[Submission 1194: “DISK”] We thank all reviewers for their insightful comments, and address their concerns.

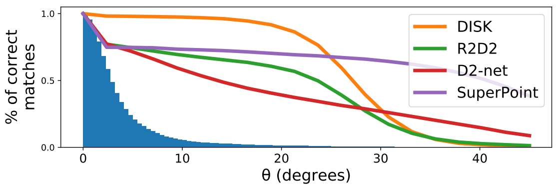

Figure A: Rotation invariance vs. rotations in data.

Cell \ NMS	3×3	5×5	7×7	9×9
8×8	0.7751	**0.7824**	0.7778	0.7586
12×12	0.7576	**0.7580**	0.7502	0.7431
16×16	0.7213	**0.7214**	0.7120	0.6999

Figure B: mAA vs. cell size & NMS on IMW2020 (val).

R1, R5: domain-specific engineering & lack of mathematical innovations. Only one other work applies policy gradient to local features [7]. It relies on non-differentiable methods based on assumptions on the pre-trained model ([7], Sec. 3.3, points 1 and 2). Instead, we optimize a simpler objective function and exploit its structure to reduce gradient variance (L129-133 in our paper), allowing us to train from scratch, unlike [7] that we outpeform on two datasets. In short, ours is the first learned, end-to-end method to outperform well-tuned baselines using hand-crafted detectors. Finally, please note that [7] was officially published after the NeurIPS submission deadline, which by NeurIPS guidelines makes it a contemporaneous submission.

R1: DISK is based on previous work (U-Net, SuperPoint) and only offers moderate innovation. The only similarity with SuperPoint is that we also use a CNN to densely find keypoint score maps and descriptors. SuperPoint is not a RL method and uses neither feature/match *distributions* nor a *reward* function. We used a U-Net because it is a proven architecture and our focus was more on the RL algorithm than on developing a specific architecture.

R1, R4, R5: Rotation invariance. Limited rotation invariance is a *deliberate* choice, because rotation estimation is counterproductive for upright images: see [14] (Sec. 6.5, Tables 10-11). As an experiment, we randomly pick 36 images from the IMW2020 test set, and extract and match features between them and their copies, rotated by θ. We compute the ratio of correct matches (within a 3px threshold) and show it in Fig. A. We also superimpose a histogram of relative image rotations between all pairs of images on the IMW2020 validation set. Our current approach is *extremely* robust to the rotations found in the data, which could be further increased by data augmentation. We will clarify this in the paper.

R1: Hyperparameters. We have relatively few of them: (1) cycle-consistency temperature θ_M (2) true positive reward λ_{tp} (3) false positive penalty λ_{fp} (4) detection cost λ_{kp}. Aside from the initial annealing of λ_{fp} and λ_{kp} (L172-175), we did not find the method sensitive to hyperparameters, including ADAM LR. They were chosen arbitrarily and found to work well. We tuned inference parameters (NMS window & RANSAC settings) by search, as described in L194-197.

R1, R3, R5: What is the contribution of individual components of the pipeline? Can they be replaced? We do not view DISK as a series of independent components. Because we maintain a probabilistic interpretation throughout the pipeline, we can easily reason about the effect of hyperparameters $\lambda_{tp}, \lambda_{fp}$ and λ_{kp}. An ablation study, such as replacing our matching scheme with a margin loss [23], would require “plumbing” to balance the respective loss terms, making the comparison unreliable. We experimented with an alternative matching relaxation, using the entropy of the match distribution as a proxy for confidence (in place of cycle-consistency). It performed comparably while requiring more hyperparameters and computation, and we dropped it from the submission due to space constraints and simplicity.

R3: Relative vs. absolute importance of features. Absolute importance measures keypoint quality. Relative importance is a mechanism to enforce feature sparsity in a differentiable manner. Absolute importance can be paired with a different sparsity mechanism – in fact, for inference we replace relative importance with NMS.

R3: Cell size vs. NMS. We find models trained with 8×8 to outperform larger grid cells, regardless of NMS window. Fig. B summarizes this for different settings, on IMW2020. For brevity, we average stereo and multiview performance.

R3: Feature “duplication” on cell borders. Experimentally, we observe that 19.9% of features from grid selection (training) have a neighbour within 2 px. This has three potential downsides. (1) Compute/memory is increased, due to redundancies. (2) It rescales λ_{kp}. Imagine that some detections are *strictly duplicated*. The probability of matching two locations remains constant – this means that learning dynamics are not impacted, other than λ_{kp} acting more strongly (on a larger number of detections). (3) Detections are *close by*, instead of duplicated, which may make the algorithm less spatially precise: since duplication means a failure of the sparsity mechanism, we learn in a regime where imprecise correspondences are more common than at inference, slightly favoring shift-invariance in the descriptors. However, DISK is #1 on HPatches, even at a 1px threshold, and attains very low reprojection error on ETH-COLMAP benchmark.

R5: Features on textureless areas. We claim that features *outside object boundaries* are matched using contextual information. Fig. 6 of the appendix illustrates this with detections on the sky (many of them matched – blue dots) near objects of interest. Since the sky has no intrinsic features, only the spatial context could be used to match them.

R5: Motivation for policy gradient and relation to [7] and [9]. Please note we discuss this in L18-L30 and L51-65.