DISK: Learning local features with policy gradient

Michał J. Tyszkiewicz[1] **Pascal Fua**[1] **Eduard Trulls**[2]
[1]École Polytechnique Fédérale de Lausanne (EPFL) [2]Google Research, Zurich
michal.tyszkiewicz@epfl.ch pascal.fua@epfl.ch trulls@google.com

Abstract

Local feature frameworks are difficult to learn in an end-to-end fashion, due to the *discreteness* inherent to the selection and matching of sparse keypoints. We introduce DISK (DIScrete Keypoints), a novel method that overcomes these obstacles by leveraging principles from Reinforcement Learning (RL), optimizing end-to-end for a high number of correct feature matches. Our simple yet expressive probabilistic model lets us keep the training and inference regimes close, while maintaining good enough convergence properties to reliably train from scratch. Our features can be extracted very densely while remaining discriminative, challenging commonly held assumptions about what constitutes a good keypoint, as showcased in Fig. 1, and deliver state-of-the-art results on three public benchmarks.

1 Introduction

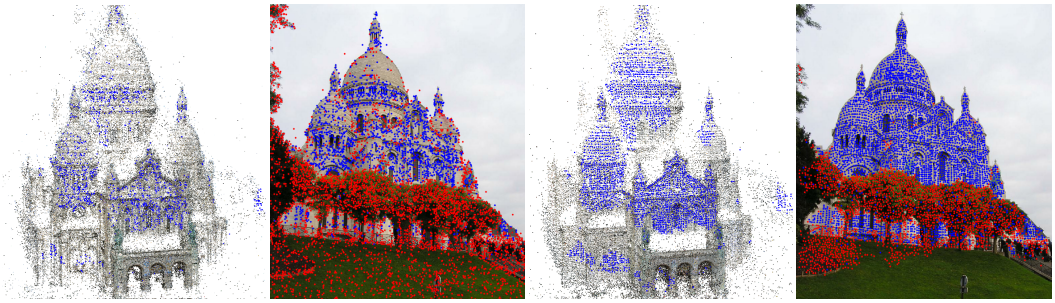
Local features have been a key computer vision technology since the introduction of SIFT [21], enabling applications such as Structure-from-Motion (SfM) [1, 15, 38], SLAM [28], re-localization [24], and many others. While not immune to the deep learning "revolution", 3D reconstruction is one of the last bastions where sparse, hand-crafted solutions remain competitive with or outperform their dense, learned counterparts [39, 36, 17]. This is due to the difficulty of designing end-to-end methods with a differentiable training objective that corresponds well enough with the downstream task.

While patch descriptors can be easily learned on predefined keypoints [41, 42, 26, 43, 13], joint detection and matching is harder to relax in a differentiable manner, due to its computational complexity. Given two images A and B with feature sets F_A and F_B, matching them is $O(|F_A| \cdot |F_B|)$. As each image pixel may become a feature, the problem quickly becomes intractable. Moreover, the "quality" of a given feature depends on the rest, because a feature that is very similar to others is less distinctive, and therefore less useful. This is hard to account for during training.

We address this issue by bridging the gap between training and inference to fully leverage the expressive power of CNNs. Our backbone is a network that takes images as input and outputs keypoint 'heatmaps' and dense descriptors. Discrete keypoints are sampled from the heatmap, and the descriptors at those locations are used to build a distribution over feature matches across images. We then use geometric ground truth to assign positive or negative rewards to each match, and perform gradient descent to maximize the expected reward $\mathbb{E} \sum_{(i,j) \in M_{A \leftrightarrow B}} r(i \leftrightarrow j)$, where $M_{A \leftrightarrow B}$ is the set of matches and r is per-match reward. In effect, this is a policy gradient method [47].

Probabilistic relaxation is powerful for discrete tasks, but its applicability is limited by the fact that the expected reward and its gradients usually cannot be computed exactly. Therefore, noisy Monte Carlo approximations have to be used instead, which harms convergence. We overcome this difficulty by careful modeling that yields analytical expressions for the gradients. As a result we can benefit from the expressiveness of policy gradient, narrowing the gap between training and inference and ultimately outperforming state-of-the-art methods, while still being able to train models from scratch.

Preprint. Under review.



(a) Upright Root-SIFT [21, 3]
179k landmarks, 22.3 observations/landmark

(b) DISK (ours)
190k landmarks, 30.0 observations/landmark

Figure 1: **SIFT vs. DISK in SfM.** We reconstruct "Sacre Coeur" from 1179 images [16] with COLMAP. For Upright Root-SIFT (left) and DISK (right) we show a point cloud and one image with its keypoints. Landmarks, and their respective keypoints, are drawn in **blue**. Keypoints which do not create landmarks are drawn in **red**. Our features can be extracted (and create associations) on seemingly textureless regions where SIFT fails to, producing more landmarks with more observations.

Our contribution therefore is a novel, end-to-end-trainable approach to learning local features that relies on policy gradient. It yields considerably more accurate matches than earlier methods, and this results in better performance on downstream tasks, as illustrated in Fig. 1 and Sec. 4.

2 Related Work

The process of extracting local features usually involves three steps: Finding a keypoint, estimating its orientation, and computing a description vector. In traditional methods such as SIFT [21] or SURF [5], this involves many hand-crafted heuristics. The first wave of local features involving deep networks featured descriptors learned from patches extracted on SIFT keypoints [52, 14, 41] and some of their successors, such as HardNet [26], SOSNet [43], and LogPolarDesc [13], are still state-of-the-art. Other learning-based methods focus on keypoints [45, 37, 19] or orientations [51].

These methods attack a single element of this process. Others have developed end-to-end-trainable pipelines [49, 9, 30, 11, 32] that can optimize the whole process and, hopefully, improve performance. However, they either use inexact approximations to the true objective [9, 32], break differentiability [30] or make big assumptions, such as extrema in descriptor space making good features [11].

Two recent approaches are attempting to bridge the gap between training and inference in a spirit close to ours. GLAMpoints [44] seeks to estimate homographies between retinal images and use Reinforcement Learning (RL) methods to find keypoints that are correctly matched by SIFT descriptors. Since matching is deterministic, Q-learning can be used to regress for the expected reward of each keypoint, rather than optimize directly in policy space. Using hand-crafted descriptors and only addressing the detection problem was motivated by domain-specific requirements of strong rotation equivariance, which most learned models lack. While it makes sense in the specific scenario it was developed for, it limits what the method can do.

Reinforced Feature Points [7] address the more difficult issue of learning with a general non-differentiable objective for the purpose of camera pose estimation, with RANSAC in the loop. Unfortunately, supervising all detection and matching decisions with a single reward means that this approach suffers from weak training signal, an endemic RL problem, and has to rely on pre-trained models from [9] that can only be fine-tuned. Our method can be seen as a relaxation of their approach, where we train for a surrogate objective: finding many correct feature matches. This allows for substantially more robust training from scratch and yields better downstream results.

3 Method

Given images A and B, our goal is first to extract a set of local features F_A and F_B from each and then match them to produce a set of correspondences $M_{A \leftrightarrow B}$. To learn how to do this through reinforcement learning, we redefine these two steps probabilistically. Let $P(F_I | I, \theta_F)$

be a distribution over sets of features F_I, conditional on image I and feature detection parameters θ_F, and $P(M_{A\leftrightarrow B}|F_A, F_B, \theta_M)$ be a distribution over matches between features in images A and B, conditional on features F_A, F_B, and matching parameters θ_M. Maximizing $P(M_{A\leftrightarrow B}|A, B, \theta)$ with respect to $\theta = \{\theta_F, \theta_M\}$ requires integrating the product of these two probabilities over all possible F_A, F_B, which is clearly intractable. However, we can estimate gradients $\nabla_\theta P(M_{A\leftrightarrow B}|A, B, \theta)$ via Monte Carlo sampling and use gradient descent to maximize the expected reward $\mathbb{E}_{M_{A\leftrightarrow B}\sim P(M_{A\leftrightarrow B}|A,B,\theta)}\, R(M_{A\leftrightarrow B})$ for arbitrarily defined $R(M_{A\leftrightarrow B})$.

Feature distribution $P(F_I|I, \theta_F)$**.** Our feature extraction network is based on a U-Net [33], with one output channel for detection and N for description. We denote these feature maps as $\mathbf{K}$ and $\mathbf{D}$, respectively, from which we extract features $F = \{K, D\}$. We pick N=128, for a direct comparison with SIFT and nearly all modern descriptors [21, 26, 22, 43, 13, 32].

The detection map $\mathbf{K}$ is subdivided into a grid with cell size $h \times h$, and we select at most one feature per grid cell, similarly to SuperPoint [9]. To do so, we crop the feature map corresponding to cell u, denoted $\mathbf{K}^u$, and use a softmax operator to normalize it. Our probabilistic framework samples a pixel p in cell u with probability $P_s(p|\mathbf{K}^u) = \text{softmax}(\mathbf{K}^u)_p$. This detection proposal p may still be rejected: we accept it with probability $P_a(\text{accept}_p|\mathbf{K}^u) = \sigma(\mathbf{K}^u_p)$, where $\mathbf{K}^u_p$ is the (scalar) value of the detection map $\mathbf{K}$ at location p in cell u, and σ is a sigmoid. Note that $P_s(p|\mathbf{K}^u)$ models *relative* preference across a set of different locations, whereas $P_a(\text{accept}_p|\mathbf{K}^u)$ models the *absolute* quality for location p. The total probability of sampling a feature at pixel p is thus $P(p|\mathbf{K}^u) = \text{softmax}(\mathbf{K}^u)_p \cdot \sigma(\mathbf{K}^u_p)$. Once feature locations $\{p_1, p_2, ...\}$ are known, we associate them with the l_2-normalized descriptors at this location, yielding a set of features $F_I = \{(p_1, \mathbf{D}(p_1)), (p_2, \mathbf{D}(p_2)), ...\}$. At inference time we replace softmax with $\arg\max$, and σ with the sign function. This is again similar to [9], except that we retain the spatial structure and interpret cell $\mathbf{K}^u$ in both a relative and an absolute manner, instead of creating an extra *reject* bin.

Match distribution $P(M_{A\leftrightarrow B}|F_A, F_B, \theta_M)$**.** Once feature sets F_A and F_B are known, we compute the l_2 distance between their descriptors to obtain a distance matrix $\mathbf{d}$, from which we can generate matches. In order to learn good local features it is crucial to refrain from matching ambiguous points due to repeated patterns in the image. Two solutions to this problem are cycle-consistent matching and the ratio test. Cycle-consistent matching enforces that two features be nearest neighbours of each other in descriptor space, cutting down on the number of putative matches while increasing the ratio of correct ones. The ratio test, introduced by SIFT [21], rejects a match if the ratio of the distances between its first and second nearest neighbours is above a threshold, in order to only return confident matches. These two approaches are often used in conjunction and have been shown to drastically improve results in matching pipelines [6, 16], but they are not easily differentiable.

Our solution is to relax cycle-consistent matching. Conceptually, we draw *forward* (A→B) matches for features $F_{A,i}$ from categorical distributions defined by the rows of distance matrix $\mathbf{d}$, and *reverse* (A←B) matches for features $F_{B,j}$ from distributions based on its columns. We declare $F_{A,i}$ to match $F_{B,j}$ if both the forward and reverse matches are sampled, *i.e.*, if the samples are consistent. The forward distribution of matches is given by $P_{A\to B}(j|\mathbf{d}, i) = \text{softmax}\left(-\theta_M \mathbf{d}(i, \cdot)\right)_j$, where θ_M is the single parameter, the inverse of the softmax temperature. $P_{A\leftarrow B}$ is analogously defined by $\mathbf{d}^T$.

It should be noted that, given features F_A and F_B, the probability of any particular match can be computed *exactly*: $P(i \leftrightarrow j) = P_{A\to B}(i|\mathbf{d}, j) \cdot P_{A\leftarrow B}(j|\mathbf{d}, i)$. Therefore, as long as reward R factorizes over matches as $R(M_{A\leftrightarrow B}) = \sum_{(i,j)\in M_{A\leftrightarrow B}} r(i \leftrightarrow j)$, given F_A and F_B, we can compute *exact* gradients $\nabla_{D,\theta_M}\, \mathbb{E}\, R(M_{A\leftrightarrow B})$, without resorting to sampling. This means that the matching step does not contribute to the overall variance of gradient estimation, unlike in [7], which we believe to be key to the good convergence properties of our model.

Reward function $R(M_{A\leftrightarrow B})$**.** As stated above, if the reward $R(M_{A\leftrightarrow B})$ can be factorized as a sum over individual matches, the formulation of $P(M_{A\leftrightarrow B}|F_A, F_B, \theta_M)$ allows for the use of closed-form formulas while training. For this reason we use a very simple reward, which rewards correct matches with λ_{tp} points and penalizes incorrect matches with λ_{fp} points. Let's assume we have ground-truth poses and pixel-to-pixel correspondences in the form of depth maps. We declare a match *correct* if depth is available at both $p_{A,i}$ and $p_{B,j}$, and both points lie within ϵ pixels of their respective reprojections. We declare a match *plausible* if depth is not available at either location, but the epipolar distance between the points is less than ϵ pixels, in which case we neither reward nor penalize it. We declare a match *incorrect* in all other cases.

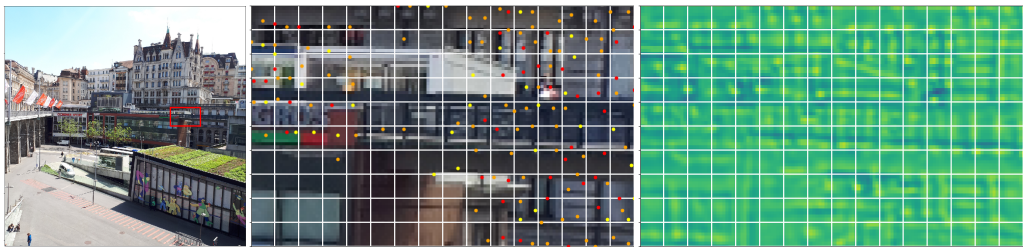


Figure 2: **Non-Maxima Suppression vs Grid-based sampling.** We demonstrate the benefits of replacing the 1-per-cell sampling approach used during training with simple NMS at inference time. For a small region of an image (left), marked by the red box, we show the chosen features (middle) and the 'heatmap' $\mathbf{K}$ (right), overlaid by the grid. Notice how maxima can be cut by cell boundaries. Keypoints are sorted by "score" and color-coded: the top third are drawn in **red**, the next third in **orange**, and the rest in **yellow**. Each cell contains at most two very salient (red) features.

Gradient estimator. With R factorized over matches and $P(i \leftrightarrow j|F_A, F_B, \theta_M)$ given as a closed formula, the application of the basic policy gradient [48] is fairly simple: with F_A, F_B sampled from their respective distributions $P(F_A|A, \theta_F), P(F_B|B, \theta_F)$ we have

$$\nabla_\theta \mathop{\mathbb{E}}_{M_{A\leftrightarrow B}} R(M_{A\leftrightarrow B}) = \mathop{\mathbb{E}}_{F_A,F_B} \sum_{i,j} \left[P(i \leftrightarrow j|F_A, F_B, \theta_M) \cdot r(i \leftrightarrow j) \cdot \nabla_\theta \Gamma_{ij}\right], \tag{1}$$

$$\text{where } \Gamma_{ij} = \log P(i \leftrightarrow j|F_A, F_B, \theta_M) + \log P(F_{A,i}|A, \theta_F) + \log P(F_{B,j}|B, \theta_F).$$

Having a closed formula for $P(i \leftrightarrow j|F_A, F_B, \theta_M)$ along with R being a sum over individual matches allows us to compute the sum in equation 1 exactly, which in the general case of REINFORCE [48] would have to be replaced with an empirical expectation over sampled matches, introducing variance in the gradient estimates. In our formulation, the only sources of gradient variance are due to mini-batch effects and approximating the expectation w.r.t. choices of F_A, F_B with an empirical sum.

It should also be noted that our formulation does not provide the feature extraction network with any supervision other than through the quality of matches those features participate in, which means that a keypoint which is never matched is considered neutral in terms of its value. This is a very useful property because keypoints may not be co-visible across two images, and should not be penalized for it as long as they do not create incorrect associations. On the other hand, this may lead to many unmatchable features on clouds and similar non-salient structures, which are unlikely to contribute to the downstream task but increase the complexity in feature matching. We address this by imposing an additional, small penalty on each sampled keypoint λ_{kp}, which can be thought of as a regularizer.

Inference. Once the models have been trained we discard our probabilistic matching framework in favor of a standard cycle-consistency check, and apply the ratio test with a threshold found empirically on a validation set. Another consideration is that our method is confined to a grid. This has two drawbacks. Firstly, it can sample at most one feature per cell. Secondly, each cell is blind to its neighbours. Our method may thus select two contiguous pixels as distinct keypoints. At inference time we can work around this issue by applying non-maxima suppression on the feature map $\mathbf{K}$, returning features at all local maxima. This addresses both issues, as illustrated in Fig. 2, at the cost of a misalignment between training and inference, which is potentially sub-optimal.

4 Experiments

We first describe our specific implementation and the training data we rely on. We then evaluate our approach on three different benchmarks, and present two ablation studies.

Training Data. We use a subset of the MegaDepth dataset [20], from which we choose 135 scenes with 63k images in total. They are posed with COLMAP, a state-of-the-art SfM framework that also provides dense depth estimates we use to establish pixel-to-pixel correspondences. We omit scenes that overlap with the test data of the Image Matching Challenge (Sec. 4.1), and apply a simple co-visibility heuristic to sample viable pairs of images. See the supplementary material for details.

	Up to 2048 features/image							Up to 8000 features/image						
	Task 1: stereo			Task 2: Multiview				Task 1: stereo			Task 2: Multiview			
Method	NM	NI	**mAA(10°)**	NM	NL	TL	**mAA(10°)**	NM	NI	**mAA(10°)**	NM	NL	TL	**mAA(10°)**
Upright Root-SIFT	194.0	112.3	0.3986	199.3	1341.7	4.09	0.5623	525.4	358.9	0.5075	542.9	4404.6	4.38	0.6792
Upright L2-Net	174.1	117.1	0.4192	179.8	1361.3	4.23	0.5968	657.3	435.7	0.5450	395.5	3603.8	4.38	0.6849
Upright HardNet	274.0	152.7	**0.4609**	201.3	1467.9	4.31	0.6354	791.7	527.6	0.5728	509.1	4250.4	4.55	0.7231
Upright GeoDesc	235.8	132.7	0.4136	161.1	1287.3	4.24	0.5837	598.9	409.9	0.5267	458.6	4146.8	4.41	0.7044
Upright SOSNet	265.6	171.2	0.4505	194.0	1442.3	4.31	0.6359	752.9	508.4	0.5738	464.4	3988.6	4.52	0.7129
Upright LogPolarDesc	296.8	162.2	0.4567	211.9	1553.4	4.33	0.6370	821.7	543.2	0.5510	505.4	4414.1	4.52	0.7109
SuperPoint	292.8	126.8	0.2964	169.3	1184.3	4.34	0.5464	–	–	–	–	–	–	–
LF-Net	191.1	106.5	0.2344	196.7	1385.0	4.14	0.5141	–	–	–	–	–	–	–
D2-Net (SS)	505.7	188.4	0.1813	513.1	**2357.9**	3.39	0.3943	1258.2	482.3	0.2228	1278.7	5893.8	3.62	0.4598
D2-Net (MS)	327.8	134.8	0.1355	337.6	**2177.3**	3.01	0.3007	1028.6	470.6	0.2506	1054.7	**6759.3**	3.39	0.4751
R2D2	273.6	213.9	0.3346	280.8	1228.4	4.29	0.6149	1408.8	**842.2**	0.4437	739.8	4432.9	4.59	0.6832
Submission #609	439.7	**270.0**	**0.4690**	280.4	1489.6	**4.69**	**0.6812**	–	–	–	–	–	–	–
Submission #578	439.5	**246.6**	0.4542	331.6	1621.7	**4.57**	**0.6741**	–	–	–	–	–	–	–
Submission #599	227.4	129.5	0.4507	176.6	1209.6	4.44	0.6609	–	–	–	–	–	–	–
Submission #611	–	–	–	–	–	–	–	945.4	622.1	**0.5887**	899.1	6086.2	**4.65**	**0.7513**
Submission #613	–	–	–	–	–	–	–	934.9	**624.1**	**0.5873**	964.8	**6350.7**	4.64	**0.7495**
Submission #625	–	–	–	–	–	–	–	945.4	605.1	**0.5878**	899.1	6095.8	**4.65**	0.7485
DISK (#708 & #709)	514.2	**404.2**	**0.5132**	527.5	**2428.0**	**5.55**	**0.7271**	1621.9	**1238.5**	0.5585	1663.8	**7484.0**	**5.92**	**0.7502**
Δ (%)	*+1.7*	*+49.7*	*+9.4*	*+2.8*	*+3.0*	*+18.3*	*+6.7*	*+15.1*	*+47.1*	*-5.4*	*+30.1*	*+10.7*	*+27.3*	*-0.1*

Table 1: **Image Matching Challenge results.** The primary metric is **(mAA)**, the mean Average Accuracy in pose estimation, up to 10^o. We also report **(NM)** the number of matches (given to RANSAC for stereo, and to COLMAP for multiview). For stereo, we also report **(NI)** the number of RANSAC inliers. For multiview, we also report **(NL)** number of landmarks (3D points), and **(TL)** track length (observations per landmark). The top 3 results are highlighted in **red**, **green** and **blue**.

Feature extraction network. We use a variation of the U-Net [33] architecture. Our model has 4 down- and up-blocks which consist of a single convolutional layer with 5×5 kernels, unlike the standard U-Net that uses two convolutional layers per block. We use instance normalization instead of batch normalization, and PReLU non-linearities. Our models comprise 1.1M parameters, with a formal receptive field of 219×219 pixels. We will release the source code for training and inference.

Optimization. Although the matching stage has a single learnable parameter, θ_M, we found that gradually increasing it with a fixed schedule works well, leaving just the feature extraction network to be learned with gradient descent. Since the training signal comes from *matching features*, we process three co-visible images A, B and C per batch. We then evaluate the summation part of equation 1 for pairs $A \leftrightarrow B$, $A \leftrightarrow C$, $B \leftrightarrow C$ and accumulate the gradients w.r.t. θ. While matching is pair-wise, we obtain three image pairs per image triplet. By contrast, two pairs of unrelated scenes would require four images. Our approach provides more matches while reducing GPU memory for feature extraction. We rescale the images such that the longer edge has 768 pixels, and zero-pad the shorter edge to obtain a square input. Grid cells are square, with each side $h = 8$ pixels.

Rewards are $\lambda_{\text{tp}} = 1$, $\lambda_{\text{fp}} = -0.25$ and $\lambda_{\text{kp}} = -0.001$. Since a randomly initialized network tends to generate very poor matches, the quality of keypoints is negative on average at first, and the network would cease to sample them at all, reaching a local maximum reward of 0. To avoid that, we anneal λ_{fp} and λ_{kp} over the first 5 epochs, starting with 0 and linearly increasing to their full value at the end.

Since our model uses instance normalization instead of batch normalization, it is batch size-agnostic. To improve our gradient estimates at a fixed memory budget, we accumulate gradients over two batches before performing a single gradient update. We use ADAM [18] with learning rate of 10^{-4}. To pick the best checkpoint, we evaluate performance in terms of pose estimation accuracy in stereo, with DEGENSAC [8]. Specifically, every 5k optimization steps we compute the mean Average Accuracy (mAA) at a 10^o error threshold, as in [17]: see Sec. 4.1 and the appendix for details.

Finally, our method produces a variable number of features. To compare it to others under a fixed feature budget, we subsample them by their "score", that is, the value of heatmap $\mathbf{K}$ at that location.

4.1 Evaluation on the 2020 Image Matching Challenge [17] – Table 1, Figures 3 and 4

The Image Matching Challenge provides a benchmark that can be used to evaluate local features for two tasks: stereo and multi-view reconstruction. For the stereo task, features are extracted across every pair of images and then given to RANSAC, which is used to compute their relative pose. The multiview task uses COLMAP to generate SfM reconstructions from small subsets of 5, 10, and 25

Figure 3: **Stereo results on the Image Matching Challenge (2k features).** Top: DoG w/ Upright HardNet descriptors [26]. Bottom: DISK. We extract cycle-consistent matches with optimal parameters and feed them to DEGENSAC [8]. We plot the resulting inliers, from green to yellow if they are correct (0 to 5 pixels in reprojection error), in red if they are incorrect (above 5), and in blue if ground truth depth is not available. Our approach can match many more points and produce more accurate poses. It can deal with large changes in scale (4th and 5th columns) but not in rotation (6th column), a problem that could be solved with more data augmentation.

images. The differentiating factor for this benchmark is that both tasks are evaluated *downstream*, in terms of the quality of the reconstructed poses, which are compared to the ground truth, by using the mean Average Accuracy (mAA) up to a 10-degree error threshold. While this requires carefully tuning components extraneous to local features, such as RANSAC hyperparameters, it measures performance on real problems, rather than intermediate metrics.

Hyperparameter Selection. We rely on a validation set of two scenes: "Sacre Coeur" and "St. Peter's Square". We resize the images to 1024 pixels on the longest edge, generate cycle-consistent matches with the ratio test, with a threshold of 0.95. For stereo we use DEGENSAC [8], which outperforms vanilla RANSAC [16], with an inlier threshold of 0.75 pixels.

Results. We extract DISK features for the nine test scenes, for which the ground truth is kept private, and submit them to the organizers for processing. The challenge has two categories: up to 2k or 8k features per image. We participate in both. We report the results in Table 1, along with baselines taken directly from the leaderboards, computed in [16]. We consider several descriptors on DoG keypoints: RootSIFT [21, 2] L2-Net [42], HardNet [26], GeoDesc [23], SOSNet [43] and LogPolarDesc [13]. For brevity, we show only their upright variants, which perform better than their rotation-sentitive counterparts on this dataset. For end-to-end methods, we consider SuperPoint [10], LF-Net [30], D2-Net [11], and R2D2 [32]. All of these methods use DEGENSAC [8] as a RANSAC variant for stereo, with their optimal hyperparameters. We also list the top 3 user submissions for each category, taken from the leaderboards on June 5, 2020 (the challenge concluded on May 31, 2020).

On the 2k category, we outperform all methods by 9.4% relative in stereo, and 6.7% relative in multiview. On the 8k category, averaging stereo and multiview, we outperform all baselines, but place slightly below the top three submissions. Our method can find many more matches than any other, easily producing 2-3x the number of RANSAC inliers or 3D landmarks. Our features used for the 2k category are a subset of those used for 8k, which indicates a potentially sub-optimal use of the increased budget, which may be solved training with larger images or smaller grid cells. We show qualitative images in Figs. 3 and 4. Further results are available in the supplementary material.

Note that we only compare with submissions using the built-in feature matcher, based on the l_2 distance between descriptors, instead of neural-network based matchers [50, 53, 35], which combined

Figure 4: **Multiview results on the Image Matching Challenge (8k features).** Top: DoG w/ Upright HardNet descriptors [26]. Bottom: DISK. COLMAP is used to reconstruct the "London Bridge" scene with 25 images. We show three of them and draw their keypoints, in blue if they are registered by COLMAP, and red otherwise. Our method generates evenly distributed features, producing 76% more landmarks with 30% more observations per landmark than HardNet. Keypoints on water or trees have low scores and are rare among the top 2k features, but appear more often when taking 8k. This suggests that our method can reach near-optimal performance on a small budget.

with state-of-the-art features obtain the best overall results. Even so, DISK places #2 below only SuperGlue [35] on the 2k category, outperforming *all other solutions* using learned matchers.

4.2 Evaluation on HPatches [4] – Fig. 5

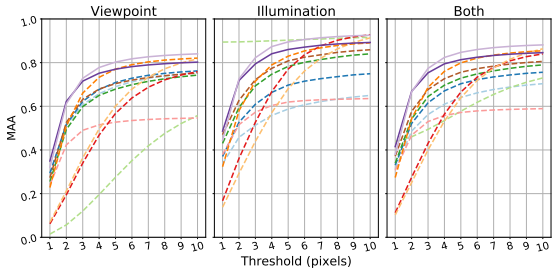
HPatches contains 116 scenes with 6 images each. These scenes are strictly planar, containing only viewpoint or illumination changes (not both), and use homographies as ground truth. Despite its limitations, it is often used to evaluate low-level matching accuracy. We follow the evaluation methodology and source code from [11]. The first image on every scene is matched to the remaining five, omitting 8 scenes with high-resolution images. Cyclic-consistent matches are computed, and performance is measured in terms of the Mean Matching Accuracy (MAA), *i.e.*, the ratio of matches with a reprojection error below a threshold, from 1 to 10 pixels, and averaged across all image pairs.

We report MAA in Fig. 5, and summarize it by its Area under the Curve (AUC), up to 5 pixels. Baselines include RootSIFT [21, 2] on Hessian-Affine keypoints [25], a learned affine region detector (HAN) [27] paired with HardNet++ descriptors [26], DELF [29], SuperPoint [10], D2-Net [11], R2D2 [32], and Reinforced Feature Points (RFP) [7]. For D2-Net we include both single- (SS) and multi-scale (MS) models. We consider DISK with number of matches restricted to 2k and 8k, for a fair comparison with different methods.

We obtain state-of-the-art performance on this dataset, despite the fact that our models are trained on non-planar data without strong affine transformations. We use the same models and hyperparameters used in the previous section to obtain 2k and 8k features, without any tuning. Our method is #1 on the viewpoint scenes, followed by R2D2, and #2 on the illumination scenes, trailing DELF. Putting them together, it outperforms its closest competitor, RFP, by 12% relative.

4.3 Evaluation on the ETH-COLMAP benchmark [40] – Table 2

This benchmark compiles statistics for large-scale SfM. We select three of the smaller scenes and report the results in Table 2. Baselines are taken from [7] and include Root-SIFT [21, 3], SuperPoint [10], and Reinforced Feature Points [7]. We obtain more landmarks than SIFT, with larger tracks and a comparable reprojection error. Note that this benchmark does not standardize the number of input features, so we extract DISK at full resolution and take the top 12k keypoints in order to remain comparable with SIFT. By comparison, a run on "Fountain" with no cap yields 67k landmarks.



Method	Num. Features	Num. Matches	Viewp. AUC(5px)	Illum. AUC(5px)	Both AUC(5px)
HesAff/RootSIFT	6710.1	2851.7	0.552	0.491	0.523
HAN/HardNet++	3860.8	1960.0	0.564	0.573	0.568
DELF	4590.0	1940.3	0.132	**0.898**	0.501
SuperPoint	1562.6	883.4	0.535	0.650	0.590
Reinforced FP	—	—	0.563	0.680	0.621
LF-Net	500.0	177.5	0.439	0.538	0.487
D2-Net (SS)	5965.1	2495.9	0.326	0.499	0.409
D2-Net (MS)	8254.5	2831.6	0.349	0.424	0.385
R2D2	4989.8	1846.4	0.580	0.647	0.613
DISK (2k)	2048.0	1024.2	0.642	0.740	0.689
DISK (8k)	7705.1	3851.8	**0.648**	0.752	**0.698**

Figure 5: **Results on HPatches.** On the left, we report Mean Matching Accuracy (MAA) at 10 pixel thresholds. On the right, we summarize MAA by its AUC, up to 5 pixels. Results for RFP [7] were kindly provided by the authors, which explains why keypoint/match counts are missing.

Scene	Method	NL	TL	ϵ_r
Fountain	Root-SIFT	15k	4.70	**0.41**
	SP	**31k**	4.75	0.97
	RFP	9k	4.86	0.87
	DISK	18k	**5.52**	0.50
Herzjesu	Root-SIFT	8k	4.22	**0.46**
	SP	**21k**	4.10	0.95
	RFP	7k	4.32	0.82
	DISK	11k	**4.71**	0.48
South Building	Root-SIFT	113k	5.92	**0.58**
	SP	**160k**	7.83	0.92
	RFP	102k	7.86	0.88
	DISK	115k	**9.91**	0.59

Table 2: **Results on ETH-COLMAP [40].** We compare Root-SIFT [21], SuperPoint [10], Reinforced Feature Points [7], and DISK. We report: **(NL)** number of landmarks, **(TL)** track length (average number of observations per landmark), and (ϵ_r) reprojection error.

Variant	2k features Stereo	2k features Multiview	8k features Stereo	8k features Multiview
Depth	**0.7218**	0.8325	**0.7767**	0.8628
Epipolar	0.7145	**0.8465**	0.7718	**0.8749**

Table 3: **Ablation: match supervision.** We compare mAA on the Image Matching Challenge validation set, for DISK models learned with pixel-to-pixel supervision or epipolar constraints.

Variant	Num. features	Num. matches	Stereo mAA(10^0)	Multiview mAA(10^0)
1-per-cell	5456.8	796.5	0.74774	0.84685
NMS 3×3	8434.6	1699.9	**0.77833**	0.86864
NMS 5×5	7656.0	1547.9	0.77657	**0.87622**
NMS 7×7	6423.4	1271.1	0.77070	0.85642
NMS 9×9	4946.2	942.0	0.75558	0.85362

Table 4: **Ablation: NMS.** We compare the feature selection strategy used for training (top) with NMS at inference time. Here we use *all detected features*, rather than subsample by score.

4.4 Ablation Study

Supervision without Depth. As outlined in Sec. 3, we use the strongest supervision signal available to us, which are depth maps. Unfortunately, this means we only reward matches on areas with reliable depth estimates, which may cause biases. We also experimented with a variant of R that relies only on *epipolar constraints*, as in a recent paper [46]. We evaluate both variants on the validation set of the Image Matching Challenge and report the results in Table 3. Performance improves for multiview but decreases for stereo. Qualitatively, we observe that new keypoints appear on textureless areas outside object boundaries, probably due to the U-Net's large receptive field (see appendix). Nevertheless, this illustrates that DISK can be learned just as effectively with much weaker supervision.

Non-Maximum Suppression. We compare the training regime, where we sample at most one feature per grid cell, against the inference regime, where we apply NMS on the heatmap. We report results in terms of pose mAA on the validation set of the Image Matching Challenge in Table 4. For this experiment we removed the budget limit and took all features provided by the model. This shows that this inference strategy is clearly beneficial, despite departing from the training pipeline.

5 Conclusions and future work

We introduced a novel probabilistic approach to learn local features end to end with policy gradient. It can easily train from scratch, and yields many more matches than its competitors. We demonstrate state-of-the-art results in pose accuracy for stereo and 3D reconstruction, placing #1 in the 2k-keypoints category of the Image Matching Challenge using off-the-shelf matchers. In future work we intend to replace the match relaxation introduced in Sec. 3, with learned matchers such as [50, 34].

6 Broader Impact

There already are many applications that rely on keypoints. The technique presented here has the potential to make them more effective but we do not foresee any further societal impact.

References

[1] S. Agarwal, N. Snavely, I. Simon, S.M. Seitz, and R. Szeliski. Building Rome in One Day. In *International Conference on Computer Vision*, 2009. 1

[2] Relja Arandjelović and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *Conference on Computer Vision and Pattern Recognition*, pages 2911–2918, 2012. 6, 7

[3] R. Arandjelovic and A. Zisserman. All About VLAD. In *Conference on Computer Vision and Pattern Recognition*, pages 1578–1585, 2013. 2, 7

[4] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk. Hpatches: A Benchmark and Evaluation of Handcrafted and Learned Local Descriptors. In *Conference on Computer Vision and Pattern Recognition*, 2017. 7

[5] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. SURF: Speeded Up Robust Features. *Computer Vision and Image Understanding*, 10(3):346–359, 2008. 2

[6] Fabio Bellavia and Carlo Colombo. Is there anything new to say about sift matching? *International Journal of Computer Vision*, pages 1–20, 2020. 3

[7] Aritra Bhowmik, Stefan Gumhold, Carsten Rother, and Eric Brachmann. Reinforced feature points: Optimizing feature detection and description for a high-level task. *arXiv preprint arXiv:1912.00623*, 2019. 2, 3, 7, 8

[8] O. Chum and J. Matas. Matching with Prosac - Progressive Sample Consensus. In *Conference on Computer Vision and Pattern Recognition*, pages 220–226, June 2005. 5, 6

[9] D. Detone, T. Malisiewicz, and A. Rabinovich. Superpoint: Self-Supervised Interest Point Detection and Description. In *Conference on Computer Vision and Pattern Recognition*, 2018. 2, 3

[10] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 224–236, 2018. 6, 7, 8

[11] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler. D2-Net: A Trainable CNN for Joint Detection and Description of Local Features. In *Conference on Computer Vision and Pattern Recognition*, 2019. 2, 6, 7

[12] Mihai Dusmanu, Johannes L Schönberger, and Marc Pollefeys. Multi-view optimization of local feature geometry. *arXiv preprint arXiv:2003.08348*, 2020. 12

[13] P. Ebel, A. Mishchuk, K. M. Yi, P. Fua, and E. Trulls. Beyond Cartesian Representations for Local Descriptors. In *International Conference on Computer Vision*, 2019. 1, 2, 3, 6

[14] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg. MatchNet: Unifying Feature and Metric Learning for Patch-Based Matching. In *Conference on Computer Vision and Pattern Recognition*, 2015. 2

[15] J. Heinly, J.L. Schoenberger, E. Dunn, and J.-M. Frahm. Reconstructing the World in Six Days. In *Conference on Computer Vision and Pattern Recognition*, 2015. 1

[16] Y. Jin, D. Mishkin, A. Mishchuk, J. Matas, P. Fua, K.M. Yi, and E. Trulls. Image Matching across Wide Baselines: From Paper to Practice. *arXiv Preprint*, 2020. 2, 3, 6

[17] Yuhe Jin, Dmytro Mishkin, Anastasiia Mishchuk, Jiri Matas, Pascal Fua, Kwang Moo Yi, and Eduard Trulls. Image Matching across Wide Baselines: From Paper to Practice. *arXiv Preprint*, 2020. 1, 5, 12

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[19] Axel Barroso Laguna, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Key. net: Keypoint detection by handcrafted and learned cnn filters. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5835–5843. IEEE, 2019. 2

[20] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 4

[21] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 20(2):91–110, November 2004. 1, 2, 3, 6, 7, 8

[22] Z. Luo, T. Shen, L. Zhou, J. Zhang, Y. Yao, S. Li, T. Fang, and L. Quan. Contextdesc: Local Descriptor Augmentation with Cross-Modality Context. In *Conference on Computer Vision and Pattern Recognition*, 2019. 3

[23] Z. Luo, T. Shen, L. Zhou, S. Zhu, R. Zhang, Y. Yao, T. Fang, and L. Quan. Geodesc: Learning Local Descriptors by Integrating Geometry Constraints. In *European Conference on Computer Vision*, 2018. 6

[24] S. Lynen, B. Zeisl, D. Aiger, M. Bosse, J. Hesch, M. Pollefeys, R. Siegwart, and T. Sattler. Large-scale, real-time visual-inertial localization revisited. *International Journal of Robotics Research*, 2020. 1

[25] K. Mikolajczyk and C. Schmid. Scale and Affine Invariant Interest Point Detectors. *International Journal of Computer Vision*, 60:63–86, 2004. 7

[26] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas. Working Hard to Know Your Neighbor's Margins: Local Descriptor Learning Loss. In *Advances in Neural Information Processing Systems*, 2017. 1, 2, 3, 6, 7

[27] D. Mishkin, F. Radenovic, and J. Matas. Repeatability is Not Enough: Learning Affine Regions via Discriminability. In *European Conference on Computer Vision*, 2018. 7

[28] R. Mur-Artal, J. Montiel, and J. Tardós. Orb-Slam: A Versatile and Accurate Monocular Slam System. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. 1

[29] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *International Conference on Computer Vision*, pages 3456–3465, 2017. 7

[30] Y. Ono, E. Trulls, P. Fua, and K. M. Yi. Lf-Net: Learning Local Features from Images. In *Advances in Neural Information Processing Systems*, 2018. 2, 6

[31] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello. Enet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation. In *arXiv Preprint*, 2016. 12

[32] J. Revaud, P. Weinzaepfel, C. De Souza, N. Pion, G. Csurka, Y. Cabon, and M. Humenberger. R2D2: Repeatable and Reliable Detector and Descriptor. In *arXiv Preprint*, 2019. 2, 3, 6, 7

[33] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Conference on Medical Image Computing and Computer Assisted Intervention*, pages 234–241, 2015. 3, 5

[34] P.E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich. SuperGlue: Learning Feature Matching with Graph Neural Networks. *Conference on Computer Vision and Pattern Recognition*, 2020. 8

[35] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. *Conference on Computer Vision and Pattern Recognition*, 2020. 6, 7

[36] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe. Understanding the limitations of CNN-based absolute camera pose regression. In *Conference on Computer Vision and Pattern Recognition*, pages 3302–3312, 2019. 1

[37] N. Savinov, A. Seki, L. Ladicky, T. Sattler, and M. Pollefeys. Quad-Networks: Unsupervised Learning to Rank for Interest Point Detection. In *Conference on Computer Vision and Pattern Recognition*, 2017. 2

[38] J.L. Schönberger and J.M. Frahm. Structure-From-Motion Revisited. In *Conference on Computer Vision and Pattern Recognition*, 2016. 1, 12

[39] J.L. Schönberger, H. Hardmeier, T. Sattler, and M. Pollefeys. Comparative Evaluation of Hand-Crafted and Learned Local Features. In *Conference on Computer Vision and Pattern Recognition*, 2017. 1

[40] Johannes Lutz Schönberger, Hans Hardmeier, Torsten Sattler, and Marc Pollefeys. Comparative Evaluation of Hand-Crafted and Learned Local Features. In *Conference on Computer Vision and Pattern Recognition*, 2017. 7, 8

[41] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer. Discriminative Learning of Deep Convolutional Feature Point Descriptors. In *International Conference on Computer Vision*, 2015. 1, 2

[42] Y. Tian, B. Fan, and F. Wu. L2-Net: Deep Learning of Discriminative Patch Descriptor in Euclidean Space. In *Conference on Computer Vision and Pattern Recognition*, 2017. 1, 6

[43] Yurun Tian, Xin Yu, Bin Fan, Fuchao Wu, Huub Heijnen, and Vassileios Balntas. SOSNet: Second order similarity regularization for local descriptor learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11016–11025, 2019. 1, 2, 3, 6

[44] P. Truong, S. Apostolopoulos, A. Mosinska, S. Stucky, C. Ciller, and S. De Zanet. Glampoints: Greedily learned accurate match points. In *International Conference on Computer Vision*, 2019. 2

[45] Y. Verdie, K. M. Yi, P. Fua, and V. Lepetit. TILDE: A Temporally Invariant Learned DEtector. In *Conference on Computer Vision and Pattern Recognition*, 2015. 2

[46] Qianqian Wang, Xiaowei Zhou, Bharath Hariharan, and Noah Snavely. Learning feature descriptors using camera pose supervision. *arXiv preprint arXiv:2004.13324*, 2020. 8

[47] R.J. Williams. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning*, 1992. 1

[48] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992. 4

[49] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua. LIFT: Learned Invariant Feature Transform. In *European Conference on Computer Vision*, 2016. 2

[50] K. M. Yi, E. Trulls, Y. Ono, V. Lepetit, M. Salzmann, and P. Fua. Learning to Find Good Correspondences. In *Conference on Computer Vision and Pattern Recognition*, 2018. 6, 8

[51] K. M. Yi, Y. Verdie, P. Fua, and V. Lepetit. Learning to Assign Orientations to Feature Points. In *Conference on Computer Vision and Pattern Recognition*, 2016. 2

[52] S. Zagoruyko and N. Komodakis. Learning to Compare Image Patches via Convolutional Neural Networks. In *Conference on Computer Vision and Pattern Recognition*, 2015. 2

[53] Jiahui Zhang, Dawei Sun, Zixin Luo, Anbang Yao, Lei Zhou, Tianwei Shen, Yurong Chen, Long Quan, and Hongen Liao. Learning two-view correspondences and geometry using order-aware network. In *Conference on Computer Vision and Pattern Recognition*, pages 5845–5854, 2019. 6

APPENDIX — DISK: Learning local features with policy gradient

Supplementary material for NeurIPS submission 1194.

Training data

In order to avoid image pairs which are not co-visible and ones which are too easy, we use a simple procedure. For every image I we access the set of their 3D keypoints $\{L\}_I$, as provided in the COLMAP [38] metadata for the dataset, and for each pair A, B we compute the ratio

$$r = \frac{|\{L\}_A \cap \{L\}_B|}{\min(|\{L\}_A|, |\{L\}_B|)}$$

which we use as a proxy for co-visibility, and pick all pairs A, B for which $0.15 \leq r \leq 0.8$. In order to obtain the image triplets used during training, we randomly sample a "seed" image A and then two more images B, C among those with which A was paired, based on the ratio criterion described above. We do not enforce that B and C are co-visible with respect to the criterion. We perform this sampling until we obtain roughly 10k triplets per scene.

We manually blacklist scenes which overlap with the test subset of the Image Matching Challenge: '0024' ("British Museum"), '0021' ("Lincoln Memorial Statue"), '0025' ("London Bridge"), '1589' ("Mount Rushmore"), '0019' ("Sagrada Familia"), '0008' ("Piazza San Marco"), '0032' ("Florence Cathedral"), and '0063' ("Milan Cathedral"). We also blacklist scenes which overlap with the validation subset of the Image Matching Challenge: '0015' ("St. Peter's Square") and '0022' ("Sacre Coeur"), which we use for the purposes of validation and hyperparameter selection, as in [17]. Finally, as per [12], we blacklist scenes with low quality depth maps: '0000', '0002', '0011', '0020', '0033', '0050', '0103', '0105', '0143', '0176', '0177', '0265', '0366', '0474', '0860', and '4541', as well as automatically remove scenes which produced less than 10k co-visible triplets.

In effect, we train on 135 scenes yielding $\approx$ 133k co-visible triplets. A full list will be provided along with the source code, with the final version of the paper.

Continuous evaluation

With so many co-visible triplets a single iteration through the dataset (epoch) would take very long. In order to continuously evaluate performance of the model, we pause every 5k optimization steps (10k triplets) and evaluate stereo performance. To do so, we re-implement the mAA(10^o) metric used by the benchmark [17], and apply it to a smaller subset of the validation set. We pick our best model according to this metric and then proceed with hyper-parameter tuning as described in Sec. 4.1. Our highest-performing model was obtained after 300k optimization steps.

Computational cost

Our code, implemented in PyTorch [31], is run on an NVIDIA V100 GPU, with F32 precision. At inference time we obtain $\approx$ 7 frames per second for 1024×1024 input and training with 768×768 input requires ≈ 1.2 seconds per triplet. Training memory consumption is dominated by descriptor distance matrices and is on the order of 11 Gb except for the first epoch, where the network indiscriminately samples lots of features. Overall we expect our research to be replicable on a GPU with 11 Gb memory, perhaps with small optimizations necessary.

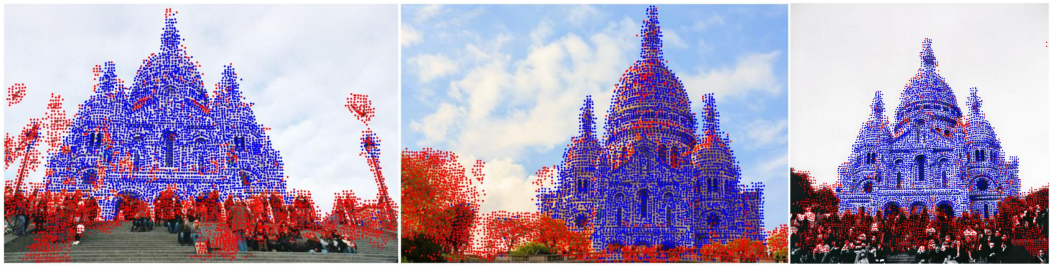
Qualitative results for epipolar supervision – Fig. 6

As outlined in Sec. 4.4, our models may be supervised with pixel-to-pixel correspondences in the form of depth maps, or with simple epipolar constraints. With the latter, points appear around 3D object boundaries, as illustrated in Fig. 6. For simplicity, the main paper focuses on models trained with depth-based supervision.

Breakdown by scene for the Image Matching Challenge [17] – Tables 5 and 6

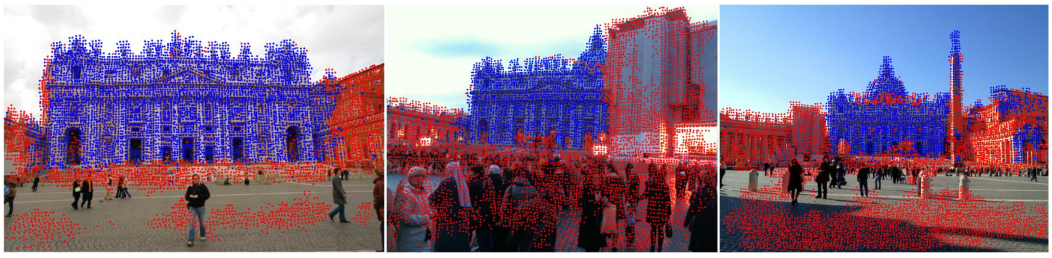
We break down out results per scene in Table 5, for 2k features, and Table 6, for 8k features. Values copied from the challenge leaderboards (submissions #708 and #709).

(a) “Sacre Coeur” w/ depth-based supervision



(b) “Sacre Coeur” w/ epipolar-based supervision

(c) “Saint Peter’s Square” w/ depth-based supervision



(d) “Saint Peter’s Square” w/ epipolar-based supervision

Figure 6: **Qualitative results: depth vs epipolar supervision.** With depth-based supervision, our models learn to (usually) avoid textureless areas such as the sky. With epipolar-based supervision, points appear on the boundaries of 3D objects. They may or may not be matched: see for instance the obelisk on the rightmost images for (c) and (d). Thin structures, such as the lamp-posts on the leftmost images for (a) and (b), create features with epipolar supervision but not with depth supervision, presumably because they are typically absent in the depth maps. To illustrate this point we use the validation set from the Image Matching Challenge, following the same convention as in Fig. 4.

Scene	Num. Features	Task 1: stereo Input Matches	Num. Inliers	**mAA(10°)**	Task 2: Multiview Input Matches	Num. Landmarks	Track Length	**mAA(10°)**
British Museum	2048.0	717.9	571.9	0.4199	716.4	2223.5	6.55	0.6947
Florence Cathedral	2048.0	514.8	400.2	0.6922	530.8	2630.4	5.27	0.7563
Lincoln Memorial Statue	2048.0	430.9	326.5	0.5909	461.2	2098.4	5.66	0.8561
London Bridge	2048.0	452.3	350.8	0.5857	544.3	2009.8	6.19	0.8078
Milan Cathedral	2048.0	685.6	555.8	0.5267	672.1	2525.4	6.15	0.6840
Mount Rushmore	2048.0	471.7	392.6	0.3786	463.5	2534.8	4.88	0.5356
Piazza San Marco	2048.0	341.6	265.1	0.2603	338.9	2726.1	4.28	0.6033
Sagrada Familia	2048.0	474.2	366.6	0.5770	466.0	2696.9	5.10	0.8107
St. Paul's Cathedral	2048.0	539.1	408.3	0.5870	554.1	2407.0	5.83	0.7949
Average	2048.0	514.2	404.2	0.5132	527.5	2428.0	5.55	0.7271

Table 5: **Image Matching Challenge: Breakdown by scene (2k features).** We report results for each of the 9 scenes, and their average.

Scene	Num. Features	Task 1: stereo Input Matches	Num. Inliers	**mAA(10°)**	Task 2: Multiview Input Matches	Num. Landmarks	Track Length	**mAA(10°)**
British Museum	7839.8	1990.9	1530.4	0.4986	2002.2	6657.3	6.67	0.7377
Florence Cathedral	7996.9	1864.7	1415.9	0.7246	1927.3	8609.1	5.84	0.7840
Lincoln Memorial Statue	7597.3	948.2	649.4	0.6249	1029.8	5977.8	5.39	0.8851
London Bridge	7421.6	1073.3	811.7	0.6312	1333.9	5297.3	6.28	0.8208
Milan Cathedral	7887.5	2165.3	1703.2	0.5764	2135.0	7381.2	6.77	0.7031
Mount Rushmore	7976.1	1996.3	1612.9	0.4394	1961.5	8209.1	5.92	0.6103
Piazza San Marco	7999.0	1141.2	871.2	0.2842	1136.8	8675.5	4.53	0.5812
Sagrada Familia	7982.0	1870.3	1408.0	0.6170	1841.1	9154.7	5.80	0.8260
St. Paul's Cathedral	7897.3	1546.7	1144.0	0.6300	1606.6	7393.8	6.09	0.8039
Average	7844.2	1621.9	1238.5	0.5585	1663.8	7484.0	5.92	0.7502

Table 6: **Image Matching Challenge: Breakdown by scene (8k features).** We report results for each of the 9 scenes, and their average.