MAD-DR: Map Compression for Visual Localization with Matchness Aware Descriptor Dimension Reduction

Qiang Wang

EasyAR Mega, wq@sightp.com

Abstract. 3D-structure based methods remain the top-performing solution for long-term visual localization tasks. However, the dimension of existing local descriptors is usually high and the map takes huge storage space, especially for large-scale scenes. We propose an asymmetric framework which learns to reduce the dimension of local descriptors and match them jointly. We can compress existing local descriptor to 1/256 of original size while maintaining high matching performance. Experiments on public visual localization datasets show that our pipeline obtains better results than existing map compression methods and non-structure based alternatives.

1 Introduction

Given one image, visual localization or image-based localization aims to recover the position and orientation of the camera relative to known scene [56, 63, 67]. The task is of great importance for applications such as augmented reality [37, 54], robotics and autonomous driving, for which centimeter-level localization accuracy is usually desired.

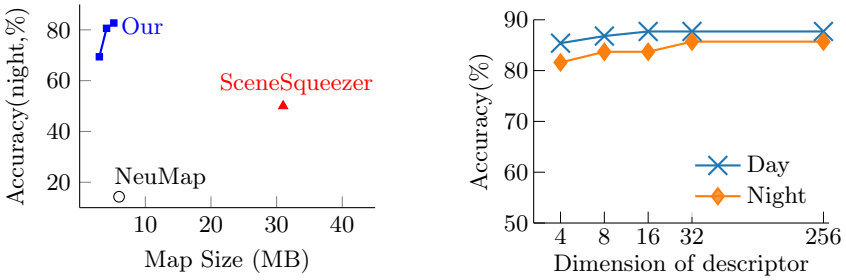

Fig. 1: Localization performance on Aachen Day-Night dataset with proposed pipeline. We beat previous best-performing map compression method SceneSqueezer [72] and non-structure based NeuMap [64] by significant margin (left). We reduce the dimension of SuperPoint [21] descriptor from 256 to 4/8/16/32, while maintaining $> 80\%$ accuracy for both day and night queries, measured under (0.25m, 2°) [56] (right).

State-of-the art methods for visual localization are mostly hierarchical and structure-based [54, 56, 67]. During the offline mapping stage, the poses for database images are obtained via structure from motion (SFM) [59] or simultaneous localization and mapping (SLAM) systems [42, 45]. Local features are matched, triangulated and bundle adjustment-ed to obtain 3D structure of the scene. The appearance information (global and local features) as well as geometric information (*e.g.* 2D/3D position) for the local features are saved offline. During the localization stage, the local features extracted from query image are matched to the top-N most similar database images to obtain 2D-3D correspondences. The match results as well as 3D positions are fed to perspective-n-point (PnP) and RANSAC algorithms to obtain the 6-DOF pose of the query image.

One of the core problems with structure-based localization methods is large amount of storage requirement. It is mainly caused by the need to store thousands of local descriptors per database image, the dimension of which is usually high, *e.g.* 128 for SIFT [35] and 256 for SuperPoint [21]). Many methods have been proposed to reduce the map size, either by selecting a subset of local features [16, 17, 39, 72, 73] or local descriptor compression [15, 22]. The issue with existing map compression methods is that they are designed without jointly tuning the matching model. They rely on heuristic nearest neighbor matching [15,19] or use pre-trained matching models [33, 53] without retraining the weights [72]. We observe that transformer-based image matching models are powerful enough to perform cross-time, cross-season even cross-modality matching tasks. This motivates us to push the limit of local descriptor compression by jointly training image matching pipeline to handle such heavily compressed features. It enables high accuracy visual localization method with extreme compact map, as shown in Figure 1. The contributions of this paper can be summarized as below.

1. We show the importance of jointly training local descriptor dimension reduction and image matching model for scene compression of structure-based visual localization methods. We compress SuperPoint descriptor to 1/256 of original size with minor impact on matching performance.
2. With the compressed descriptor, we provide a simple yet strong baseline which obtains $> 80\%$ accuracy under (0.25m,2°) with map size of 4.1MB for Aachen Day-Night dataset.
3. Detailed experiments and ablation studies show our method offers better accuracy/memory trade-off and generalization ability than existing map compression methods and non-structure based alternatives.

2 Related Work

2.1 Visual localization

Visual localization which aims to recover the 6-DOF pose of query image with regard to given scene has been well studied over the past decade. Traditional methods are structure-based which rely on explicit matching hand-crafted local features such as SIFT [55] and binary features [37]. Hierarchical structure-based

methods [26,52,57] have become the de-facto standards for long-term localization tasks known for high scalability and exceptional accuracy. They first obtain most similar database images via image retrieval with global features [4, 9, 10, 46, 48]. Local feature matching [21, 23, 35, 49, 53, 68] is then performed between query image and retrieved database images to obtain 2D-3D correspondences for pose estimation.

Since the advent of deep learning era, various end-to-end methods have been proposed to tackle the visual localization task. Some methods formulate the task as absolute pose regression [28] or relative pose regression [6] problem without 2D-3D matching. Other methods [11, 12, 64] train scene coordinate regression models which directly predict 3D coordinates for pixels in query image. End-to-end methods claimed to require less storage [13, 71] and have shown promising results for small-size to medium-size scenes [28]. But the scalability to large scenes and accuracy under challenging cases still fall behind of structure-based methods on several benchmarks [54, 56, 58, 63, 67].

2.2 Local feature matching

Detector based methods such as SIFT [35] have been long standing golden standard for local feature detection and matching. Many follow-up works [3,8,30,50] focus on improving the speed of matching by using binary features instead. Being fast and compact, they are ideal for resource limited platforms like mobile devices and robotics. But the matching performance of binary feature is worse than SIFT, which exhibits strong results for well-illuminated, texture-rich image pairs [56]. Many methods have been proposed to learn local features to replace hand-crafted ones, usually with convolution neural network. Some works [21, 23, 68] learn local detector and descriptors jointly while others [25,36,65,66] focus on descriptors only. Trained with extensive data augmentation, learnt features show better matching performance for challenging cases with strong lighting changes [67] and few view overlap.

On the other hand, detector-free methods [18, 62] have been proposed which show impressive matching results for low-texture scenes where detector-based methods struggle with low keypoint repeatability. But they are usually slower, take more storage and show no clear advantage for visual localization task [33].

Context aggregation methods [53, 62] which employ transformer architecture [69] turn out to boost image matching performance greatly. They perform well for challenging scenarios such as day-night matching and cross-seasonal matching where traditional methods usually fail. The pioneering SuperGlue [53] model resembles the transformer [69] architecture and re-formulate it within graph neural network framework for end-to-end training. Thanks to the representation power of attention-based reasoning, such formalization has been proven to be effective for several image matching applications including homography estimation, relative pose estimation and 3D reconstruction [51].

Sarlin *et al*. [52] integrated such method within hierarchical pipeline for visual localization task. Their system named *hloc* delivered strong results across several public benchmarks. However, it takes huge storage due to the need of storing

thousands of high dimension local descriptors per database image [15, 71, 72]. For applications such as city-scale visual positioning systems, the size of 3D map especially local descriptor needs to be compressed heavily [24, 38].

2.3 Map compression

Many works have been proposed to reduce the map size of structure-based methods either by map sparsification [39, 73], descriptor compression [19] or hybrid approach [15, 72]. Map sparsification [16, 17, 31] methods select the most salient features while removing least informative ones. Descriptor compression methods reduce the size of each local descriptor while trying to maintain matching performance. Cheng *et al*. [19] represented the SIFT descriptors of each 3D point as an integer mean descriptor per visual word, which is further converted into binary signature using hamming embedding. Lynen *et al*. [37] performed binary features projection and product quantization for real time localization. Hybrid methods [15] kept a small set of points with full information and larger set of points with compressed information. Their motivation is that keeping more points in the map is important for accurate pose estimation.

Recently, Yang *et al*. [72] employed multi-stage pipeline to compress the scene for learning-based feature and matching pipeline. They perform frame selection, point selection and feature quantization to compress local descriptors [21]. The compressed descriptors are passed into de-quantization network which is used as input to pre-trained SuperGlue [53] model for matching. By using multi-layer perceptron (MLP) and differentiable soft quantization, they compress SuperPoint descriptor to 2048-bit (compression ratio 1/8). Dong *et al*. [22] proposed a siamese training method which uses MLP to perform descriptor dimensionality reduction. The authors projected several features [7, 35, 40, 41] to dimension of 64/34/24/16 while minimizing descriptor reconstruction loss and patch similarity loss. The matching performance dropped significantly for night scene when matched at low dimension(< 64) [22]. We infer partial of the reason for such degradation is L2-distance ambiguity for low-dimensional descriptor. Some works [70, 75] went to the extreme by storing no visual descriptors but only 3D coordinates in the map. Though the idea is interesting, they only report results on relatively less challenging datasets [28, 60] which makes their performance under difficult scenarios questionable.

The key motivation for this paper is that end-to-end attention based context aggregation matching pipeline [53] greatly boosted performance of existing local feature [21, 35] that previously use nearest neighbor criteria or ratio test [21, 35, 68], especially for challenging scenes [56, 63]. We show such pipeline can perform well when matching local descriptors reconstructed from low-dimension embedding if properly trained [51]. Though nearest neighbor or ratio test starts to fail [22] when decreasing descriptor embedding dimension, aggregating context utilizing attention [69] and positional encoding will make up the performance gap when jointly trained with such data. As shown in the experiment section, such joint learning is important since directly applying pre-

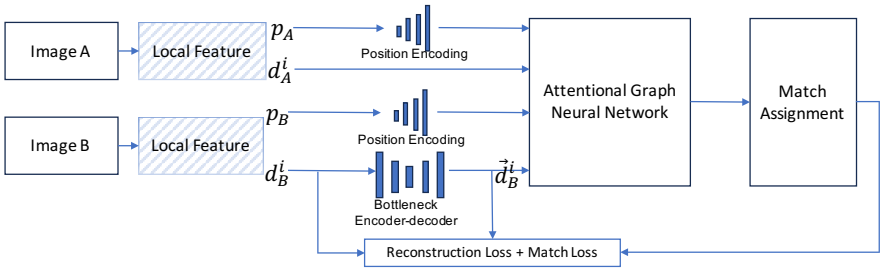

Fig. 2: Framework of our pipeline that jointly trains bottleneck encoder-decoder module for descriptor compression and image matching. Given a pair of images (A,B), the descriptors from image B pass through encoder-decoder to obtain reconstructed ones, which are trained to match with correspondences from image A.

trained weights [72] with reconstructed descriptors will yield sub-optimal results, assuming due to domain gap introduced by descriptor compression.

3 Matching Aware Descriptor Dimension Reduction

The objective of this paper is to reduce the storage of 3D scene representation for structure-based visual localization pipelines [26, 52]. Each database image in the map is associated with one global feature (with dimension D_G) as well as locations and descriptors for local features (number of local features as N_L, the dimension of the descriptor as D_L). Since $D_L \times N_L \gg D_G$, local descriptors bring significant storage issue for large scene. We solve this problem by reducing D_L to much smaller embedding D_{LR} (for example SuperPoint [21] with $D_L = 256$, $D_{LR} = 4$) while keeping high localization performance. We propose to train the encoder-decoder based dimension reduction module within attention based image matching pipeline [53].

In this section, we first briefly review attention-based context aggregation pipeline for image matching. Then we detail the asymmetric framework of jointly learning descriptor compression and image matching. The design of encoder-decoder module and training loss are introduced. Finally, the full pipeline for visual localization with proposed method is shown.

3.1 Preliminaries: attention based context aggregation pipeline for image matching

We follow the design of SuperGlue [53] framework for matching original and compressed descriptors. Given a pair of images A, B to match, local feature detector generates a set of salient features with descriptors and 2D positions (d_I^i, p_I), in which $I \in \{A, B\}$, $p_I \in \mathbb{R}^2$, $d_I^i \in \mathbb{R}^{D_L}$.

The normalized 2D positions p_I are fed into position encoding module to obtain high dimension representation to inject keypoint position information into

the network. The positional embedding is summed with the original descriptor to obtain initial representation

$$f_I^0 = \mathrm{PE}(p_I) + d_I^i, I \in \{A, B\} \tag{1}$$

in which $\mathrm{PE}(p_I)$ denotes position encoding module, f_I^0 acts as the input to multi-layer message-passing graph neural network to obtain updated representations. In each layer of GNN, multi-layer perceptron updates the representation of each image with message $m_{\epsilon \to i}$ aggregated either from itself or from other image to be matched

$$f_i^{n+1} = f_i^n + \mathrm{MLP}\left[f_i^n || m_{\epsilon \to i}\right] \tag{2}$$

where $|.|$ denotes concatenation. The message passing process is carried out via alternating computing self-attention and cross-attention layers. The message is computed by softmax attention as in the transformer [69] framework

$$m_{\epsilon \to i} = \sum_{j:(i,j) \in \epsilon} \mathrm{Softmax}(q_i^T k_j) v_j \tag{3}$$

where q_i, k_j, v_j are projected query, key and value. The attentional GNN module aggregates the neighboring geometric information as well as local patch appearance information for holistic matching [53].

Finally, the matching scores between the images are computed based on the pairwise similarity of enhanced representations.

$$S_{ij} = \langle f_i^A, f_j^B \rangle, \forall i, j \in (A, B) \tag{4}$$

where $\langle , \rangle$ denotes inner product.

The problem can be viewed as optimal transport problem between distributions, which was initially solved by Sinkhorn algorithm [20,53] and later by more efficient dual-softmax assignment [33, 44, 62]. The final matches can be obtained by retaining most confident matches from the assignment matrix.

3.2 Joint training of descriptor compression and image matching

The framework of joint training descriptor dimension compression and image matching pipeline is shown in Figure 2. We adopt an asymmetric design to embed descriptor dimension reduction module into image matching pipeline. To be specific, after running local feature detector on both images to obtain 2D location and descriptor $(d_I^i, p_I), I \in A, B$, we keep descriptors corresponding to image A unchanged (d_A^i), while passing the descriptors corresponding to the other image (d_B^i) into encoder-decoder dimension reduction module to obtain reconstructed descriptors

$$\overline{d}_B^i = \mathrm{Decoder}(\mathrm{Encoder}(d_B^i)) \tag{5}$$

The position encoding module takes uncompressed 2D feature position and project them to positional embedding $\mathrm{PE}(p_A)$,$\mathrm{PE}(p_B)$. We feed $(d_A^i, \mathrm{PE}(p_A))$ and $(\overline{d}_B^i, \mathrm{PE}(p_A))$ to attentional GNN module for further matching.

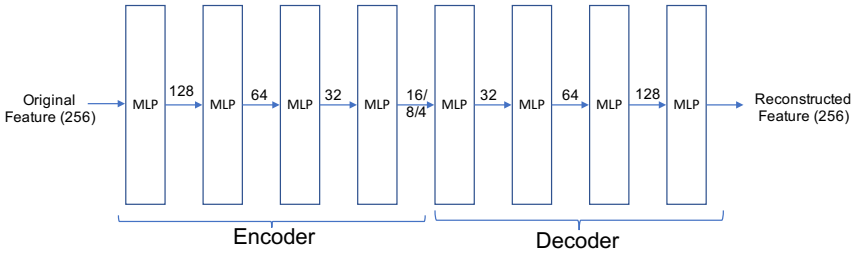

Fig. 3: Structure of our encoder-decoder module. Given original local feature (take SuperPoint as example), we gradually reduce the dimension to L_{DR} then reconstruct the descriptor with MLP layers. ReLU and batch normalization are omitted.

The choice of asymmetric design is deliberate for several reasons. On one hand, the branch which retains the original descriptors d_A^i serves as anchor for matching. Performing symmetric dimension reduction and reconstruction for both images may lead to descriptor space co-shift using extreme low-dimension compression. On the other hand, for the task of visual localization, we can keep local descriptors for query image as distinctive as possible by performing no compression since it can be computed on-the-fly. As shown in the experiments section, such asymmetric design does achieve better performance than symmetric counterpart.

We choose to re-project the descriptor to original dimension to act as input of attentional graph neural network. We did not perform matching with lower dimension to share the same network capacity with the baseline. The encoder-decoder module is lightweight compared to attention GNN module. It brings negligible computation overhead to both model training and inference. Instead of storing original descriptor with dimension D_L in the map, we save the encoded embedding with reduced dimension D_{LR}.

3.3 Encoder-decoder for descriptor compression

In the encoder-decoder module, we first use encoder composed by a stack of MLP layers to gradually decrease descriptor dimension to the lowest embedding dimension (*e.g.* 4/8/16) similar to Dong *et al.* [22]. Several MLP layers are then progressively applied to increase the dimension back to original dimension, denoted as decoder as shown in Figure 3. Additional normalization is applied to make reconstructed descriptors to be of unit length. All MLP layers are followed by ReLU [43] and batch normalization [27].

$$d_{i+1} = \mathrm{BN}(\mathrm{ReLU}(\mathrm{MLP}(d_i))) \tag{6}$$

3.4 Training losses

We minimize the negative log likelihood (NLL) loss following the SuperGlue framework [53]

$$L_{match} = -\sum_{(i,j)\in\mathcal{M}} log\ P_{i,j} - \sum_{i\in\mathcal{I}} log\ P_{i,N+1} - \sum_{j\in\mathcal{J}} log\ P_{M+1,j} \quad (7)$$

where $\mathcal{M}$ denotes ground truth matches, $\mathcal{IJ}$ denote unmatched keypoints.

We additional enforce encoder-decoder module to reconstruct original descriptor as closely as possible. Local descriptor reconstruction loss is added, which is defined as average reconstruction error of all descriptors within the image

$$L_{reconstruct} = \frac{1}{N_L}\sum_{k=1}^{N_L} ||\bar{d}_B^k - d_B^k||^2 \quad (8)$$

where N_L denotes number of local features in this image.

The total loss is defined as weighted sum of matching loss and descriptor reconstruction loss.

$$L_{total} = L_{match} + \lambda L_{reconstruct} \quad (9)$$

where λ is tunable parameter to balance the two loss terms. The role of reconstruction loss is discussed in the experiments section.

3.5 Full pipeline for map compression

We run the MLP layers corresponding to encoder to independently project each local descriptor to lowest embedding dimension. We further utilize post-training quantization of local descriptor from full-precision (FP32) to 4-bit. The compressed descriptor size is only 16-bit($D_{LR=4}$) or 32-bit($D_{LR=8}$) .

Instead of performing co-visibility based keypoint selection [16,72], we retain top-K (*e.g.* 256,512) keypoints with highest score for each image. This makes our method work for non-SfM data (InLoc [63]/Niantic Map-free dataset [5]) while previous compression methods [72] hardly can. We optionally filter SfM results by removing 3D points with max_reproj_error>2, min_tri_angle<1 [59].

Compared to moderate feature quantization (2048-bit) in previous works [72], we show that extreme compression of the descriptors (16/32/64-bit) is a better choice. Our method generalizes well for SuperPoint, DISK [68], ALIKED [74] and SIFT as shown in experiments and supplementary.

4 Experiments

Training details. We train our joint descriptor dimension reduction and image matching pipeline on MegaDepth dataset [32]. We use single model for evaluation across all datasets, both indoor [63] and outdoor [28, 56] ones. The training settings for experiments with SuperPoint are shown below. For other features, we use glue-factory [1] and follow default settings.

We extracted at most 1024 features and pad with random points for batching purpose. Our model is trained on single machine with 8 GPU card using PyTorch framework. We use Adam [29] optimizer with initial learning rate of 5.0e-05. We

set the learning rate schedule to exponential decay of 0.999999. We use batch size of 8 per GPU, making total batch size of 64. We find larger batch size slightly increases training performance thus we perform gradient accumulation [33, 34]. We train for 400K steps since we find additional training steps bring little benefit. We set λ of the training loss to be 1.0 unless specified.

Localization datasets. Localization experiments are performed on various public datasets including the Cambridge Landmark dataset [28], Aachen Day-Night [56] and InLoc dataset [63]. Cambridge dataset is a medium-scale outdoor dataset recorded with video sequences. Aachen Day-Night dataset is a large-scale, outdoor dataset with 4,479 database images. It's captured with consumer cameras and reconstructed with structure from motion methods [59]. It is intended to benchmark visual localization methods across different times across the whole day. InLoc dataset focuses on indoor scenarios which is challenging with poor texture and high similarity.

Localization experiments details. We follow the *hloc* [52] framework to validate the impact of dimension reduction on visual localization tasks. For Cambridge Landmark dataset and Aachen Day-Night dataset, we triangulate the 3D maps using given poses. We use officially released feature (SuperPoint/ ALIKED/ DISK) and SuperGlue weights during the mapping process. We compress the local descriptors of resulting map with our trained encoder and store the low dimension embedding descriptors after quantization.

During the localization stage, we first use global feature to retrieve top-N database images, where N is fixed across all settings. The compressed local descriptors are retrieved and passed though the decoder to obtain reconstructed descriptors. The descriptors from the query image and the reconstructed descriptors are fed into our jointly-trained matcher to obtain 2D-2D matches. The final pose is obtained by PnP and RANSAC with all 2D-3D correspondences.

We use EigenPlace [10] with dimension 256 (D_G=256) as global feature, which offers similar accuracy as NetVLAD [4] on Cambridge and Aachen dataset while being much more compact. We quantify EigenPlace features similar to SceneSqueezer [72] to 8-bit. Different from original *hloc*, we extract fewer keypoints for database image but more for query images. We find such simple setting minimizes the map size while achieving similar localization performance. We optionally filter keypoint observation with large re-projection error [59]. We report results of our pipeline with D_{LR}=4 and quantized to 4-bit unless specified, reducing the descriptor size to 1/256 of original feature (D=256,FP16).

4.1 Quantitative comparison

Evaluation Metrics. For Cambridge Landmark dataset, we report median translation error (in meters) and orientation error (in degrees) on four scenes, following previous work [28, 72, 75]. For Aachen and InLoc dataset with hidden ground truth, we submit our results to the visual localization benchmark server [56] and report the percentage of successfully localized images under three different thresholds: (0.25m, 2°), (0.5m, 5°) and (5m, 10°).

Methods	Shop Facade			Old Hospital			King's College			Church		
	Size (MB)	Tran.Err (m)	Rot.Err (°)	Size (MB)	Tran.Err (m)	Rot.Err (°)	Size (MB)	Tran.Err (m)	Rot.Err (°)	Size (MB)	Tran.Err (m)	Rot.Err (°)
SP+SG [52]	373	0.04	0.20	1436	0.15	0.31	1687	0.11	0.21	1957	0.07	0.22
AS [55]	38.7	0.04	0.21	140	0.20	0.36	275	0.13	0.22	359	0.08	0.25
PoseNet [28]	50	1.46	8.08	50	2.31	5.38	50	1.92	5.40	50	2.65	8.48
DSAC++ [11]	207	0.06	0.30	207	0.20	**0.30**	207	0.18	0.30	207	0.13	0.40
NeuMap [64]	0.3	0.06	**0.25**	**0.2**	0.19	0.36	**0.3**	0.17	0.53	**0.4**	0.17	0.53
QP+RootSIFT [39]	0.41	0.72	1.4	1.1	0.9	2.17	2.2	1.53	1.09	3.3	0.56	0.89
hybrid [15]	0.16	0.19	0.54	0.62	0.75	1.01	1.01	0.81	0.59	1.34	0.5	0.49
KC [31]	0.85	0.51	0.87	6	1.35	1.06	3.1	1.48	1.23	18	0.46	0.69
KCP [16]	1.30	0.44	0.8	8.2	1.19	1	5.9	0.99	0.86	24	0.4	0.61
SceneSqueezer [72]	0.13	0.11	0.38	0.53	0.37	0.53	**0.3**	0.27	0.38	0.95	0.15	0.37
BPnPNet [14]	-	7.53	107	-	24.8	163	-	26.7	107	-	11.1	49.7
GoMatch [75]	-	0.48	4.77	-	2.83	8.14	-	0.25	0.64	-	3.35	9.94
DGC-GNN [70]	-	0.15	1.57	-	0.75	2.83	-	0.18	0.47	-	1.06	4.03
Our SP+SG(256)	0.12	**0.05**	**0.25**	0.40	**0.17**	0.31	0.43	**0.15**	**0.24**	0.94	**0.08**	**0.23**
Our SP+SG(128)	**0.07**	0.06	0.29	0.25	0.26	0.45	0.42	0.17	0.25	0.64	0.10	0.30

Table 1: Comparison with existing methods on Cambridge Landmark dataset [28]. We extract maximum 256 or 128 features per database image. We obtain higher accuracy with compact map compared to existing methods.

Cambridge Landmark datasets. The results on Cambridge Landmark datasets are shown in Table 1. We compare with structure-based methods, end-to-end methods, previous map compression methods and descriptor free methods.

Structure-based methods (SuperPoint+SuperGlue,SP+SG [52,53] and active search,AS [55]) obtain the highest accuracy while the accuracy of descriptor-free methods (BPnPnet [14], GoMatch [75], DGC-GNN [70]) is not as good. Our pipeline introduces minor performance drop compared to original SP+SG baseline while reducing the map size by $> 99\%$. We obtain higher accuracy (half translation error) compared with previous S.O.T.A compression method SceneSqueezer [72] using smaller map size on most scenes. They use a complex multi-stage pipeline (co-visible frames clustering and pruning, differentiable point selection, feature quantization) while our method is simpler and easier to implement. The only exception is King's College Scene, we obtain higher accuracy with slightly larger map (0.4MB *vs*. 0.3MB). Among end-to-end methods, NeuMap [64] performs better than PoseNet [28], DSAC++ [11] with similar map size ($<$ 1MB) and accuracy compared to structure-based methods. Considering the map size and accuracy of leading methods on Cambridge dataset are almost saturated, we suggest all future work should move on to more challenging datasets.

Aachen Day-Night datasets. We show the results of our methods on Aachen Day-Night dataset in Table 2. We extract 512/256 keypoints for each database image while for query image we extract at most 2048 keypoints. Our method using embedding dimension of 4 performs on a par with original *hloc* pipeline, indicating existing local descriptors with dimension=128 or 256 are highly redundant when matched with context aggregation methods. For example, our

	Methods	Size (MB)	Aachen Day 0.25,2/0.5,5/5,10	Aachen Night 0.25,2/0.5,5/5,10
FM	SP+SG [53]	6336	88.2/ 94.8/ 97.9	85.7/ 92.9/ 99.0
	SceneSqueezer [72]	31	75.5/ 89.7/ 96.2	50.0/ 67.3/ 78.6
	Cascaded [19]	140	76.7/ 88.6/ 95.8	33.7/ 48.0/ 62.2
	QP+R.SIFT [39]	31	62.6/ 76.3/ 84.7	16.3/ 18.4/ 24.5
	DISK+MNN(PQ, M=64, b=4)	21.8	84.1/ 93.8 /97.3	80.6/ 89.8/ 96.9
	DISK+MNN(PQ, M=32, b=4)	13.9	80.0/ 91.0 /96.1	70.4/ 81.6/ 91.8
E2E	NeuMap [64](10m,10)	170	76.2/ 88.5/ 95.5	37.8/ 62.2/ 87.8
	ESAC [12]	1315	42.6/ 59.6/ 75.5	6.1/ 10.2/ 18.4
Proposed	Our SP+SG(512)	5.2	84.6/ 92.4/ 96.7	82.7/ 90.8/ 98.0
	Our SP+SG(256)	3.0	80.0/ 89.2/ 95.4	69.4/ 81.6/ 92.9
	Our SP+SG(512,filtering)	4.1	84.5/ 92.6/ 97.0	80.6/ 90.8/ 98.0
	Our SP+SG(256,filtering)	2.4	79.0/ 88.0/ 94.8	65.3/ 80.6/ 92.9

Table 2: Localization results and map size on Aachen Day-Night dataset. We extract at most 512/256 keypoints per database image and optionally filter 3D points. Our method offers better accuracy with compact size than previous map compression methods and recent end-to-end methods.

pipeline obtains 84.5%/80.6% localization rate under (0.25 meter, 2°) for day and night queries with map size only 4.1MB, beating all previous works.

SceneSqueezer [72] uses SuperPoint and SuperGlue similar to us, but they use official weights without re-training. Our performance drop is much smaller than their pipeline. With 13.2%(4.1 *vs*. 31) map size, we localize 80.6% night queries compared to them with 50% under (0.25m,2°). We even obtain higher accuracy (65.3%) than them with only 7.7% memory needed(2.4MB*vs*. 31MB).

Feature matching methods with SIFT features (Cascaded [19] and QP+R.SIFT [39]) are out-performed by learnt features with 33.7% and 16.3% localized. Our pipeline also achieves better results than end-to-end methods (NeuMap [64] and ESAC [11]) which localize 48.0% and 6.1% queries each.

Surprisingly, a carefully tuned baseline method which matches DISK feature via mutual nearest neighboring and compresses with product quantization beats previous state-of-the-art SceneSqueezer [72] in both memory and accuracy. That shows the necessity of proper baseline methods in the community. Our method beats such DISK+MNN+PQ baseline with only 18.8%(4.1/21.8) memory size at similar accuracy. Refer to the supplementary material for detailed discussion.

InLoc datasets. We show the results of our methods on InLoc [63] dataset in Table 3. Our method with dimension 4 localizes 48.0%/45.8% queries under threshold (0.25m,2°), only minor drop (1.5%,5.3%) compared to uncompressed baseline. Dong *et al*. [22] also applies MLP to SIFT [35] and HardNet [40] descriptors for dimension reduction. Their performance drops drastically when using descriptor dimension < 64. We attribute the high performance of our pipeline

Methods	Dim	InLoc duc1	InLoc duc2
SP+SG (No compression)	256	49.5/69.7/81.3	51.1/75.6/82.4
DISK+NN (No compression)	128	34.3/56.6/67.7	30.5/40.5/57.3
SIFT+SV+NN [22]	64	33.3/46.0/57.6	26.0/39.7/45.8
SIFT+SV+NN [22]	16	24.2/36.9/43.9	15.3/26.7/30.5
HardNet+SS+NN [22]	64	40.9/56.6/71.2	32.1/45.8/51.9
HardNet+SS+NN [22]	16	22.7/32.8/41.9	12.2/19.8/22.9
Our SP+SG	4	48.0/67.7/79.3	45.8/69.5/76.3

Table 3: Localization results of our methods on the InLoc dataset. Our method outperforms Dong *et al*. [22] (SS:self-supervised, SV:supervised) by a large margin.

to both advanced image matching pipeline and joint training process as shown in the ablation studies. Note the DISK+MNN baseline which works well for Aachen Day-Night dataset performs not good (>15% worse than our under 0.25m,2°) on InLoc dataset, which indicates DISK may be overfitted on outdoor/building scenes. Previous methods [39, 72] seldom report results on InLoc dataset. Considering the performance gap of DISK+MNN on Aachen Day-Night and InLoc datasets, results on various datasets are important for fair comparison of different methods.

Methods	Aachen Night	InLoc duc1	InLoc duc2
	0.25,2/0.5,5/5,10	0.25,2/0.5,5/5,10	0.25,2/0.5,5/5,10
Our full	82.7/89.8/**99.0**	**46.5**/**66.7**/**79.3**	**48.1**/**68.7**/76.3
No re-training	71.4/82.7/92.9	42.9/59.1/71.7	45.0/65.6/74.0
No recons loss	**83.7**/89.8/**99.0**	40.9/56.1/69.2	42.0/62.6/**77.9**
Asym query+db	**83.7**/**90.8**/**99.0**	44.9/62.6/75.3	39.7/61.1/68.7
Sym db only	72.4/83.7/93.9	30.8/46.0/57.6	29.0/49.6/56.5
Sym query+db	81.6/89.8/**99.0**	37.4/56.6/67.7	32.8/55.7/64.9

Table 4: Using pre-trained SuperGlue weights [53] without re-training obtains poor results for Aachen night queries. Training without reconstruction loss yield slightly worse result on InLoc dataset. Symmetric trained pipeline or applying asymmetrically trained encoder-decoder module to both database and query yield worse results. Reported with SuperPoint, $D_{LR} = 8$.

Method	Feature Size(↓)	Aachen Night(↑)	InLoc duc1(↑)	InLoc duc2(↑)
DISK+MNN+PQ	128-bit	70.4/81.6/91.8	34.3/52.0/63.6	22.9/34.4/42.0
Our DISK+SG	32-bit	**84.7/90.8/100.**	**40.9/59.1/71.2**	**36.6/62.6/74.8**
SIFT+MNN+PQ	128-bit	38.8/45.9/55.1	20.2/26.3/32.8	14.5/22.9/26.0
SIFT+PQ+SG	64-bit	19.4/23.5/27.6	19.7/24.2/25.8	6.9/12.2/18.3
Our SIFT+SG	32-bit	44.9/51.0/67.3	30.3/44.4/55.6	22.1/37.4/45.8

Table 5: Additional localization results with SIFT and DISK.

4.2 Ablation studies

The importance of joint training. We show the importance of retraining model weights by feeding our reconstructed descriptors into pre-trained network, as done in previous work [72]. The results are shown in Table 4. The significant performance drop of on Aachen night queries (from 82.7% to 71.4%) indicates pre-trained image matching pipeline performs poor without knowing the descriptor compression and needs re-training. Descriptor dimension reduction module trained within our pipeline also performs better than SceneSqueezer [72] (71.4% *vs*. 50%) when using the same pre-trained SuperGlue weights.

Symmetric *vs*. asymmetric design. We train a symmetric variant and show the results in Table 4. The symmetric pipeline is inferior to our asymmetric pipeline especially on InLoc dataset with about 10% gap, which justifies our design of asymmetric pipeline.

We additionally feed both database image and query image through encode-decoder module with asymmetrically trained pipeline. Interesting, the results outperform symmetric trained model. Considering the performance drop with compressed query descriptor is not much, for current visual positioning systems which perform server-side computation [2,47], it is feasible to send low dimension descriptor embedding of query images to the localization server only, minimizing data transmission and partially addressing privacy concerns [61].

Generalization to other feature We apply our pipeline to DISK [68] and SIFT, both of which are compressed to 32-bit. The results for visual localization tasks are shown in Table 5. For DISK, we compare with the baseline compression of product quantization (PQ). We can obtain better localization results with 1/4 memory. For SIFT, our method beats PQ-based methods, either with mutual nearest neighbor (MNN) or off-the-shelf pre-trained SuperGlue model.

Understanding the training process. We monitor the descriptor reconstruction loss when training with different embedding dimension. The reconstruction loss increases as we decrease the embedding dimension, as shown in Figure 4. When using low embedding dimension such as 4, it is impossible to reconstruct the original descriptor as information loss increases due to bottleneck design, but the matching performance (measured with AUC10, matching score and precision) is similar to uncompressed baseline. This validates our idea that though the descriptor information is heavily compressed, attention-based image matching pipeline works well when trained with such data.

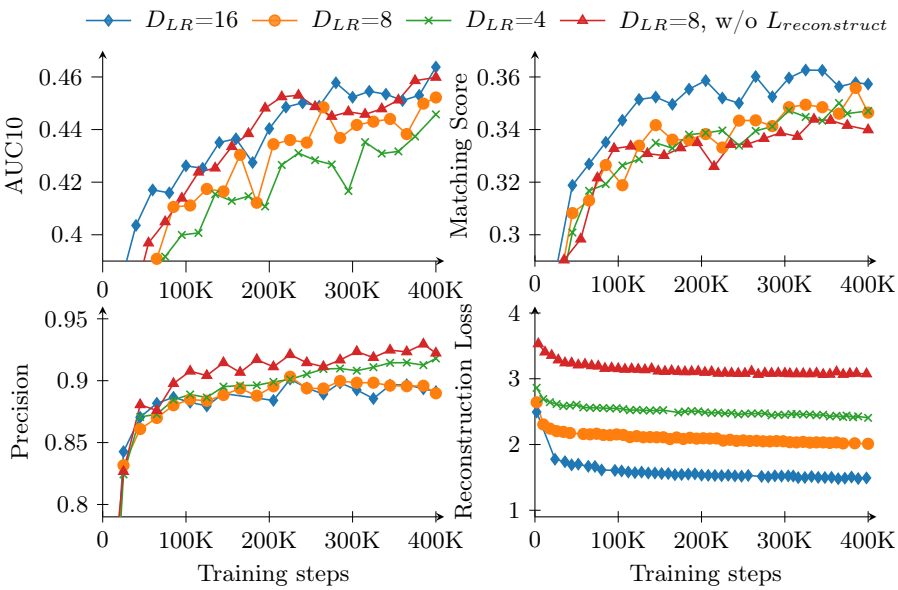

Fig. 4: Comparison of training process with varying D_{LR} (16/8/4). We monitor the AUC10, matching score, precision and descriptor reconstruction loss during the training process. Our pipeline performs well when decreasing bottleneck dimension though the reconstruction loss increases. Training without reconstruction loss ($L_{reconstruct}$) shows similar results, indicating matching loss can work as the only supervision signal.

We investigate the role of descriptor reconstruction loss by training with matching loss alone, as shown in Table 4. The performance is comparable to our full pipeline on Aachen Day-Night dataset and slightly worse on InLoc dataset. We monitor the reconstruction loss during training though it is not used during back-propagation in Figure 4. The model trained with only matching loss implicitly learn to recover original descriptor though the supervision is weaker compared to our full pipeline. This indicates the reconstruction loss is not important even optional with our framework. This is distinct from previous works [22, 72] which rely heavily on descriptor reconstruction loss. Our pipeline is matchness driven while descriptor reconstruction can be implicitly inferred during training.

5 Conclusion

We propose a novel framework to perform local descriptor compression by jointly training within image matching pipeline for visual localization. Each local descriptor can be represented with 1/256 original size with minor impact on the image matching performance. Our pipeline offers a significant reduction in storage requirements for structure-based visual localization methods while retaining high localization accuracy on several public datasets.

References

1. GlueFactory. `https://github.com/cvg/glue-factory` (Retrieved Jul 15,2024)
2. Niantic Lightship. `https://lightship.dev/products/vps` (Retrieved Jul 15,2024)
3. Alahi, A., Ortiz, R., Vandergheynst, P.: FREAK: Fast retina keypoint. In: CVPR (2012)
4. Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: NetVLAD: CNN architecture for weakly supervised place recognition. In: CVPR (2016)
5. Arnold, E., Wynn, J., Vicente, S., Garcia-Hernando, G., Monszpart, A., Prisacariu, V., Turmukhambetov, D., Brachmann, E.: Map-free visual relocalization: Metric pose relative to a single image. In: ECCV. Springer (2022)
6. Balntas, V., Li, S., Prisacariu, V.: RelocNet: Continuous metric learning relocalisation using neural nets. In: ECCV (2018)
7. Balntas, V., Riba, E., Ponsa, D., Mikolajczyk, K.: Learning local feature descriptors with triplets and shallow convolutional neural networks. In: Proc. BMVC. (2016)
8. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded up robust features. In: ECCV. Springer (2006)
9. Berton, G., Masone, C., Caputo, B.: Rethinking visual geo-localization for large-scale applications. In: CVPR (2022)
10. Berton, G., Trivigno, G., Caputo, B., Masone, C.: Eigenplaces: Training viewpoint robust models for visual place recognition. In: CVPR (2023)
11. Brachmann, E., Rother, C.: Learning less is more-6d camera localization via 3d surface regression. In: CVPR (2018)
12. Brachmann, E., Rother, C.: Expert sample consensus applied to camera re-localization. In: ICCV (2019)
13. Brachmann, E., Rother, C.: Visual camera re-localization from RGB and RGB-D images using DSAC. IEEE PAMI **44**(9), 5847–5865 (2021)
14. Campbell, D., Liu, L., Gould, S.: Solving the blind perspective-n-point problem end-to-end with robust differentiable geometric optimization. In: ECCV (2020)
15. Camposeco, F., Cohen, A., Pollefeys, M., Sattler, T.: Hybrid scene compression for visual localization. In: CVPR (2019)
16. Cao, S., Snavely, N.: Minimal scene descriptions from structure from motion models. In: CVPR (2014)
17. Chang, M.F., Zhao, Y., Shah, R., Engel, J.J., Kaess, M., Lucey, S.: Long-term visual map sparsification with heterogeneous GNN. In: CVPR (2022)
18. Chen, H., Luo, Z., Zhou, L., Tian, Y., Zhen, M., Fang, T., Mckinnon, D., Tsin, Y., Quan, L.: ASpanFormer: detector-free image matching with adaptive span transformer. In: ECCV (2022)
19. Cheng, W., Lin, W., Chen, K., Zhang, X.: Cascaded parallel filtering for memory-efficient image-based localization. In: ICCV (2019)
20. Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. NIPS (2013)
21. DeTone, D., Malisiewicz, T., Rabinovich, A.: SuperPoint: Self-supervised interest point detection and description. In: CVPR Workshop (2018)
22. Dong, H., Chen, X., Dusmanu, M., Larsson, V., Pollefeys, M., Stachniss, C.: Learning-based dimensionality reduction for computing compact and effective local feature descriptors. In: ICRA (2023)
23. Dusmanu, M., Rocco, I., Pajdla, T., Pollefeys, M., Sivic, J., Torii, A., Sattler, T.: D2-net: A trainable cnn for joint description and detection of local features. In: CVPR (2019)

24. Dymczyk, M., Lynen, S., Bosse, M., Siegwart, R.: Keep it brief: Scalable creation of compressed localization maps. In: Proc. IROS (2015)
25. He, K., Lu, Y., Sclaroff, S.: Local descriptors optimized for average precision. In: CVPR (2018)
26. Humenberger, M., Cabon, Y., Guerin, N., Morat, J., Leroy, V., Revaud, J., Rerole, P., Pion, N., de Souza, C., Csurka, G.: Robust image retrieval-based visual localization using kapture. arXiv:2007.13867 (2020)
27. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: ICML (2015)
28. Kendall, A., Grimes, M., Cipolla, R.: PoseNet: A convolutional network for real-time 6-dof camera relocalization. In: ICCV (2015)
29. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv:1412.6980 (2014)
30. Leutenegger, S., Chli, M., Siegwart, R.Y.: BRISK: Binary robust invariant scalable keypoints. In: ICCV (2011)
31. Li, Y., Snavely, N., Huttenlocher, D.P.: Location recognition using prioritized feature matching. In: ECCV (2010)
32. Li, Z., Snavely, N.: MegaDepth: Learning single-view depth prediction from internet photos. In: CVPR (2018)
33. Lindenberger, P., Sarlin, P.E., Pollefeys, M.: LightGlue: local feature matching at light speed. ICCV (2023)
34. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: RoBERTa: A robustly optimized bert pretraining approach. arXiv:1907.11692 (2019)
35. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV **60**, 91–110 (2004)
36. Luo, Z., Shen, T., Zhou, L., Zhu, S., Zhang, R., Yao, Y., Fang, T., Quan, L.: GeoDesc: Learning local descriptors by integrating geometry constraints. In: ECCV (2018)
37. Lynen, S., Sattler, T., Bosse, M., Hesch, J.A., Pollefeys, M., Siegwart, R.: Get out of my lab: Large-scale, real-time visual-inertial localization. In: Robotics: Science and Systems (2015)
38. Lynen, S., Zeisl, B., Aiger, D., Bosse, M., Hesch, J., Pollefeys, M., Siegwart, R., Sattler, T.: Large-scale, real-time visual–inertial localization revisited. The International Journal of Robotics Research **39**(9), 1061–1084 (2020)
39. Mera-Trujillo, M., Smith, B., Fragoso, V.: Efficient scene compression for visual-based localization. In: 3DV (2020)
40. Mishchuk, A., Mishkin, D., Radenovic, F., Matas, J.: Working hard to know your neighbor's margins: Local descriptor learning loss. NIPS (2017)
41. Mukundan, A., Tolias, G., Bursuc, A., Jégou, H., Chum, O.: Understanding and improving kernel local descriptors. IJCV **127**(11-12) (2019)
42. Mur-Artal, R., Tardós, J.D.: ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. IEEE Transactions on Robotics **33**(5), 1255–1262 (2017)
43. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: ICML. pp. 807–814 (2010)
44. Pautrat, R., Suárez, I., Yu, Y., Pollefeys, M., Larsson, V.: GlueStick: robust image matching by sticking points and lines together. ICCV (2023)
45. Qin, T., Li, P., Shen, S.: VINS-Mono: A robust and versatile monocular visual-inertial state estimator. IEEE Transactions on Robotics **34**(4), 1004–1020 (2018)

46. Radenović, F., Tolias, G., Chum, O.: Fine-tuning CNN image retrieval with no human annotation. IEEE PAMI **41**(7), 1655–1668 (2018)
47. Reinhardt, T.: Using global localization to improve navigation. `https://research.google/blog/using-global-localization-to-improve-navigation/` (Retrieved Jul 15,2024)
48. Revaud, J., Almazán, J., Rezende, R.S., Souza, C.R.d.: Learning with average precision: Training image retrieval with a listwise loss. In: CVPR (2019)
49. Revaud, J., De Souza, C., Humenberger, M., Weinzaepfel, P.: R2D2: Reliable and repeatable detector and descriptor. NIPS (2019)
50. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: An efficient alternative to sift or surf. In: ICCV (2011)
51. Sarlin, P.E.: Image matching challenge 2020 winner entry (2020), `https://www.cs.ubc.ca/research/image-matching-challenge/2020/submissions/sid-00612-sp-k2048-nms4-refine2-r1600forcecubic-down128-masked-d.001-adapt50_sg-t.2-it150_degensac-th1.1/`
52. Sarlin, P.E., Cadena, C., Siegwart, R., Dymczyk, M.: From coarse to fine: Robust hierarchical localization at large scale. In: CVPR (2019)
53. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: SuperGlue: Learning feature matching with graph neural networks. In: CVPR (2020)
54. Sarlin, P.E., Dusmanu, M., Schönberger, J.L., Speciale, P., Gruber, L., Larsson, V., Miksik, O., Pollefeys, M.: LaMAR: Benchmarking localization and mapping for augmented reality. In: ECCV (2022)
55. Sattler, T., Leibe, B., Kobbelt, L.: Fast image-based localization using direct 2d-to-3d matching. In: ICCV (2011)
56. Sattler, T., Maddern, W., Toft, C., Torii, A., Hammarstrand, L., Stenborg, E., Safari, D., Okutomi, M., Pollefeys, M., Sivic, J., et al.: Benchmarking 6DOF outdoor visual localization in changing conditions. In: CVPR (2018)
57. Sattler, T., Weyand, T., Leibe, B., Kobbelt, L.: Image retrieval for image-based localization revisited. In: Proc. BMVC. (2012)
58. Sattler, T., Zhou, Q., Pollefeys, M., Leal-Taixe, L.: Understanding the limitations of CNN-based absolute camera pose regression. In: CVPR (2019)
59. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: CVPR (2016)
60. Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., Fitzgibbon, A.: Scene coordinate regression forests for camera relocalization in rgb-d images. In: CVPR (2013)
61. Speciale, P., Schonberger, J.L., Kang, S.B., Sinha, S.N., Pollefeys, M.: Privacy preserving image-based localization. In: CVPR (2019)
62. Sun, J., Shen, Z., Wang, Y., Bao, H., Zhou, X.: LoFTR: Detector-free local feature matching with transformers. In: CVPR (2021)
63. Taira, H., Okutomi, M., Sattler, T., Cimpoi, M., Pollefeys, M., Sivic, J., Pajdla, T., Torii, A.: InLoc: Indoor visual localization with dense matching and view synthesis. In: CVPR (2018)
64. Tang, S., Tang, S., Tagliasacchi, A., Tan, P., Furukawa, Y.: NeuMap: neural coordinate mapping by auto-transdecoder for camera localization. In: CVPR (2023)
65. Tian, Y., Fan, B., Wu, F.: L2-Net: Deep learning of discriminative patch descriptor in euclidean space. In: CVPR (2017)
66. Tian, Y., Yu, X., Fan, B., Wu, F., Heijnen, H., Balntas, V.: SOSNet: second order similarity regularization for local descriptor learning. In: CVPR (2019)
67. Toft, C., Maddern, W., Torii, A., Hammarstrand, L., Stenborg, E., Safari, D., Okutomi, M., Pollefeys, M., Sivic, J., Pajdla, T., et al.: Long-term visual localization revisited. IEEE PAMI **44**(4), 2074–2088 (2020)

68. Tyszkiewicz, M., Fua, P., Trulls, E.: DISK: Learning local features with policy gradient. NIPS (2020)
69. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. NIPS (2017)
70. Wang, S., Kannala, J., Barath, D.: DGC-GNN: descriptor-free geometric-color graph neural network for 2D-3D matching. arXiv:2306.12547 (2023)
71. Xue, F., Budvytis, I., Cipolla, R.: Pram: Place recognition anywhere model for efficient visual localization (2024)
72. Yang, L., Shrestha, R., Li, W., Liu, S., Zhang, G., Cui, Z., Tan, P.: SceneSqueezer: Learning to compress scene for camera relocalization. In: CVPR (2022)
73. Zhang, X., Liu, Y.H.: Efficient map sparsification based on 2D and 3D discretized grids. In: CVPR (2023)
74. Zhao, X., Wu, X., Chen, W., Chen, P.C.Y., Xu, Q., Li, Z.: Aliked: A lighter keypoint and descriptor extraction network via deformable transformation. IEEE Transactions on Instrumentation & Measurement **72**, 1–16 (2023)
75. Zhou, Q., Agostinho, S., Ošep, A., Leal-Taixé, L.: Is geometry enough for matching in visual localization? In: ECCV (2022)