum, O. (2016). CNN image retrieval learns from BoW: Unsupervised fine-tuning with hard examples. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 3–20). Springer International Publishing.
Radwan, N., Valada, A., & Burgard, W. (2018). VLocNet++: Deep multitask learning for semantic visual localization and odometry. *IEEE Robotics and Automation Letters, 3*(4), 4407–4414.
Revaud, J., De Souza, C., Humenberger, M. & Weinzaepfel, P. (2019). R2D2: Reliable and repeatable detector and descriptor. In: *Advances in neural information processing systems (NeurIPS)* (Vol. 32, pp. 12,405–12,415). Curran Associates, Inc.
Rogez, G., Weinzaepfel, P. & Schmid, C. (2017). LCR-Net: Localization-classification-regression for human pose. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 3433–3441).
Rogez, G., Weinzaepfel, P., & Schmid, C. (2019). LCR-Net++: Multi-person 2D and 3D pose detection in natural images. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 42*(5), 1146–1161.
Rublee, E., Rabaud, V., Konolige, K. & Bradski, G.R. (2011). ORB: An efficient alternative to SIFT or SURF. In: *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)* (pp. 2564–2571).
Saha, S., Varma, G. & Jawahar, C. (2018). Improved visual relocalization by discovering anchor points. In *Proceedings of the British machine vision conference (BMVC)*
Sarlin, P.E., Cadena, C., Siegwart, R. & Dymczyk, M. (2019). From coarse to fine: Robust hierarchical localization at large scale. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp 12,716–12,725).
Sarlin, P.E., DeTone, D., Malisiewicz, T. & Rabinovich, A. (2020). Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp 4938–4947).
Sarlin, P.E., Unagar, A., Larsson, M., Germain, H., Toft, C., Larsson, V., Pollefeys, M., Lepetit, V., Hammarstrand, L., Kahl, F., & Sattler, T. (2021). Back to the feature: Learning robust camera localization from pixels to pose. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 3247–3257).
Sattler, T., Leibe, B., & Kobbelt, L. (2011). Fast image-based localization using direct 2d-to-3d matching. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)* (pp. 667–674).
Sattler, T., Leibe, B., & Kobbelt, L. (2012). Improving image-based localization by active correspondence search. In *Proceedings of the European Conference on computer vision (ECCV)* (pp. 752–765). Springer International Publishing.
Sattler, T., Leibe, B., & Kobbelt, L. (2016). Efficient & effective prioritized matching for large-scale image-based localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*(9), 1744–1756.
Sattler, T., Leibe, B., & Kobbelt, L. (2016). Efficient & effective prioritized matching for large-scale image-based localization. *IEEE Transactions on Pattern Analysis And Machine Intelligence, 39*(9), 1744–1756.
Sattler, T., Maddern, W., Toft, C., Torii, A., Hammarstrand, L., Stenborg, E., Safari, D., Okutomi, M., Pollefeys, M., Sivic, J., Kahl, F., Pajdla, T. (2018). Benchmarking 6DoF outdoor visual localization in changing conditions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 8601–8610).
Sattler, T., Zhou, Q., Pollefeys, M. & Leal-Taixe, L. (2019). Understanding the limitations of CNN-based absolute camera pose regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 3302–3312).
Schönberger, J.L., Zheng, E., Pollefeys, M. & Frahm, J.M. (2016). Pixelwise view selection for unstructured multi-view stereo. In *Proceedings of the European conference on computer vision (ECCV)*
Shavit, Y., Ferens, R. & Keller, Y. (2021). Learning multi-scene absolute pose regression with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)* (pp 2733–2742).
Shavit, Y. & Keller, Y. (2022). Camera pose auto-encoders for improving pose regression. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 140–157). Springer International Publishing
Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., & Fitzgibbon, A. (2013). Scene coordinate regression forests for camera relocalization in RGB-D images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 2930–2937).
Simo-Serra, E., Trulls, E., Ferraz, L., Kokkinos, I., Fua, P., & Moreno-Noguer, F. (2015). Discriminative learning of deep convolutional feature point descriptors. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)* (pp. 118–126).
Sun, J., Shen, Z., Wang, Y., Bao, H. & Xiaowei, Z. (2021). LoFTR: Detector-free local feature matching with transformers. In *Pro-*

ceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) (pp. 8922–8931).

Taira, H., Okutomi, M., Sattler, T., Cimpoi, M., Pollefeys, M., Sivic, J., Pajdla, T., & Torii, A. (2018). Inloc: Indoor visual localization with dense matching and view synthesis. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 7199–7209).

Tian, Y., Fan, B. & Wu, F. (2017). L2-net: Deep learning of discriminative patch descriptor in euclidean space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 661–669).

Tyszkiewicz, M., Fua, P., & Trulls, E. (2020). DISK: Learning local features with policy. In *Advances in neural information processing systems (NeurIPS)* (Vol. 33, pp. 14,254–14,265). Curran Associates, Inc.

Valada, A., Radwan, N. & Burgard, W. (2018). Deep auxiliary learning for visual localization and odometry. In *Proceedings of the IEEE international conference on robotics and automation (ICRA)* (pp. 6939–6946).

Valentin, J., Dai, A., Nießner, M., Kohli, P., Torr, P., Izadi, S., & Keskin, C. (2016). Learning to navigate the energy landscape. In *International conference on 3D vision (3DV)* (pp. 323–332).

Valentin, J., Nießner, M., Shotton, J., Fitzgibbon, A., Izadi, S., & Torr, P.H. (2015). Exploiting uncertainty in regression forests for accurate camera relocalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 4400–4408).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems (NeurIPS)* (Vol. 30, pp. 5998–6008). Curran Associates, Inc.

Walch, F., Hazirbas, C., Leal-Taixe, L., Sattler, T., Hilsenbeck, S., & Cremers, D. (2017). Image-based localization using LSTMs for structured feature correlation. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)* (pp. 627–637).

Wang, Q., Zhou, X., Hariharan, B., & Snavely, N. (2020). Learning feature descriptors using camera pose supervision. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 757–774). Springer International Publishing

Wang, S., Laskar, Z., Melekhov, I., Li, X., & Kannala, J. (2021). Continual learning for image-based camera localization. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)* (pp. 3252–3262).

Wang, Y., Ma, X., Chen, Z., Luo, Y., Yi, J., & Bailey, J. (2019). Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)* (pp. 322–330).

Weinzaepfel, P., Csurka, G., Cabon, Y., & Humenberger, M. (2019). Visual localization by learning objects-of-interest dense match regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 5634–5643).

Xue, F., Wang, X., Yan, Z., Wang, Q., Wang, J., & Zha, H. (2019). Local supports global: Deep camera relocalization with sequence enhancement. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)* (pp. 2841–2850).

Xue, F., Wu, X., Cai, S., & Wang, J. (2020). Learning multi-view camera relocalization with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 11,375–11,384).

Zagoruyko, S., & Komodakis, N. (2015). Learning to compare image patches via convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 4353–4361).

Zhou, Q., Sattler, T., & Leal-Taixé, L. (2021). Patch2Pix: Epipolar-guided pixel-level correspondences. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 4669–4678).