SFD2: Semantic-guided Feature Detection and Description

Fei Xue Ignas Budvytis Roberto Cipolla
University of Cambridge
{fx221, ib255, rc10001}@cam.ac.uk

Abstract

Visual localization is a fundamental task for various applications including autonomous driving and robotics. Prior methods focus on extracting large amounts of often redundant locally reliable features, resulting in limited efficiency and accuracy, especially in large-scale environments under challenging conditions. Instead, we propose to extract globally reliable features by implicitly embedding high-level semantics into both the detection and description processes. Specifically, our semantic-aware detector is able to detect keypoints from reliable regions (e.g. building, traffic lane) and suppress unreliable areas (e.g. sky, car) implicitly instead of relying on explicit semantic labels. This boosts the accuracy of keypoint matching by reducing the number of features sensitive to appearance changes and avoiding the need of additional segmentation networks at test time. Moreover, our descriptors are augmented with semantics and have stronger discriminative ability, providing more inliers at test time. Particularly, experiments on long-term large-scale visual localization Aachen Day-Night and RobotCar-Seasons datasets demonstrate that our model outperforms previous local features and gives competitive accuracy to advanced matchers but is about 2 and 3 times faster when using 2k and 4k keypoints, respectively. Code is available at https://github.com/feixue94/sfd2.

1. Introduction

Visual localization is key to various applications including autonomous driving and robotics. Structure-based algorithms [54, 57, 64, 69, 73, 79] involving mapping and localization processes still dominate in large-scale localization. Traditionally, handcrafted features (*e.g.* SIFT [3, 35], ORB [53]) are widely used. However, these features are mainly based on statistics of gradients of local patches and thus are prone to appearance changes such as illumination and season variations in the long-term visual localization task. With the success of CNNs, learning-based features [14, 16, 37, 45, 51, 76, 81] are introduced to replace handcrafted ones and have achieved excellent performance.

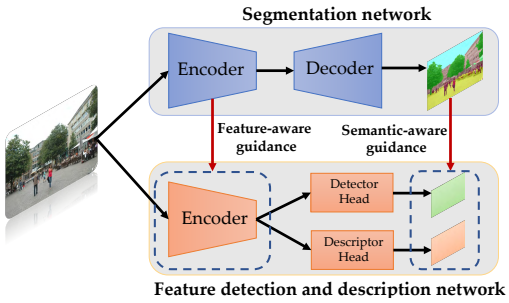

Figure 1. **Overview of our framework.** Our model implicitly incorporates semantics into the detection and description processes with guidance of an off-the-shelf segmentation network during the training process. Semantic- and feature-aware guidance are adopted to enhance its ability of embedding semantic information.

With massive data for training, these methods should be able to automatically extract keypoints from more reliable regions (*e.g.* building, traffic lane) by focusing on discriminative features [64]. Nevertheless, due to the lack of explicit semantic signals for training, their ability of selecting globally reliable keypoints is limited, as shown in Fig. 2 (detailed analysis is provided in Sec. B.1 in the supplementary material). Therefore, they prefer to extract locally reliable features from objects including those which are not useful for long-term localization (*e.g.* sky, tree, car), leading to limited accuracy, as demonstrated in Table 2.

Recently, advanced matchers based on sparse keypoints [8, 55, 65] or dense pixels [9, 18, 19, 33, 40, 47, 67, 74] are proposed to enhance keypoint/pixel-wise matching and have obtained remarkable accuracy. Yet, they have quadratic time complexity due to the attention and correlation volume computation. Moreover, advanced matchers rely on spatial connections of keypoints and perform image-wise matching as opposed to fast point-wise matching, so they take much longer time than nearest neighbor matching (NN) in both mapping and localization processes because of a large number of image pairs (much larger than the number of images) [8]. Alternatively, some works leverage semantics [64, 73, 79] to filter unstable features to eliminate wrong correspondences and report close even better accu-

racy than advanced matchers [79]. However, they require additional segmentation networks to provide semantic labels at test time and are fragile to segmentation errors.

Instead, we implicitly incorporate semantics into a local feature model, allowing it to extract robust features automatically from a single network in an end-to-end fashion. In the training process, as shown in Fig. 1, we provide explicit semantics as supervision to guide the detection and description behaviors. Specifically, in the detection process, unlike most previous methods [14, 16, 32, 37, 51] adopting exhaustive detection, we employ a semantic-aware detection loss to encourage our detector to favor features from reliable objects (*e.g.* building, traffic lane) and suppress those from unreliable objects (*e.g.* sky). In the description process, rather than utilizing triplet loss widely used for descriptor learning [16, 41], we employ a semantic-aware description loss consisting of two terms: inter- and intra-class losses. The inter-class loss embeds semantics into descriptors by enforcing features with the same label to be close and those with different labels to be far. The intra-class loss, which is a soft-ranking loss [23], operates on features in each class independently and differentiates these features from objects of the same label. Such use of soft-ranking loss avoids the conflict with inter-class loss and retains the diversity of features in each class (*e.g.* features from buildings usually have larger diversity than those from traffic lights). With semantic-aware descriptor loss, our model is capable of producing descriptors with stronger discriminative ability. Benefiting from implicit semantic embedding, our method avoids using additional segmentation networks at test time and is less fragile to segmentation errors.

As the local feature network is much simpler than typical segmentation networks *e.g.* UperNet [10], we also adopt an additional feature-consistency loss on the encoder to enhance its ability of learning semantic information. To avoid using costly to obtain ground-truth labels, we train our model with outputs of an off-the-shelf segmentation network [11, 34], which has achieved SOTA performance on the scene parsing task [83], but other semantic segmentation networks (*e.g.* [10]) can also be used.

An overview of our system is shown in Fig. 1. We embed semantics implicitly into the feature detection and description network via the feature-aware and semantic-aware guidance in the training process. At test time, our model produces semantic-aware features from a single network directly. We summarize contributions as follows:

- We propose a novel feature network which implicitly incorporates semantics into detection and description processes at training time, enabling the model to produce semantic-aware features end-to-end at test time.
- We adopt a combination of semantic-aware and feature-aware guidance strategy to make the model embed semantic information more effectively.
- Our method outperforms previous local features on the long-term localization task and gives competitive accuracy to advanced matchers but has higher efficiency.

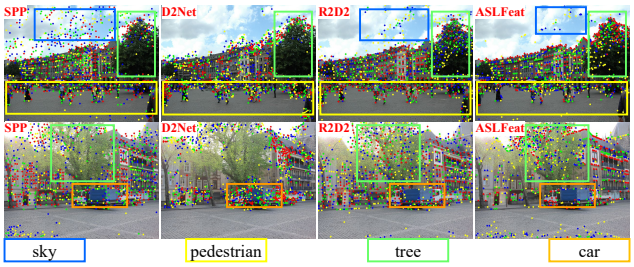

Figure 2. **Locally reliable features**. We show top 1k keypoints (reliability high→low: 1-250, 251-500, 501-750, 751-1000) of prior local features including SPP [14], D2Net [16], R2D2 [51], and ASLFeat [37]. They indiscriminatively give high reliability to patches with rich textures even from objects *e.g.* sky, tree, pedestrian and car, which are reliable for long-term localization (best view in color).

Experiments show our method achieves a better trade-off between accuracy and efficiency than advanced matchers [8, 55, 65] especially on devices with limited computing resources. We organize the rest of this paper as follows. In Sec. 2, we introduce related works. In Sec. 3, we describe our method in detail. We discuss experiments and limitations in Sec. 4 and Sec. 5 and conclude the paper in Sec. 6.

2. Related Work

In this section, we discuss related work on visual localization, feature extraction and matching, and knowledge distillation.

Visual localization. Visual localization methods can be roughly categorized as image-based and structure-based. Image-based systems recover camera poses by finding the most similar one in the database with global features, *e.g.* NetVLAD [2], CRN [27]. Due to the limited number of images in the database, they can only give approximate poses. To obtain more precise poses, structure-based methods build a sparse 3D map via SfM and estimate the pose via PnP from 2D-3D correspondences [12, 54, 57, 58, 69, 79]. Some other works have tried to predict the camera pose directly from images, *e.g.* PoseNet [28] and its variations [80], or regress scene coordinates [5–7, 26]. However, the former have been proved to perform similar to image retrieval [61] and latter are hard to scale to large-scale scenes [31].

Local features. Handcrafted features [3, 35, 53] have been investigated for decades and we refer readers a survey [38] for more details and focus on learned features. With the success of CNNs, learned features are proposed to

replace handcrafted descriptors [15,17,36,41–43,49,71,72], detectors [13,68,70], or both [14,16,32,37,51,76,81]. HardNet [41] focuses on metric learning by maximizing the distance between the closest positive and negative examples. Instead of using pixel-wise correspondences for training, CAPS [77], PoSFeat [32] and PUMP [50] utilize camera pose and local consistency of matches for supervision. SuperPoint (SPP) [14] takes keypoint detection as a supervised task, training detector from synthetic geometric shapes. D2-Net [16] uses local maxima across the channels as score map. R2D2 [51] considers both the repeatability and reliability and adopts the average precision loss [23] for descriptor training. ASLFeat [37] employs deformable CNNs to learn shape-aware dense features. As they focus mainly on *local reliability* of features, regardless of their superior accuracy to handcrafted features, their performance is limited in the long-term large-scale localization task. To further improve the accuracy, some works [22,46,75] learn to filter unstable keypoints with extra matching score, repeatability or semantic labels. Essentially different with these methods, our model detects and extracts semantic-aware features automatically in an end-to-end fashion. As a result, our features are able to produce more accurate localization results.

Advanced matcher. As NN matching is unable to incorporate spatial connections of keypoints for matching, advanced matchers are proposed to enhance the accuracy by leveraging the spatial context of a set of keyppoints [8, 55, 65] or an image patch [9, 18, 33, 52, 67, 84]. SuperGlue (SPG) [55] utilizes graph neural networks with attention mechanism to propagate information among keypoints. It produces impressive accuracy, whereas its time complexity is quadratic to the number of keypoints. This problem is partially mitigated by using seeded matching [8] and cluster matching [65], but the time is still thousands of times slower than NN matching. Dense matchers [9, 33, 52, 67] compute pixel-wise correspondence from correlation volumes, so they undergo the high time and memory cost as sparse matchers [8, 55, 65]. Moreover, advanced matchers operates on image pairs as opposed to keypoints, so considering the number of image pairs, systems with advanced matchers could be much slower in real applications, as analyzed in [8]. In this paper, we embed high-level semantic information into local features implicitly to enhance both feature detection and description, enabling our model with simple NN matching to yield comparable results to advanced matchers. Our work provides a good trade-off of time and accuracy especially on devices with limited computing resources.

Visual semantic localization. Compared to local features, high-level semantics are more robust to appearance changes and have been widely used in visual localization [7, 25, 26, 30, 31, 44, 62–64, 66, 73, 78]. LLN [78] and SVL [63] use the discriminative landmarks for place recognition. ToDayGAN [1] transfers night images to day images with GAN [20]. MFC [30], SMC [73], SSM [64], and DASGIL [25] incorporate segmentation networks into a standard localization pipeline to reject semantically-inconsistent matches. More recently, LBR [79] learns to recognize global instances for both coarse and fine localization. In fine localization, it filters unstable features and conducts instance-wise matching, achieving close accuracy to advanced matchers [55]. Unlike these methods, which require additional models to provide explicit semantic labels at test time, we embed the semantic information into the network and produce semantic-aware features directly from a single network.

Knowledge distillation. Knowledge distillation techniques have been widely used for tasks including model compression [54] and knowledge transfer [82]. Our usage of pseudo ground-truth local reliability and semantic labels predicted by off-the-shelf networks is more like a knowledge transfer task. In this paper, we focus mainly on how to effectively leverage the high-level semantics for low-level feature extraction.

3. Method

As shown in Fig. 1, our model consists of an encoder $\mathcal{F}_{enc}$ and two decoders $\mathcal{F}_{det}$ and $\mathcal{F}_{desc}$. $\mathcal{F}_{enc}$ extracts high-level features $\mathbf{X}$ from image $\mathbf{I} \in R^{3\times H\times W}$. $\mathcal{F}_{det}$ predicts the reliability map $\mathbf{S} \in R^{H\times W}$ and $\mathcal{F}_{desc}$ produces descriptors $\mathbf{X}_{desc} \in R^{128\times \frac{H}{4}\times \frac{W}{4}}$. H and W are the height and width of the image. In this section, we give details about how to implicitly incorporate semantic information into our feature detection and description processes.

3.1. Semantic-guided Feature Detection

The detector predicts the reliability map as $\mathbf{S} = \mathcal{F}_{det}(\mathbf{X})$. Previously, the reliability map $\mathbf{S}$ is defined by the richness of textures in patches (*e.g.* response to corners [14] or blobs [35]). Recently, learned local features [16, 32, 37, 51] define the reliability on the discriminative ability of descriptors. As shown in Fig. 2, these two definitions, however, only reveal the reliability of pixels at a local level but lack the stability at a global level. Instead, we redefine the reliability of features by taking both the local reliability $\mathbf{S}_{rel}$ and global stability $\mathbf{S}_{sta}$ into consideration.

Local reliability. Local reliability shows the robustness of a keypoint to appearance changes and viewpoint variations. Previous learning-based features adopt two strategies for reliable feature learning: learning from groundtruth corners [14] and learning from the discriminative ability of descriptors [16, 32, 37, 51]. We find that corners [14] are more robust compared to purely learned detectors, as shown in [50, 79], where SPP detector achieves better results when replacing other detectors. Therefore, following [54], we use the detection score $\mathbf{S}_{rel}$ of SPP [14] as pseudo groundtruth, which is one of the best corner detectors. At the same time,

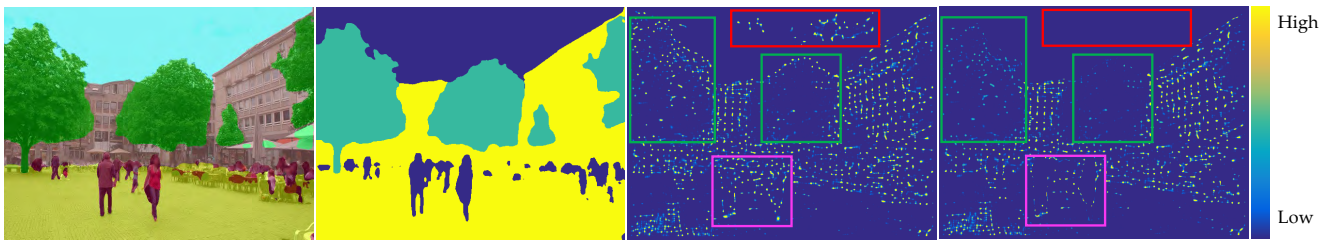

Figure 3. **Semantic-guided feature detection.** From left to right: semantic segmentation mask predicted by [11, 34], stability map $\mathbf{S}_{sta}$ generated according to Table 1, local reliability map $\mathbf{S}_{rel}$ produced by SuperPoint [14], and the final global reliability map $\mathbf{S}$. Local reliability map gives very high score to clouds (red), trees (green), and pedestrians (pink) in addition to buildings, while the global reliability map removes unstable regions (sky, pedestrians), suppresses short-term objects (trees), and retains stable areas (buildings).

local reliability is slightly adjusted by the discriminative ability of descriptors in the training process (see Sec. 3.2).

Global stability. The global stability of a pixel is assigned based on the semantic label which it belongs to. Specifically, we group all 120 semantic labels in ADE20k dataset [83], according to how they change over time, into four categories, denoted as *Volatile*, *Dynamic*, *Short-term*, and *Long-term* in Table 1. Volatile objects (*e.g.* sky, water) are constantly changing and are redundant for localization. Dynamic objects (*e.g.* car, pedestrian) are moving everyday and could cause localization error by introducing wrong matches. Short-term objects (*e.g.* tree) can be used for short-term localization tasks (*e.g.* VO/SLAM), yet they are sensitive to changes of illumination (low albedo) and season conditions. Long-term objects (*e.g.* building, traffic light) are resistant to aforementioned changes and are ideal for long-term localization.

Instead of directly **filtering** unstable features [64, 73, 79], we **rerank** features with stability values assigned empirically according to the extent of desired suppression. In detail, Long-term objects are robust for both short and long-term localization, so their stability value is set to 1.0. Short-term objects are useful for short-term localization, so we set their stability to 0.5. The stability value of Volatile and Dynamic categories is set to 0.1 as they are not useful for both short/long-term localization. Note that we set it to 0.1 rather than 0. Our reranking strategy encourages the model to use stable features preferentially and uses keypoints from other objects as compensation when insufficient stable keypoints can be found, increasing the robustness of our model to various tasks (*e.g.* feature matching, short-term localization). Fig. 3 shows stability map $\mathbf{S}_{sta}$ transformed from Table 1. Our current global stability is assigned based on predefined semantic labels, but a learned one might provide better performance and deserves further exploration.

Semantic-guided detection. The global reliability map $\mathbf{S}_{gt}$ is generated by multiplying the local reliability map $\mathbf{S}_{rel}$ and global stability map, as $\mathbf{S}_{gt} = \mathbf{S}_{rel} \odot \mathbf{S}_{sta}$ ($\odot$ is element-wise multiplication). Fig. 3 shows that local reliability map gives high score for all pixels with rich textures even those from the sky, pedestrians, and trees, which are useless for localization. However, the global reliability map considering both local reliability and global stability discards these sensitive features and suppresses short-term keypoints effectively. The detection loss is defined as:

$$L_{det} = BCE(\mathbf{S}, \mathbf{S}_{gt}), \tag{1}$$

where BCE denotes the *binary cross-entropy* loss.

Category	Volatile	Dynamic	Short-term	Long-term	Stability
sky, water	✓				0.1
vehicle, pedestrian		✓			0.1
plant, grass			✓		0.5
building, traffic light				✓	1.0

Table 1. **Stability map**. Semantic labels are categorized into four groups denoted as *Volatile*, *Dynamic*, *Short-term*, and *Long-term*. Four categories are empirically assigned with different stability values according to their robustness to appearance changes.

3.2. Semantic-guided Feature Description

We also enhance the discriminative ability of descriptors by embedding semantics into them directly. Unlike previous descriptors [14, 37, 41, 50, 51, 77], which only differentiate keypoints based on local patch information, our descriptors enforce similarities of features in the same class while retain dissimilarities for intra-class matching. However, the two forces conflict with each other during the training process, because class-level discriminative ability needs to squeeze the space of descriptors in the same class and intra-class discriminative ability has to increase the space. A simple solution could be to set a hard margin for all classes (Fig. 4 left), but it would lead to the loss of objects' inner diversity (*e.g.*, almost all traffic lights are similar but different buildings vary dramatically), which is indispensable for intra-class matching. To solve this problem, we design the inter-class and intra-class losses based on two different metrics (Fig. 4 right).

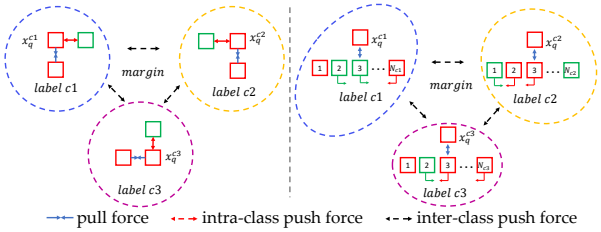

Figure 4. **Semantic-guided feature description**. Simply optimizing inter- and intra-class losses with hard margins may cause accuracy loss due to two conflicting forces (push forces of inter- and intra-class) (left). Instead, we optimize the intra-class force with a hard margin, but apply a soft ranking loss for inter-class force to avoid conflicts and retain the inner diversity of each object (right).

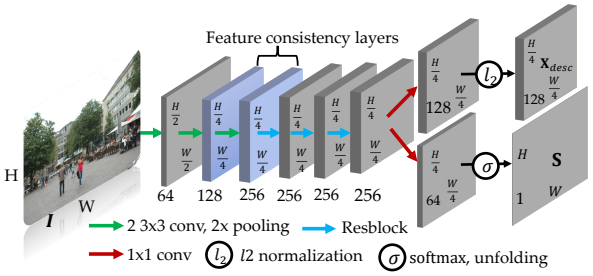

Figure 5. **Architecture of our network.** Features are 4× downsampling to save time and memory cost. Resblocks [24] are adopted to enhance the model's capacity. We enforce the consistency between outputs of the middle two layers of our encoder and features of the segmentation network to enhance the ability of our model to embed semantic information during the training process.

Inter-class loss. We first enforce the semantic consistency of features by maximizing the Euclidean distance between descriptors with different labels. This allows features to find correspondences from candidates with the same labels, reducing the search space and thus improving the matching accuracy. We define the inter-class loss based on triplet loss with a hard margin to separate all possible positive and negative keypoints with different labels in a batch:

$$L_{inter} = \frac{1}{N}\sum(||\mathbf{x}_i^{c_1} - \mathbf{x}_j^{c_1}||_2 - ||\mathbf{x}_i^{c_1} - \mathbf{x}_k^{c_2}||_2 + m), \quad (2)$$

where $\mathbf{x}_i^{c_1}$, $\mathbf{x}_j^{c_1}$, and $\mathbf{x}_k^{c_2}$ are vectorized descriptors with labels of c_1, c_1, and c_2 ($c_1 \neq c_2$). m is the margin and set to 1.0. This loss is conducted on features in the whole batch and N is the total number of features in a batch.

Intra-class loss. To make sure that the intra-class loss doesn't conflict with the inter-class loss, we relax the limitation of distances between descriptors with the same label. Instead of using triplet loss with hard margins, we adopt a soft ranking loss [23] by optimizing the rank of positive and negative samples rather than their distances. We use the same strategy as [51] to generate positive and negative samples for each feature $\mathbf{x}_i^c$ from self and the other images respectively, but enforce both positive and negative samples to share the same class label c as $\mathbf{x}_i^c$. By optimizing ranks of all samples rather than forcing a hard boundary between positive and negative pairs as the triplet loss with a hard margin does, the soft ranking loss also retains the diversity of features on objects in the same class, as shown in Fig. 4 (right). The ranking loss is based on the averaging precision (AP) loss [23, 51]:

$$L_{intra} = \frac{1}{C}\sum_{c=1}^{C}\frac{1}{N_c}\sum_{i=1}^{N_c}(1 - AP(\mathbf{x}_i^c, \mathbf{S}^{\mathbf{x}_i^c})), \quad (3)$$

where $\mathbf{x}_i^c$ and $\mathbf{S}^{\mathbf{x}_i^c}$ are the query descriptor with label c and corresponding predicted local reliability. C and N_c are the total number of classes and features in class c. Note that the AP loss for sample $\mathbf{x}_i^c$ is weighted by its reliability $\mathbf{S}^{\mathbf{x}_i^c}$.

Our final descriptor loss L_{desc} is the combination of L_{inter} and L_{intra}, balanced by w_{inter} and w_{intra}:

$$L_{desc} = w_{inter}L_{inter} + w_{intra}L_{intra}. \quad (4)$$

3.3. Implicit Feature Guidance

With semantic-aware detection and description losses, our model is able to learn semantic-aware features. However, compared with feature learning, semantic prediction is a more complicated task, requiring powerful encoders and aggregation layers [4, 34] for semantic-aware feature embedding. To further improve the ability of our model to embed semantic information, we take inspiration from current knowledge distillation tasks [21] and introduce a feature consistency loss on intermediate outputs of the first three layers of the encoder.

Fig. 5 shows the architecture of our network. We take intermediate outputs of the encoder of ConvNeXt [34] as supervision signal and enforce l_1 loss on features of the middle 2 layers of our model:

$$L_{feat} = \frac{1}{2}\sum_{i=1}^{2}|\mathbf{X}_i - \mathbf{X}_i^{ConvNeXt}|, \quad (5)$$

where $\mathbf{X}_i$ and $\mathbf{X}_i^{ConvNeXt}$ are features of the ith layer in our model and ConvNeXt [34], respectively. The total loss L_{total} is the combination of detection loss L_{det}, description loss L_{desc}, and feature consistency loss L_{feat} with weights of w_{det}, w_{desc}, and w_{feat}:

$$L_{total} = w_{det}L_{det} + w_{desc}L_{desc} + w_{feat}L_{feat}. \quad (6)$$

4. Experiments

We first give implementation details. Then, we test our method on visual localization tasks in Sec. 4.1 and analyze the running time in Sec. 4.2. Finally, we perform an ablation

Group	Method	Day $(2°, 0.25m)/(5°, 0.5m)/(10°, 5m)$	Night $(2°, 0.25m)/(5°, 0.5m)/(10°, 5m)$
C	AS_v1.1 [57]	85.3 / 92.2 / 97.9	39.8 / 49.0 / 64.3
	CSL [69]	52.3 / 80.0 / 94.3	29.6 / 40.8 / 56.1
	CPF [12]	76.7 / 88.6 / 95.8	33.7 / 48.0 / 62.2
	Ours	**88.2 / 96.0 / 98.7**	**87.8 / 94.9 / 100.0**
S	SSM [64]	71.8 / 91.5 / 96.8	58.2 / 76.5 / 90.8
	VLM [78]	62.4 / 71.8 / 79.9	35.7 / 44.9 / 54.1
	SMC [73]	52.3 / 80.0 / 94.3	29.6 / 40.8 / 56.1
	LBR [79]	**88.3** / 95.6 / **98.8**	84.7 / 93.9 / **100.0**
	Ours	88.2 / **96.0** / 98.7	**87.8 / 94.9 / 100.0**
L	SIFT [35]	82.8 / 88.1 / 93.1	30.6 / 43.9 / 58.2
	SPP [14]	80.5 / 87.4 / 94.2	42.9 / 62.2 / 76.5
	D2Net [16]	84.8 / 92.6 / 97.5	84.7 / 90.8 / 96.9
	R2D2 [51]		76.5 / 90.8 / **100.0**
	SIFT+CAPS [35, 77]		77.6 / 86.7 / 99.0
	SPP+CAPS [14, 77]		82.7 / 87.8 / **100.0**
	SPP+LISRD [14, 49]		78.6 / 86.7 / 98.0
	SPP+PUMP [14, 50]		74.4 / 88.0 / 98.4
	R2D2+PUMP [14, 50]		73.3 / 86.9 / 98.4
	R2D2+LLF [14, 68]		72.4 / 90.8 / 99.0
	SOSNet+D2D [70, 72]		73.5 / 83.7 / 96.9
	PoSFeat [32]		81.6 / 90.8 / **100.0**
	ASLFeat [37]		81.6 / 87.8 / **100.0**
	Ours	**88.2 / 96.0 / 98.7**	**87.8 / 94.9 / 100.0**
M	ENCNet [52]		76.5 / 84.7 / 98.0
	Dual-RCNet [33]		79.6 / 88.8 / **100.0**
	PDCNet [74]		80.6 / 87.8 / **100.0**
	DGCNet [40]	22.9 / 49.8 / 84.7	14.3 / 37.8 / 79.6
	Pixloc [56]	84.7 / 94.2 / **98.8**	81.6 / 93.9 / **100.0**
	AHM [18]	47.8 / 72.2 / 91.3	30.6 / 53.1 / 78.6
	S2DNet [19]	84.5 / 90.3 / 95.3	74.5 / 82.7 / 94.9
	Patch2Pix [84]	84.6 / 92.1 / 96.5	82.7 / 92.9 / 99.0
	SPP+SPG [14, 55]	**89.6** / 95.4 / **98.8**	86.7 / 93.9 / **100.0**
	SPP+SGMNet [8, 14]	86.8 / 94.2 / 97.7	83.7 / 91.8 / 99.0
	SPP+ClusterGNN [14, 65]	89.4 / 95.5 / 98.5	81.6 / 93.9 / **100.0**
	Ours	88.2 / **96.0** / 98.7	**87.8 / 94.9 / 100.0**

Table 2. **Results on Aachen dataset [59, 60].** The best and second best results are highlighted with **bold** and red fonts.

Group	Method	Day $(2°, 0.25m)/(5°, 0.5m)/(10°, 5m)$	Night $(2°, 0.25m)/(5°, 0.5m)/(10°, 5m)$
S	LBR [79]	**89.1 / 96.1 / 99.3**	77.0 / 90.1 / **99.5**
	Ours	88.2 / 96.0 / 98.7	**78.0 / 92.1 / 99.5**
H	SIFT [35]	72.2 / 78.4 / 81.7	19.4 / 23.0 / 27.2
	SPP [14]	87.9 / 93.6 / 96.8	70.2 / 84.8 / 93.7
	D2Net [16]	84.1 / 91.0 / 95.5	63.4 / 83.8 / 92.1
	R2D2 [51]	**88.8** / 95.3 / 97.8	72.3 / 88.5 / 94.2
	ASLFeat [37]	88.0 / 95.4 / 98.2	70.7 / 84.3 / 94.2
	CAPS+SIFT [35, 77]	82.4 / 91.3 / 95.9	61.3 / 83.8 / 95.3
	LISRD+SPP [14, 49]		73.3 / 86.9 / 97.9
	LLF+R2D2 [14, 68]		71.2 / 81.2 / 94.2
	PoSFeat [32]		73.8 / 87.4 / 98.4
	Ours	88.2 / **96.0 / 98.7**	**78.0 / 92.1 / 99.5**
M	SPP+SGMNet [8, 14]	88.7 / **96.2** / 98.9	75.9 / 89.0 / 99.0
	SPP+SPG [14, 55]	**89.8** / 96.1 / **99.4**	77.0 / 90.6 / **100.0**
	Patch2Pix [84]	86.4 / 93.0 / 97.5	72.3 / 88.5 / 97.9
	LoFTER [67]	88.7 / 95.6 / 99.0	**78.5** / 90.6 / 99.0
	ASpanFormer [9]	89.4 / 95.6 / 99.0	77.5 / 91.6 / 99.5
	Ours	88.2 / 96.0 / 98.7	78.0 / **92.1** / 99.5

Table 3. **Results on Aachen_v1.1 dataset [59, 60].** The best and second best results are highlighted with **bold** and red fonts.

study in Sec. 4.3. More implementation details, results and analysis are provided in the **supplementary material**.

Implementation. We use the identical training dataset as R2D2 [51]. The training dataset consists of reference images in Aachen_v1.0 dataset [60] and web images. As R2D2 [51] and LBR [79], training images are augmented with style transfer. To mitigate segmentation uncertainties caused by style transfer, semantic labels of stylized images are generated from their corresponding normal images. The network is implemented in PyTorch [48] and trained using Adam [29] optimizer with $\beta_1 = 0.9, \beta_2 = 0.99$, batch size of 4, weight decay of 4×10^{-4} on a single 3090Ti GPU for 40 epochs. The hyper-parameter w_{intra} is set to 0.5, while w_{inter}, w_{det}, w_{desc}, and w_{feat} are set to 1.0.

4.1. Long-term Large-scale Localization

We test our method on Aachen (v1.0 and v1.1) [59, 60] and RobotCar-Seasons (RoCaS) [39, 60] datasets under various illumination, season, and weather conditions. Aachen_v1.0 contains 4,328 reference and 922 (824 day, 98 night) query images captured around the Aachen city center. Aachen_v1.1 expands v1.0 by adding 2,369 reference and 93 night query images. RoCaS has 26,121 reference and 11,934 query images. It is challenging because of various conditions of day query images (rain, snow, dusk, winter) and poor lighting of night query images in suburban areas. We adopt the success ratio at error thresholds of $(2°, 0.25m), (5°, 0.5m), (10°, 5m)$ as metric. We additionally provide results on Extended CMU-Seasons dataset in the **supplementary material**.

Baselines. Baselines include classic systems (C) *e.g.* AS_v1.1 [57], CSL [69], and CPF [12] and methods using semantics (S), *e.g.* LLN [78], SMC [73], SSM [64], DASGIL [25], ToDayGAN [1] and LBR [79]. We also compare our model with learned features [14, 16, 37, 50, 51, 77] (L). As prior methods [14, 16, 37, 50, 51, 77], we use HLoc [54] pipeline for reconstruction and mutual nearest matching (MNN). NetVLAD [2] is used to offer 50 and 20 candidates for Aachen and RoCaS datasets, respectively. We additionally compare our approach with advanced sparse/dense matchers (M) *e.g.*, Superglue (SPG) [55], SGMNet [8], ClusterGNN [65] and ASpanFormer [9], LoFTER [67], Patch2Pix [84], Dual-RCNet [33]. Their results are obtained from the visual benchmark[1] or original papers.

Comparison with classic methods (C). As shown in Table 2 and 4, our model outperforms all classic methods. As most these methods use SIFT [35], they are more sensitive to weather and illumination changes than learned features.

Comparison with methods using explicit semantics (S). By leveraging semantic labels to filter potentially wrong matches, these models achieve better performance for day and night images in Table 2 and 4 but require segmentation results at test time. Our model outperforms all other approaches (except LBR [79]). LBR [79] reports

[1] https://www.visuallocalization.net/benchmark/

Group	Method	day	night	night-rain
		$(2°, 0.25m)/(5°, 0.5m)/(10°, 5m)$		
C	AS [57]	43.6 / 76.0 / 94.0	1.6 / 3.9 / 10.5	2.0 / 10.9 / 18.0
	CSL [69]	45.3 / 73.5 / 90.1	0.2 / 0.9 / 5.3	0.9 / 4.3 / 9.1
	CPF [12]	48.0 / 78.0 / 94.2	2.3 / 6.6 / 15.3	4.5 / 12.3 / 18.6
	Ours	**56.9 / 81.6 / 97.4**	**27.6 / 66.2 / 90.2**	**43.0 / 71.1 / 90.0**
S	SSM [64]	54.5 / 81.6 / 96.7	10.0 / 23.7 / 45.4	14.5 / 33.2 / 47.5
	VLM [78]	7.9 / 30.0 / 85.9	11.9 / 26.0 / 55.0	15.7 / 34.5 / 60.5
	SMC [73]	50.3 / 79.3 / 95.2	6.2 / 18.5 / 44.3	8.0 / 26.4 / 46.4
	DASGIL-FD [25]	8.7 / 30.7 / 81.3	1.6 / 4.8 / 19.9	1.8 / 4.3 / 21.6
	ToDayGAN [1, 16]	52.2 / 80.1 / 95.9	16.4 / 43.2 / 73.3	24.1 / 50.5 / 74.1
	LBR [79]	56.7 / **81.7** / **98.2**	24.9 / 62.3 / 86.1	**47.5 / 73.4 / 90.0**
	Ours	**56.9** / 81.6 / **97.4**	**27.6 / 66.2 / 90.2**	43.0 / 71.1 / **90.0**
L	SIFT [35]	53.5 / 77.6 / 92.6	7.8 / 13.9 / 22.1	9.5 / 14.5 / 17.0
	SPP [14]	56.5 / 81.5 / 97.1	16.9 / 41.6 / 71.5	22.0 / 45.0 / 68.0
	D2Net [16]	54.5 / 80.0 / 95.3	18.0 / 39.7 / 53.9	22.7 / 40.5 / 56.1
	R2D2 [51]	**57.4 / 81.9** / 97.9	18.3 / 43.4 / 67.8	29.1 / 50.2 / 68.2
	CAPS [77]	56.0 / 81.5 / 96.5	21.9 / 54.3 / 86.8	27.0 / 58.9 / 85.9
	ASLFeat [37]	57.1 / **81.9 / 98.4**	23.5 / 55.9 / 80.1	41.1 / 66.8 / 86.1
	Ours	56.9 / 81.6 / 97.4	**27.6 / 66.2 / 90.2**	**43.0 / 71.1 / 90.0**
M	SPP+SPG [14, 55]	**56.9** / 81.7 / **98.1**	24.2 / 62.6 / 87.4	42.3 / 69.3 / 90.2
	Pixloc [56]	**56.9 / 82.0 / 98.1**	24.2 / 62.8 / 88.4	**45.5 / 72.5** / 90.7
	AHM [18]	45.7 / 78.0 / 95.1	16.2 / 55.3 / **93.6**	28.4 / 68.4 / **95.5**
	Ours	**56.9** / 81.6 / 97.4	**27.6 / 66.2** / 90.2	43.0 / 71.1 / 90.0

Table 4. **Results on RobotCar-Seasons dataset [39,59].** The best and second best results are highlighted with **bold** and red fonts.

excellent accuracy by selecting keypoints from buildings and performing instance-wise matching. Our method gives close results to LBR on day images but better performance on most night images, because our model does not require explicit semantic labels and is less fragile to segmentation errors especially for night images. LBR performs better than ours on night-train images in Table 4 because it is trained on augmented night rainy images, while our model is trained only on generated night images as R2D2.

Comparison with learned features (L). Benefiting from training with massive data, learned features such as R2D2 [51], ASLFeat [37] and PoSFeat [32], outperform SIFT [3, 35]. As they extract keypoints indiscriminately, they are still more sensitive to appearance changes especially on night images than semantic-aware methods [79], as shown in Table 2 and 4. Our model extracts semantic-aware features directly, so it gives higher accuracy.

Comparison with advanced matchers (M). In Table 2, 3 and 4, we also show the results of previous advanced matchers. We find that our approach outperforms the recent efficient variations of SPG [55] (*e.g.* SGMNet [8], ClusterGNN [65]) and gives competitive results to SPG, which achieves the best accuracy and is also the slowest method. Note that our model only uses the simple MNN for matching. We provide a detailed analysis of time and memory usage in Sec. 4.2 and argue that our method provides a good trade off between running time and accuracy especially on devices with limited computing resources.

Robustness to the number of keypoints. We observe that most previous methods [16, 32, 37, 50, 51] extract keypoints with the number ranging from 10k (R2D2, ASLFeat) to 40k (PosFeat) for evaluation. Although increasing the number improves the accuracy, it causes the time cost as well, which should be taken into consideration especially on devices with limited computing resources. In this experiment, we test the ability of previous and our methods of extracting fewer but more robust features by reducing the number of keypoints from 4k to 1k. Note that results of [14, 51, 55, 79] are from LBR [79] and results of ASLFeat [37] are obtained by running official source code.

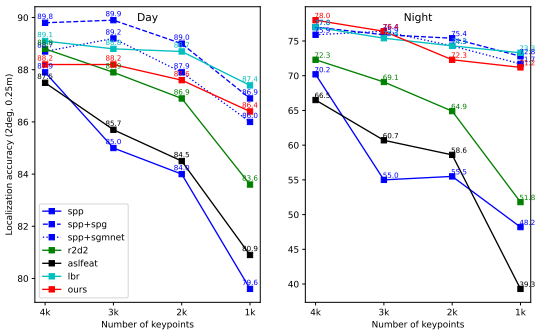

Figure 6. **Influence of the number of keypoints.** We report results of different number (from 4k to 1k) of keypoints on Aachen_v1.1 [59, 60] at error threshold of $(2°, 0.25m)$.

Fig. 6 shows that as the number of keypoints decreases, all previous features [14, 37, 51] undergo dramatic accuracy loss especially for night images. SPP+SPG and LBR perform more robust because of the global context or semantics. With implicitly embedded semantics, our feature outperforms R2D2, ASLFeat, and SPP especially on night images and gives competitive results to SPP+SPG and LBR.

Qualitative comparison. Fig. 7 shows the detection and matching results of query images under conditions of large illumination and season changes. Compared with prior features [14, 37, 51], which prefer keypoints from areas with rich textures, our method favors keypoints from objects robust for long-term localization (*e.g.* buildings). When insufficient keypoints can be found from stale regions, our model also uses keypoints from Short-term objects *e.g.* trees from compensation but assigns them with lower reliability. Besides, our feature gives more inliers for query images with large occlusions of trees and severe illumination changes.

4.2. Running time analysis

Table 5 demonstrates the test time of previous features [14, 37, 51, 79], matchers [8, 55], and our method. Our method (33.2ms) is faster than R2D2 (72.4ms) [51] and slower than SPP (13.1ms) [14], but has higher accuracy. Besides, our method is faster than LBR (9.2+30.1ms) [79], which uses explicit instances to filter keypoints and advanced matchers including SPP+SPG (13.1+52.2/146.5ms) [14,55] and SPP+SGM (13.1+33.2/97.6ms) [8, 14]. As the matching method is ex-

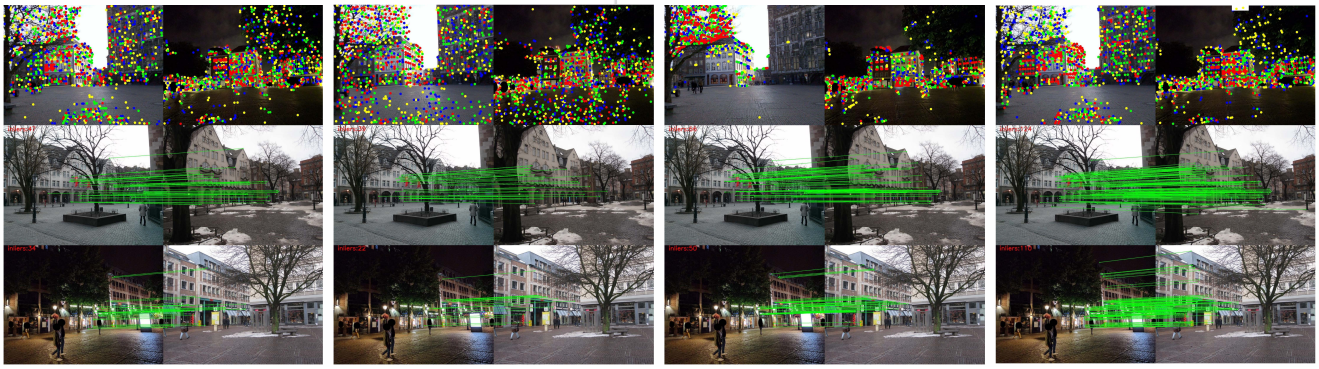

Figure 7. **Qualitative comparison of detection and matching.** Left→right: SPP [14], R2D2 [51], ASLFeat [37] and our method. Our model favors keypoints on stable areas (reliability high→low: 1-250, 251-500, 501-750, 751-1000) and gives more inliers.

Model	Input size	Running time (ms)
SPP [14]	1024×1024	13.1
R2D2 [51]	1024×1024	72.4
ASLFeat* [37]	1024×1024	112.3
LBR (feature) [79]	1024×1024	30.1
LBR (segmentation) [79]	256×256	9.2
SPG [55]	2k×2k, 4k×4k	52.2, 146.5
SGMNet [8]	2k×2k, 4k×4k	85.5, 97.6
Ours	1024×1024	33.2

Table 5. **Running time.** We report test time of prior features [14, 37, 51], matchers [8, 55] and our method (*in TensorFlow).

SD	SS	SF	Day $(2^\circ, 0.25m)/(5^\circ, 0.5m)/(10^\circ, 5m)$	Night $(2^\circ, 0.25m)/(5^\circ, 0.5m)/(10^\circ, 5m)$
✗	✗	✗	85.4 / 93.6 / 97.9	71.2 / 84.3 / 98.4
✓	✗	✗	87.3 / 94.3 / 97.8	72.8 / 88.5 / **99.5**
✗	✓(triplet)	✗	87.9 / 95.1 / **98.9**	73.8 / 86.9 / 99.0
✗	✓(ranking)	✗	87.9 / 95.3 / 98.7	74.9 / 89.5 / 99.0
✓	✓	✗	**88.2** / 95.5 / 98.8	75.9 / 89.0 / **99.5**
✓	✓	✓	**88.2** / **96.0** / 98.7	**78.0** / **92.1** / **99.5**

Table 6. **Ablation study.** We test the efficacy of semantic detection (SD), semantic description (SS), and semantic feature consistency (SF) losses. The best results are highlighted.

tensively used for each image pair in mapping and localization processes, SPG/SGMNet are about 18.3/3.3 times slower than NN on Aachen dataset [59, 60] in the mapping process as discussed in [8]. Therefore, our approach could be a good trade off between accuracy and efficiency.

4.3. Ablation study

In Table 6, we verify the effectiveness of all components in our network by progressively adding the semantic detection (SD), description (SS), and feature consistency (SF) losses. We also compare results of SS loss with triplet and ranking as intra-class loss. Our baseline is trained with detection scores of SPP [14] as detector supervision and ap [23] loss for descriptor learning. After adding SD loss, the model performs better especially for night images. The accuracy is further improved by introducing SS loss because it augments the discriminative ability of descriptors with semantics. Compared to triplet loss with carefully tuned margin, ranking loss improves more accuracy as objects' inner diversity can be better retained by optimizing ranks of samples. SF loss enhances the network's ability of embedding semantic information, leading to further improvements.

5. Limitations

The first limitation is the hand-defined stability values. A learned stability map from training data could be more robust and further improve the localization accuracy. Besides, semantic labels used in the paper are from ADE20K [83] and the number of these labels is limited. Fine grained semantic labels [31] from automatic segmentation might be more reliable in real applications. Moreover, this work focuses mainly on outdoor localization and may not work very well in indoor scenarios due to the significant differences of object classes. Better performance for indoor scenes can be achieved by retraining the model with redefined stability map for indoor objects.

6. Conclusions

In this paper, we implicitly incorporate semantic information into the feature detection and description processes, enabling the model to extract globally reliable features from a single network end-to-end. Specifically, we leverage outputs of an off-the-shelf semantic segmentation network as guidance and adopt a combination of semantic- and feature-aware guidance strategies to enhance the ability of embedding semantic information at training time. Experiments on large-scale visual localization datasets demonstrate that our method outperforms prior local features and gives competitive performance to advanced matchers but has higher efficiency. We argue that our approach could be a good trade-off between accuracy and efficiency.

Supplementary Material

In the supplementary material, we first show localization results on Extended CMU-Seasons dataset in Table 7. Next, we give more qualitative comparison of previous and our methods on feature extraction and matching in Sec. A. Then we provide a detailed ablation study of our approach in Sec. B. Finally, we introduce the process of generating global stability and the architecture of our network in Sec. C and Sec. D, respectively.

A. Analysis of Feature Detection and Matching

In this section, we show more qualitative results of keypoints detection and matching in comparison with previous popular local features including SuperPoint [14], D2Net [16], R2D2 [51], and ASLFeat [37].

A.1. Feature detection

For each method, we detect top 1k keypoints with highest scores from the query images of Aachen_v1.1 dataset [58, 60] at the original resolution and visualize these keypoints with different colors according to their scores (high→low: 1-250, 251-500, 501-750, 751-1000). As shown in Fig. 8, we can see that:

- D2Net [16] and ASLFeat [37] favor regions with rich textures especially objects such as tress and pedestrians, partially because D2Net and ASLFeat adopt the similar detection strategy: spatial locations with high values of the high-level features. As a result, they detect many keypoints from objects *e.g.* sky, tree, car, pedestrian, which are not useful for long-term localization.

- R2D2 [51] detects keypoints almost uniformly from the whole image due to its maximization of responses in a fixed sliding window with size of 16×16. Therefore, R2D2 [51] also detects a large number of keypoints from unstable objects.

- SuperPoint [14] is a good corner detector. As corners also exist in objects *e.g.* sky, tree, car, pedestrian, SuperPoint [14] detects many keypoints from the aforementioned unstable objects.

- Our detector is partially supersized by results of SuperPoint, so it favors corners as well. Because we rerank the corners with the *stability* of semantic labels, our method prefers to detect keypoints from stable objects *e.g.* building, with more red keypoints from buildings. Although we can see keypoints from unstable objects, their scores are relatively smaller (with blue or yellow colors).

- The distribution of keypoints detected by prior methods and our model indicates that without explicit semantic labels, previous approaches don't perform well of selecting globally reliable keypoints although they are trained to detect keypoints which have strong discriminative ability.

A.2. Feature matching

We detect 4k keypoints for SuperPoint [14], R2D2 [51], ASLFeat [37], and our method and visualize the inliers between query and reference images with illumination changes, season variations, dynamic objects in the Aachen_v1.1 dataset [58, 60]. From Fig. 9, we can see that:

- For image pairs with small illumination and season changes, almost all methods could give many inliers.

- For image pairs with season changes or occlusions from trees or dynamic objects *e.g.* car, SuperPoint, R2D2, and ASLFeat give fewer inliers than our model.

- For extremely challenging image pairs with illumination changes, season variations, and high occlusions of trees, almost all prior approaches fail to give enough inliers, resulting in the failure of localization. However, our method is still able to find enough inliers from robust regions. We analyze the reasons of improvements in Sec. B.

B. Ablation Study of Feature Detection and Matching

In this section, we verify the efficacy of the proposed semantic-aware detection (SD), semantic-aware description (SS), and semantic-consistency (SF) losses by visualizing the detection and matching results. The base model is trained with results of SuperPoint [14] as supervision and a general ap loss [23] for descriptor learning as R2D2 [51]. Our full model comprises SD, SS, and SF three components.

B.1. Ablation study of detection

As in Sec. A.1, we visualize 1k keypoints with the highest scores and show them with different colors according to their scores (high→low: 1-250, 251-500, 501-750, 751-1000). As shown in Fig. 8, we can see the effectiveness of SD, SS, and SF in detail:

- Our base model performs closely to SuperPoint [14] (as shown in Fig. 8) with high response to corners as the detector is partially supervised with results of SuperPoint [14]. Meanwhile, the base model is also sensitive to unstable objects *e.g.* sky, tree, pedestrian, and car.

Group	Method	urban	suburban	park	overcast	sunny	foliage	mixed foliage	no foliage	low sun	cloudy	snow
C	SIFT [35]	56.9/63.9/70.2	37.8/45.3/55.4	20.0/24.4/31.7	36.1/42.6/50.5	30.9/36.3/43.6	32.7/38.2/45.7	35.5/42.2/51.4	59.5/67.5/74.7	43.7/50.8/59.2	43.0/49.6/58.3	46.1/54.2/63.1
	AS [57]	81.0/87.3/92.4	62.6/70.9/81.0	45.5/51.6/62.0	64.1/70.8/78.6	55.2/62.3/71.3	58.8/65.3/73.9	59.2/67.5/77.4	83.3/88.9/94.6	65.8/73.4/82.8	71.6/77.6/84.2	73.0/81.0/90.5
	CSL [69]	71.2/74.6/78.7	57.8/61.7/67.5	34.5/37.0/42.2	52.2/55.4/60.3	43.3/46.6/51.9	47.0/50.2/55.3	52.4/56.1/62.0	80.3/83.2/86.6	61.7/65.3/70.7	63.3/66.3/70.5	69.9/73.7/78.7
S	VLM [78]	17.3/42.5/89.0	5.8/19.4/76.1	6.6/23.1/73.0	11.5/30.8/80.8	9.7/27.1/76.1	9.5/26.7/77.4	10.3/28.4/79.0	9.4/30.3/84.6	9.3/27.6/79.2	9.4/28.0/83.7	7.6/27.6/75.9
	SSM [64]	88.8/93.6/96.3	78.0/83.8/89.2	63.6/70.3/77.3	79.1/84.9/89.7	69.2/75.4/81.3	73.4/79.1/84.2	75.1/81.8/87.9	90.9/94.5/97.1	78.5/84.5/90.1	86.4/90.5/92.9	84.1/89.8/94.6
L	SPP [14]	89.5/94.2/97.9	76.5/82.7/92.7	57.4/ 64.4/80.4	77.1/82.8/91.8	65.1/72.3/86.8	69.2/ 75.5/88.3	75.2/81.7/90.8	88.7/92.8/96.4	78.0/83.9/91.8	83.4/87.7/94.0	80.7/86.6/93.2
	D2Net(MS) [16]	82.6/94.8/98.4	75.9/86.8/93.8	66.6/82.6/ 88.6	76.3/89.0/94.1	68.2/83.8/92.0	70.4/85.2/92.5	75.8/88.6/93.8	86.2/94.4/96.7	78.6/89.9/94.4	79.1/90.7/95.1	82.0/91.1/93.8
	R2D2 [51]	89.7/96.6/98.3	76.1/83.8/89.0	64.4/72.1/76.5	79.9/87.0/90.6	70.3/78.3/83.2	74.1/81.2/85.6	75.7/84.1/87.9	86.6/93.3/95.3	77.8/85.7/89.3	84.1/90.0/92.5	79.8/87.6/91.1
M	PixLoc [84]	92.8/95.1/98.5	91.9/93.4/95.8	84.0 /85.8/90.9	90.3/92.2/96.2	85.3/88.8/94.0	87.1/89.9/94.7	90.5/91.9/95.1	95.1/95.7/96.8	91.2/92.3/94.8	93.9/94.8/97.4	91.6/92.3/94.0
	AHM [18]	65.7/82.7/91.0	66.5/82.6/92.9	54.3/71.6/84.1	62.8/78.8/89.4	56.6/74.5/87.2	58.5/75.7/87.8	62.9/79.6/89.4	72.0/87.7/94.5	64.0/81.0/90.2	69.4/84.4/92.8	61.7/80.6/90.3
	SPP+SPG [14, 55]	95.5/98.6/99.3	90.9/94.2/97.1	85.7/89.0/91.6	92.3/95.3/96.9	86.1/91.3/94.6	88.3/92.5/95.3	91.6/94.5/96.2	95.4/97.1/98.3	91.8/94.4/96.3	95.2/97.0/98.0	92.3/94.6/96.6
	Ours	95.0/97.5/98.6	90.5/92.7/95.3	86.4/89.1/91.2	92.1/94.0/95.8	86.3/90.3/93.4	87.9/91.0/93.9	91.9/94.0/95.5	95.3/96.6/97.6	92.4/94.4/95.8	93.3/94.7/96.3	92.9/94.6/96.0

Table 7. **Localization accuracy on the Extended CMU-Seasons dataset [59].** Results at error thresholds of $(0.25m, 2^\circ), (0.5m, 5^\circ), (5m, 10^\circ)$ are reported.

- The SD loss (W/ SD) is the key to rerank the keypoints. With SD loss, keypoints from unstable objects *e.g.* sky, car, pedestrian are suppressed. Keypoints from trees have lower score (with color of blue or yellow) and keypoints from stable objects *e.g.* building are favored (with color of red).
- The SS loss doesn't contribute to the detection process, so it shows the similar results as the base model, which again indicates that the importance of explicit semantic labels to detection as discussed in Sec. A.1,
- The full model with SF incorporated performs better than the model W/ SD, as it further enhances the ability of our model in learning semantic-aware features.

B.2. Ablation study of feature matching

We additionally visualize the effectiveness of SD, SS, SF losses in feature matching. From Fig. 12, we can see that:

- Benefiting from the corner detector and ap loss, the base model is already able to give promising performance in comparison with previous methods [14, 37, 51].
- The SD loss (W/ SD) marginally improves the matches possibly because those reranked keypoints from stable objects don't have strong discriminative ability by purely adopting ap loss over all keypoints.
- The SS loss (W/ SS) effectively solves the limitation of SD loss, as it augments the discriminative ability of descriptors with semantics.
- The full model gives the best performance because it combines the advantages of SD, SS, and SF losses.

C. Global stability map generation

During the training process, we utilize UperNet [11] with ConvNet [34] as encoder trained on ADE20k [83] dataset to provide semantic segmentation labels and high-level features for semantic-wise and feature-wise guidance, respectively. There are 150 labels in total which are categorized into 4 groups as shown in Table 8. Since large-scale localization happens mainly in outdoor environments, only several objects such as sky, water, pedestrian, car, tree, plant, and building are frequently used.

Category	Semantics
Volatile	sky, mountain, curtain, water, sea, mirror, rug, field, bathtub, stand, sand, sink, river, hill, bench,light,dirt,land,fountain, swimming pool, waterfall, lake
Dynamic	person, automobile, boat, truck
Short-term	tree, grass, plant, flower, palm, airplane, van, ship, minibike, bike, shower
Long-term	wall, building, floor, ceiling, road, bed, window, cabinet, sidewalk, ground, door, chair, painting, sofa, shelf, house, armchair, seat, fence, rock, wardrobe, lamp, rail, cushion, box, pillar, signboard, chest, counter, skyscraper, fireplace, grandstand, path, stairs, runway, case, table, pillow, screen, stairway, bridge, bookcase, toilet, book, countertop, stove, kitchen, computer, swivel, bar, arcade, hovel, tower, chandelier, sunshade, streetlight, booth, television, clothes, pole, bannister, escalator, ottoman, bottle, buffet, poster, stage, conveyer, canopy, washer, toy, stool, cask, basket, tent, bag, cradle, oven, ball, food, step, tank, trade name, pot, dishwasher, screen, blanket, sculpture, hood, sconce, vae, traffic light, tray, dustbin, plate, monitor, bulletin, glass, clock, flag

Table 8. **Stability map of different labels**. All semantic labels are categorized into four groups denoted as *Volatile*, *Dynamic*, *Short-term*, and *Long-term* according to their reliability in the visual localization task.

D. Network

Alike to SuperPoint [14], we adopt 8 times downsampling to reduce the resolution of high-dimension features, making the model efficient at test time. To increase the representation ability of our model, we introduce 3 ResBlocks [24]. Details of the network are shown in Fig. 10.

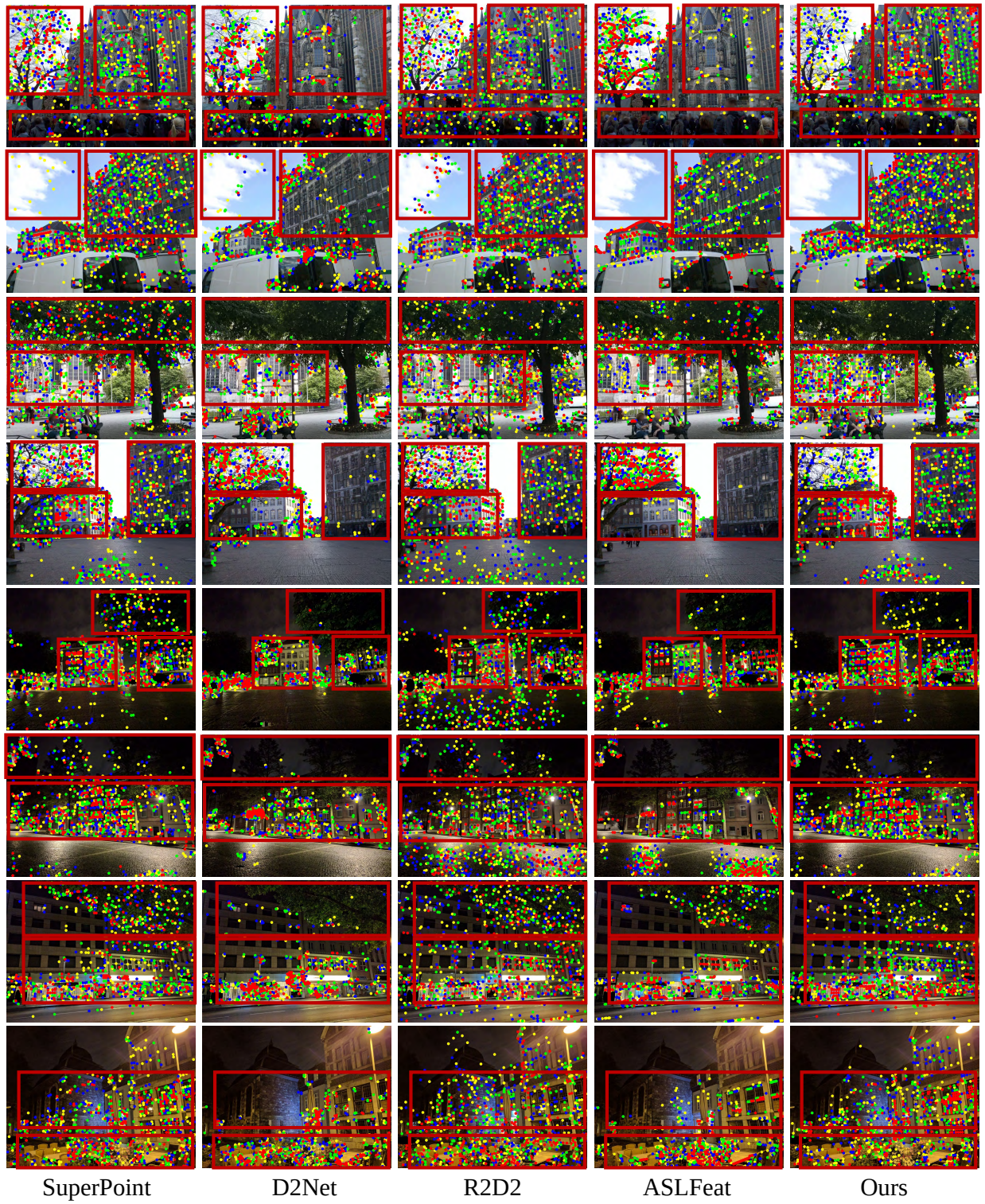

Figure 8. **Comparison of feature detection.** We show top 1k keypoints with highest scores (high→low: 1-250, 251-500, 501-750, 751-1000) of prior SOTA local features including SuperPoint [14], D2Net [16], R2D2 [51], and ASLFeat [37]. They are more sensitive to regions with rich textures even those from objects *e.g.* sky, tree, pedestrian, car, which are unstable for long-term localization. By introducing the semantics for reranking keypoints, our model prefers keypoints from stable objects *e.g.* building.

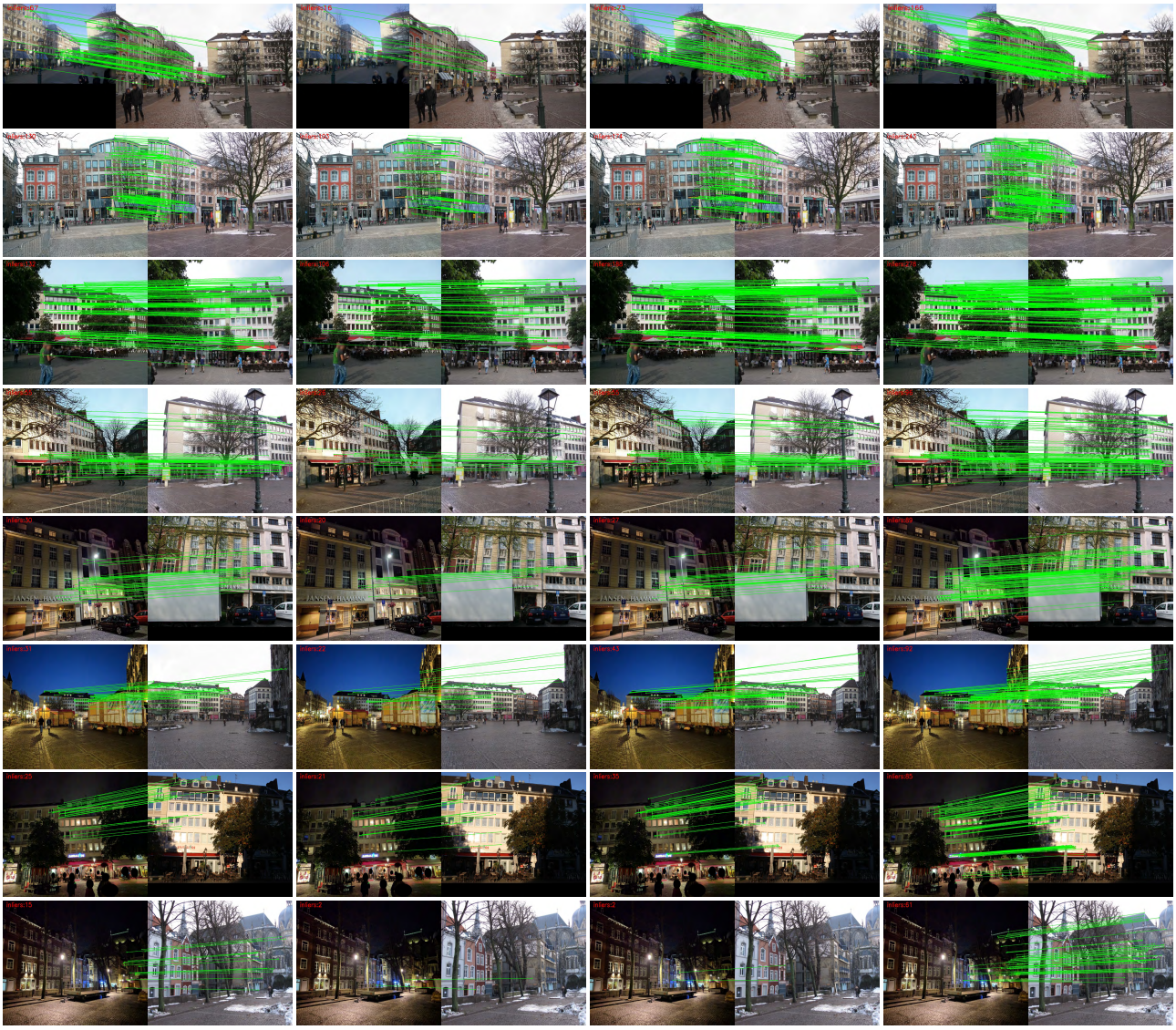

Figure 9. **Comparison of feature matching.** We show inliers between query and reference images from the Aachen_v1.1 [58, 60] dataset under challenges of illumination changes, season variations, and dynamic objects. Results of SuperPoint [14], R2D2 [51], ASLFeat [37], and our model are visualized. Compared with prior methods, our model is able to produce more inliers even under extremely challenging conditions when others fail to give enough inliers to guarantee the success of localization.

References

[1] Asha Anoosheh, Torsten Sattler, Radu Timofte, Marc Pollefeys, and Luc Van Gool. Night-to-day image translation for retrieval-based localization. In *ICRA*, 2019. 3, 6, 7

[2] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *CVPR*, 2016. 2, 6

[3] Relja Arandjelović and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, 2012. 1, 2, 7

[4] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *TPAMI*, 2017. 5

[5] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. DSAC-differentiable RANSAC for camera localization. In *CVPR*, 2017. 2

[6] Eric Brachmann and Carsten Rother. Learning less is more-6d camera localization via 3d surface regression. In *CVPR*, 2018. 2

[7] Ignas Budvytis, Marvin Teichmann, Tomas Vojir, and Roberto Cipolla. Large scale joint semantic re-localisation

and scene understanding via globally unique instance coordinate regression. In *BMVC*, 2019. 2, 3

[8] Hongkai Chen, Zixin Luo, Jiahui Zhang, Lei Zhou, Xuyang Bai, Zeyu Hu, Chiew-Lan Tai, and Long Quan. Learning to match features with seeded graph matching network. In *ICCV*, 2021. 1, 2, 3, 6, 7, 8

[9] Hongkai Chen, Zixin Luo, Lei Zhou, Yurun Tian, Mingmin Zhen, Tian Fang, David Mckinnon, Yanghai Tsin, and Long Quan. ASpanFormer: Detector-Free Image Matching with Adaptive Span Transformer. In *ECCV*, 2022. 1, 3, 6

[10] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 2

[11] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 2, 4, 10

[12] Wentao Cheng, Weisi Lin, Kan Chen, and Xinfeng Zhang. Cascaded parallel filtering for memory-efficient image-based localization. In *ICCV*, 2019. 2, 6, 7

[13] Titus Cieslewski, Konstantinos G Derpanis, and Davide Scaramuzza. SIPs: Succinct interest points from unsupervised inlierness probability learning. In *3DV*, 2019. 3

[14] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPRW*, 2018. 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 16

[15] Mihai Dusmanu, Ondrej Miksik, Johannes L Schonberger, and Marc Pollefeys. Cross-Descriptor Visual Localization and Mapping. In *ICCV*, 2021. 3

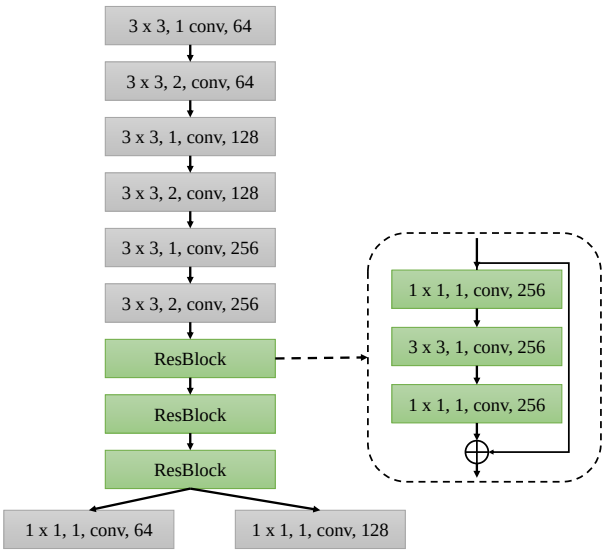

Figure 10. **Architecture of the network.** We adopt 6 Convolution layers with kernel size of 3×3 to generate high-level features with $8\times$ downsampling (implementation by using stride of 2). Then 3 ResBlocks [24] are followed to further enhance the ability of the model.

[16] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-Net: A trainable CNN for joint description and detection of local features. In *CVPR*, 2019. 1, 2, 3, 6, 7, 9, 10, 11

[17] Patrick Ebel, Anastasiia Mishchuk, Kwang Moo Yi, Pascal Fua, and Eduard Trulls. Beyond cartesian representations for local descriptors. In *ICCV*, 2019. 3

[18] Hugo Germain, Guillaume Bourmaud, and Vincent Lepetit. Sparse-to-dense hypercolumn matching for long-term visual localization. In *3DV*, 2019. 1, 3, 6, 7, 10

[19] Hugo Germain, Guillaume Bourmaud, and Vincent Lepetit. S2DNet: Learning accurate correspondences for sparse-to-dense feature matching. In *ECCV*, 2022. 1, 6

[20] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. In *NIPS*, 2014. 3

[21] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *IJCV*, 2021. 5

[22] Wilfried Hartmann, Michal Havlena, and Konrad Schindler. Predicting matchability. In *CVPR*, 2014. 3

[23] Kun He, Yan Lu, and Stan Sclaroff. Local descriptors optimized for average precision. In *CVPR*, 2018. 2, 3, 5, 8, 9

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5, 10, 13

[25] Hanjiang Hu, Zhijian Qiao, Ming Cheng, Zhe Liu, and Hesheng Wang. DASGIL: Domain adaptation for semantic and geometric-aware image-based localization. *TIP*, 2020. 3, 6, 7

[26] Zhaoyang Huang, Han Zhou, Yijin Li, Bangbang Yang, Yan Xu, Xiaowei Zhou, Hujun Bao, Guofeng Zhang, and Hongsheng Li. VS-Net: Voting with segmentation for visual localization. In *CVPR*, 2021. 2, 3

[27] Hyo Jin Kim, Enrique Dunn, and Jan-Michael Frahm. Learned contextual feature reweighting for image geo-localization. In *CVPR*, 2017. 2

[28] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *ICCV*, 2015. 2

[29] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[30] Nikolay Kobyshev, Hayko Riemenschneider, and Luc Van Gool. Matching features correctly through semantic understanding. In *3DV*, 2014. 3

[31] Mans Larsson, Erik Stenborg, Carl Toft, Lars Hammarstrand, Torsten Sattler, and Fredrik Kahl. Fine-grained segmentation networks: Self-supervised segmentation for improved long-term visual localization. In *ICCV*, 2019. 2, 3, 8

[32] Kunhong Li, Longguang Wang, Li Liu, Qing Ran, Kai Xu, and Yulan Guo. Decoupling Makes Weakly Supervised Local Feature Better. In *CVPR*, 2022. 2, 3, 6, 7

[33] Xinghui Li, Kai Han, Shuda Li, and Victor Prisacariu. Dual-resolution correspondence networks. In *NeurIPS*, 2020. 1, 3, 6

[34] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022. 2, 4, 5, 10

[35] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 1, 2, 3, 6, 7, 10

[36] Zixin Luo, Tianwei Shen, Lei Zhou, Jiahui Zhang, Yao Yao, Shiwei Li, Tian Fang, and Long Quan. Contextdesc: Local descriptor augmentation with cross-modality context. In *CVPR*, 2019. 3

[37] Zixin Luo, Lei Zhou, Xuyang Bai, Hongkai Chen, Jiahui Zhang, Yao Yao, Shiwei Li, Tian Fang, and Long Quan. ASLFeat: Learning local features of accurate shape and localization. In *CVPR*, 2020. 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12

[38] Jiayi Ma, Xingyu Jiang, Aoxiang Fan, Junjun Jiang, and Junchi Yan. Image matching from handcrafted to deep features: A survey. *IJCV*, 2021. 2

[39] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The Oxford RobotCar dataset. *IJRR*, 2017. 6, 7

[40] Iaroslav Melekhov, Aleksei Tiulpin, Torsten Sattler, Marc Pollefeys, Esa Rahtu, and Juho Kannala. DGCNet: Dense geometric correspondence network. In *WACV*, 2019. 1, 6

[41] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *NeurIPS*, 2017. 2, 3, 4

[42] Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Repeatability is not enough: Learning affine regions via discriminability. In *ECCV*, 2018. 3

[43] Arun Mukundan, Giorgos Tolias, and Ondrej Chum. Explicit spatial encoding for deep local descriptors. In *CVPR*, 2019. 3

[44] Tayyab Naseer, Gabriel L Oliveira, Thomas Brox, and Wolfram Burgard. Semantics-aware visual localization under challenging perceptual conditions. In *ICRA*, 2017. 3

[45] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. LF-Net: learning local features from images. In *NeurIPS*, 2018. 1

[46] Alexandra I Papadaki and Ronny Hansch. Match or no match: Keypoint filtering based on matching probability. In *CVPRW*, 2020. 3

[47] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016. 1

[48] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 6

[49] Rémi Pautrat, Viktor Larsson, Martin R Oswald, and Marc Pollefeys. Online invariance selection for local feature descriptors. In *ECCV*, 2020. 3, 6

[50] Jérome Revaud, Vincent Leroy, Philippe Weinzaepfel, and Boris Chidlovskii. PUMP: Pyramidal and Uniqueness Matching Priors for Unsupervised Learning of Local Descriptors. In *CVPR*, 2022. 3, 4, 6, 7

[51] Jerome Revaud, Philippe Weinzaepfel, César Roberto de Souza, and Martin Humenberger. R2D2: Repeatable and reliable detector and descriptor. In *NeurIPS*, 2019. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12

[52] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. Efficient neighbourhood consensus networks via submanifold sparse convolutions. In *ECCV*, 2020. 3, 6

[53] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *ICCV*, 2011. 1, 2

[54] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From Coarse to Fine: Robust Hierarchical Localization at Large Scale. In *CVPR*, 2019. 1, 2, 3, 6

[55] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 1, 2, 3, 6, 7, 8, 10

[56] Paul-Edouard Sarlin, Ajaykumar Unagar, Mans Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, et al. Back to the feature: learning robust camera localization from pixels to pose. In *CVPR*, 2021. 6, 7

[57] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized for large-scale image-based localization. *TPAMI*, 2016. 1, 2, 6, 7, 10

[58] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. Benchmarking 6dof outdoor visual localization in changing conditions. In *CVPR*, 2018. 2, 9, 12, 17

[59] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. Benchmarking 6dof outdoor visual localization in changing conditions. In *CVPR*, 2018. 6, 7, 8, 10

[60] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image retrieval for image-based localization revisited. In *BMVC*, 2012. 6, 7, 8, 9, 12, 17

[61] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe. Understanding the limitations of cnn-based absolute camera pose regression. In *CVPR*, 2019. 2

[62] Johannes L Schönberger, Marc Pollefeys, Andreas Geiger, and Torsten Sattler. Semantic visual localization. In *CVPR*, 2018. 3

[63] Johannes L Schönberger, Marc Pollefeys, Andreas Geiger, and Torsten Sattler. Semantic visual localization. In *CVPR*, 2018. 3

[64] Tianxin Shi, Shuhan Shen, Xiang Gao, and Lingjie Zhu. Visual localization using sparse semantic 3D map. In *ICIP*, 2019. 1, 3, 4, 6, 7, 10

[65] Yan Shi, Jun-Xiong Cai, Yoli Shavit, Tai-Jiang Mu, Wensen Feng, and Kai Zhang. ClusterGNN: Cluster-based Coarse-to-Fine Graph Neural Network for Efficient Feature Matching. In *CVPR*, 2022. 1, 2, 3, 6, 7

[66] Erik Stenborg, Carl Toft, and Lars Hammarstrand. Long-term visual localization using semantically segmented images. In *ICRA*, 2018. 3

[67] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. In *CVPR*, 2021. 1, 3, 6

[68] Suwichaya Suwanwimolkul, Satoshi Komorita, and Kazuyuki Tasaka. Learning of low-level feature keypoints for accurate and robust detection. In *WACV*, 2021. 3, 6

[69] Linus Svärm, Olof Enqvist, Fredrik Kahl, and Magnus Oskarsson. City-scale localization for cameras with known vertical direction. *TPAMI*, 2016. 1, 2, 6, 7, 10

[70] Yurun Tian, Vassileios Balntas, Tony Ng, Axel Barroso-Laguna, Yiannis Demiris, and Krystian Mikolajczyk. D2D: Keypoint extraction with describe to detect approach. In *ACCV*, 2020. 3, 6

[71] Yurun Tian, Bin Fan, and Fuchao Wu. L2-net: Deep learning of discriminative patch descriptor in euclidean space. In *CVPR*, 2017. 3

[72] Yurun Tian, Xin Yu, Bin Fan, Fuchao Wu, Huub Heijnen, and Vassileios Balntas. SOSNet: Second order similarity regularization for local descriptor learning. In *CVPR*, 2019. 3, 6

[73] Carl Toft, Erik Stenborg, Lars Hammarstrand, Lucas Brynte, Marc Pollefeys, Torsten Sattler, and Fredrik Kahl. Semantic match consistency for long-term visual localization. In *ECCV*, 2018. 1, 3, 4, 6, 7

[74] Prune Truong, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning accurate dense correspondences and when to trust them. In *CVPR*, 2021. 1, 6

[75] Prune Truong, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning accurate dense correspondences and when to trust them. In *CVPR*, 2021. 3

[76] Michał J Tyszkiewicz, Pascal Fua, and Eduard Trulls. DISK: Learning local features with policy gradient. In *NeurIPS*, 2020. 1, 3

[77] Qianqian Wang, Xiaowei Zhou, Bharath Hariharan, and Noah Snavely. Learning feature descriptors using camera pose supervision. In *ECCV*, 2020. 3, 4, 6, 7

[78] Zhe Xin, Yinghao Cai, Tao Lu, Xiaoxia Xing, Shaojun Cai, Jixiang Zhang, Yiping Yang, and Yanqing Wang. Localizing discriminative visual landmarks for place recognition. In *ICRA*, 2019. 3, 6, 7, 10

[79] Fei Xue, Ignas Budvytis, Daniel Olmeda Reino, and Roberto Cipolla. Efficient Large-scale Localization by Global Instance Recognition. In *CVPR*, 2022. 1, 2, 3, 4, 6, 7, 8

[80] Fei Xue, Xin Wu, Shaojun Cai, and Junqiu Wang. Learning multi-view camera relocalization with graph neural networks. In *CVPR*, 2020. 2

[81] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *ECCV*, 2016. 1, 3

[82] Long Zhao, Xi Peng, Yuxiao Chen, Mubbasir Kapadia, and Dimitris N Metaxas. Knowledge as priors: Cross-modal knowledge generalization for datasets without superior knowledge. In *CVPR*, 2020. 3

[83] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *IJCV*, 2019. 2, 4, 8, 10

[84] Qunjie Zhou, Torsten Sattler, and Laura Leal-Taixe. Patch2pix: Epipolar-guided pixel-level correspondences. In *CVPR*, 2021. 3, 6, 10

Figure 11. **Ablation study of feature detection.** We show top 1k keypoints with the highest scores (high→low: 1-250, 251-500, 501-750, 751-1000) of our base model, model with SD loss (W/ SD), SS loss (W/ SS) and the full model (with SD, SS, SF). The base model is more sensitive to regions with rich corners as SuperPoint [14]. SD loss effectively mitigates this problem by introducing semantic-aware detection loss. SS loss focus mainly on descriptor learning, so it gives similar results to the base model. The full model additionally introduces SF loss, which further enhances the detection process.

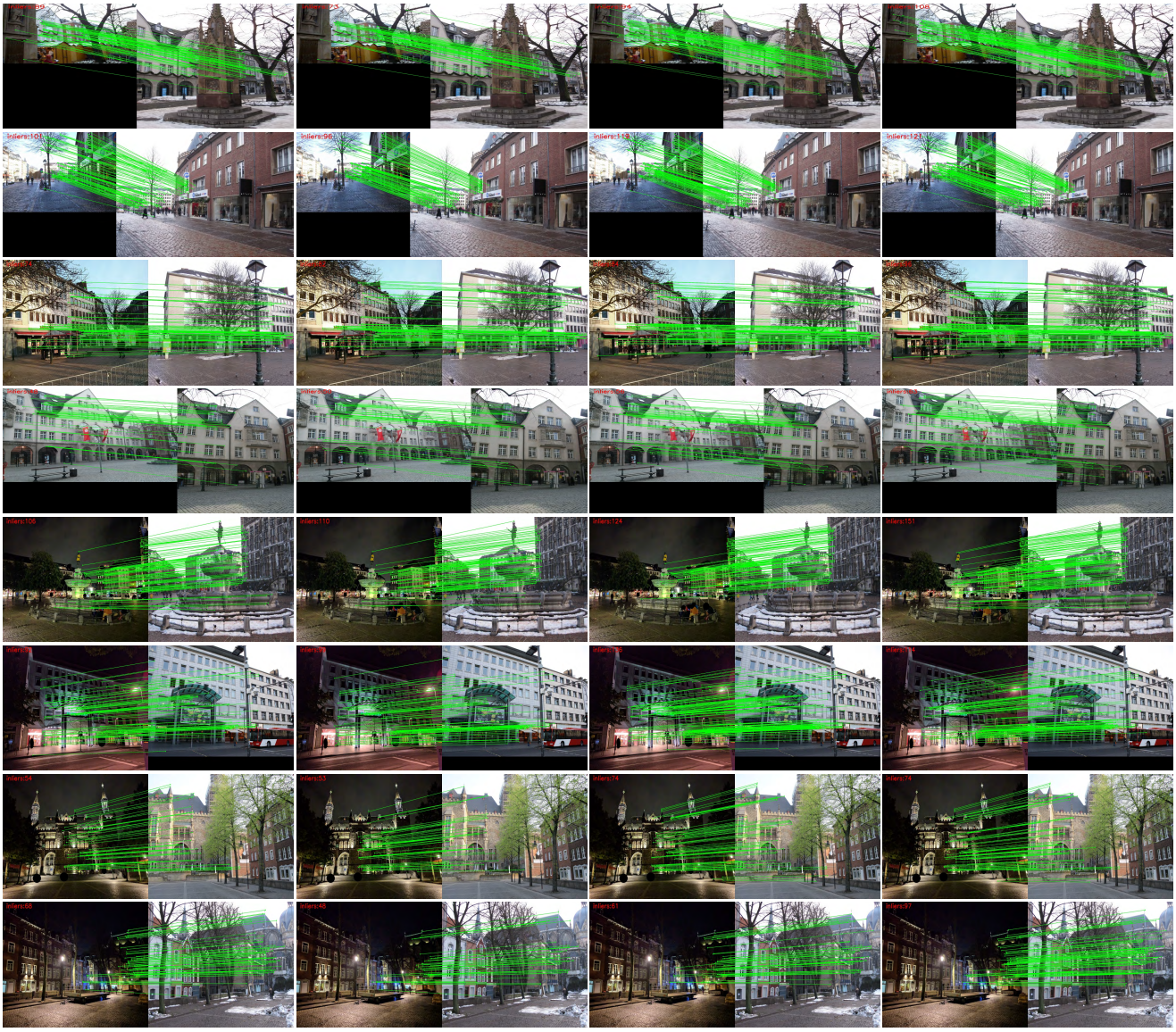

Figure 12. **Ablation study of feature matching.** We show the inliers between query and reference images from the Aachen_v1.1 [58, 60] dataset under challenges of illumination changes, season variations and dynamic objects. Results of the base model, with SD loss (W/ SD), with SS loss (W/ SS) and the full model (with SD, SS, SF) are visualized. SD loss slightly improves the matching as it focus mainly the detection process. SS loss effectively augments the matching accuracy by introducing semantic labels. Results of SS loss are further improved by the full model, which has an additional SF loss to enhance the model's ability of learning semantic-aware features.