RaCo: Ranking and Covariance for Practical Learned Keypoints

Abhiram Shenoi[1] Philipp Lindenberger[1] Paul-Edouard Sarlin[2] Marc Pollefeys[1,3]
[1]ETH Zürich [2]Google [3]Microsoft Mixed Reality & AI Lab

Abstract

*This paper introduces **RaCo**, a lightweight neural network designed to learn robust and versatile keypoints suitable for a variety of 3D computer vision tasks. The model integrates three key components: the repeatable keypoint detector, a differentiable ranker to maximize matches with a limited number of keypoints, and a covariance estimator to quantify spatial uncertainty in metric scale. Trained on perspective image crops only, RaCo operates without the need for covisible image pairs. It achieves strong rotational robustness through extensive data augmentation, even without the use of computationally expensive equivariant network architectures. The method is evaluated on several challenging datasets, where it demonstrates state-of-the-art performance in keypoint repeatability and two-view matching, particularly under large in-plane rotations. Ultimately, RaCo provides an effective and simple strategy to independently estimate keypoint ranking and metric covariance without additional labels, detecting interpretable and repeatable interest points. The code is available at:* https://github.com/cvg/RaCo

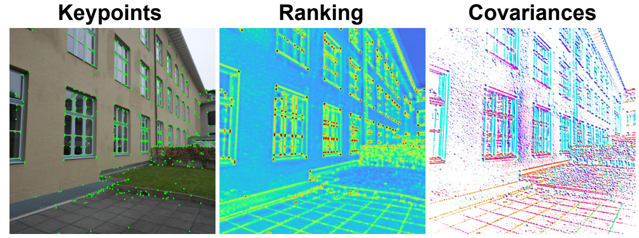

Figure 1. **Practical interest point detection.** RaCo detects repeatable and interpretable corners (left), learned from perspective image crops. A dedicated ranking head (middle) maximizes the downstream accuracy-speed trade-off by ranking matchable points higher. The estimated 2D metric covariances (right) describe the keypoints' spatial uncertainty in pixels (colored by the angle of the first eigenvector, and whitened where the variance is large).

1. Introduction

Sparse interest points are a key building block for large-scale 3D computer vision systems, enabling applications like 3D reconstruction [2, 50] and visual localization [46, 49]. A keypoint is a 2D point on a distinctive image region that can be reliably detected from various viewpoints and under different appearance changes, enabling multi-view association. A set of keypoints form a sparse representation of an image, which reduces repetitive computations in downstream algorithms and thus increases their scalability. Initially estimated via heuristics on low-level image statistics [23, 35, 43], modern data-driven approaches have proven to be significantly more robust [15, 16, 42, 58]. These detectors are trained using either synthetic data with ground-truth labels [15] or self-supervision objectives without ground-truth labels [58, 67]. One then typically finds correspondences between keypoints across images using local descriptors.

The community has recently evaluated the performance of keypoint detection and description jointly on downstream tasks that are based on correspondences, such as relative pose estimation [38] and Structure-from-Motion (SfM) [25]. This however conflates the performance of both steps. While deep learning has greatly enhanced the robustness of descriptors, keypoint detection has not improved at the same rate: classical algorithms, like SIFT [35], remain highly competitive, especially in terms of orientation invariance and localization accuracy [44, 48]. One reason for this disparity is the difficulty in obtaining high-quality supervision for keypoints, which are poorly defined, compared to the relative ease in obtaining ground-truth correspondences [14, 31]. Furthermore, recent improvements in dense feature matching [17, 19, 57] have made feature descriptors less important. Keypoints, however, remain critical to create discrete multi-view tracks for SfM and scale to large scenes. This thus results in new requirements for keypoint detection: keypoints should be highly precise and repeatable, but also agnostic to the matching that they are paired with. Consequently, we evaluate keypoints in isolation and introduce a keypoint detector that is tailored for the challenges of the deep learning era, illustrated in Fig. 1, by exhibiting the following key qualities:

Rotation Robustness: Simple in-plane rotations of images can cause both detections and correspondences to break catastrophically [9, 44]. This has long been overlooked because such cases were not covered in existing datasets, but has recently gained increased attention [9, 29, 44]. We show that careful rotation augmentations during training suffices to obtain competitive rotational detection robustness.

Keypoint Scoring: For compute-constrained settings like edge devices, it is crucial to subsample keypoints to reduce memory and runtime in downstream tasks. We claim that the inherent ranking of existing keypoint detectors via their confidence [15, 58, 67] is suboptimal because it ignores the spatial distribution and matchability of points. We propose to train a plug-and-play ranking head that maximizes the number of matches at different keypoint budgets and is compatible with all state-of-the-art keypoint detectors.

Spatial Uncertainty: Detections are subject to noise [21], but their spatial covariance is rarely studied. However, estimating this is crucial for error propagation in downstream tasks, *e.g.*, bundle adjustment in SfM [1, 50]. Previous works quantified spatial uncertainty either through an up-to-scale anisotropic covariance [56, 64] or a spatial confidence score [45]. In contrast, we propose to learn a metric, anisotropic covariance estimator via homographic adaptation that can be used for end-to-end uncertainty propagation from keypoints to pose.

To summarize, in this work, we present 1) an isolated keypoint evaluation strategy that reflects modern requirements and challenges, 2) introduce a competitive keypoint detector, named *RaCo*, trained with reinforcement learning on synthetic homographies only, and finally 3) propose a simple yet effective strategy to estimate the keypoint ranking and its metric covariance without additional labels.

2. Related Work

Decoupling keypoints and descriptors was predominant in the era of handcrafted local-features [6, 35, 43], which followed a `detect-then-describe` paradigm. There, interest points are first detected on corners [23, 35, 52] and then robust descriptors are extracted for each keypoint [35]. Early works on learned local features adopted this approach [5, 36, 63]. Later works proposed to couple both tasks under the `detect-and-describe` approach, where keypoints and descriptors are learned end-to-end in one network [15, 16, 42, 66, 67]. Recently, Li et al. [30] questioned this design decision and observed that weak descriptors can affect detection accuracy. Consequently, recent works decouple [30] both tasks, use completely separate networks for each [11, 18, 40, 44] or focus exclusively on detection [20] or description [36, 60, 61]. Furthermore, recent success in feature matching [17, 19, 32, 47, 54] reduced or alleviated the dependency on descriptors [17, 19], but keypoints remain necessary for large-scale applications and classic SfM [50]. Hence, we propose to train a lightweight detector independently and perform an unbiased evaluation of said detections using geometric correspondences.

Learned keypoint detection has widely replaced and outperformed hand-crafted methods over the past years [12, 26]. A pioneering work is SuperPoint [15], which pre-trained on projections of synthetic shapes with clear corners, and finetuned with homography adaptation on random images to close the synthetic-to-real domain gap. Other works also couple keypoint detection and match success on more general image pairs [16, 42, 67], where MegaDepth [31] emerged as the de-facto standard training collection despite the lack of data diversity (phototourism). Some works address this limitation via differentiable pose estimation on posed images [7] or arbitrary image pairs [28]. DeDoDe [18] learns tracks from large-scale SfM, while DISK [58] used descriptors and ground truth depth or epipolar constraints to calculate reward values. S-TREK [44] improves over DISK by replacing patch-based sampling, subject to border artifacts, with sequential sampling, and uses equivariant convolutions [10, 29, 62] to increase rotational robustness. In contrast, we show that state-of-the-art rotational stability can be achieved with effective data augmentations, and even without the added complexity and cost of such equivariant architectures. RDD [11] and DeDoDe [18] decouple detection and description, while Edstedt et al. [20] drop the dependence on descriptors in approaches trained with a policy gradient, and identify that light and dark detectors emerge. We also train our detector with a policy gradient, but on challenging homography adaptations of real images like in [13, 59], but without any pretraining like SuperPoint [15].

Keypoint uncertainty is rarely studied in the literature. In practice, we are interested in uncertainty to i) filter keypoints which are unrepeatable, e.g. in the sky, ii) rank and sub-sample keypoints for increased computational efficiency and iii) propagating the spatial uncertainty to downstream algorithms. Most networks output a keypoint score, which measures the model's confidence that a pixel is a keypoint [15, 42, 58, 67]. This score often suffices to filter bad keypoints [67], but is suboptimal to rank keypoints because it ignores their spatial distribution and localization error, both important for accurate pose estimation [50]. Consequently, benchmarks often use a fixed number of keypoints [15, 58, 67], while keypoint subsampling, a critical hyperparameter in practice [46], is rarely ablated [22]. The spatial uncertainty of keypoints is studied by Muhle *et al.* [37] using differentiable relative pose estimation. DAC [56] proposes two post-hoc covariance estimates which are up-to-scale, derived from the score map of local feature extractors [64]. UAPoint [65] models aleatoric and epistemic uncertainty [27] during training, which improves repeatability and matchability. Santellani *et al.* [45] analyze the spatial variance of keypoints under different image augmentations in a gaussian mixture model (GMM) to both refine and score keypoints. We study i) differentiable keypoint ranking to

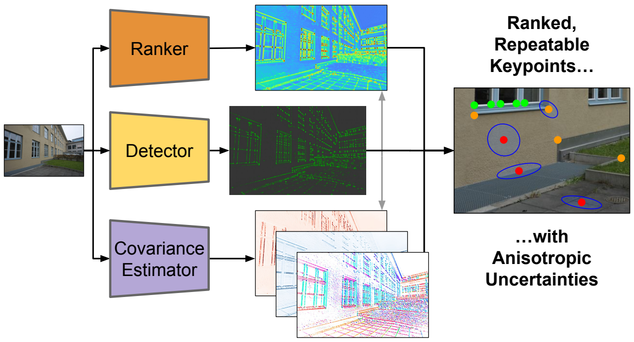

Figure 2. **Overview.** Our method consists of three branches: i) A detector head that produces a scoremap with repeatable keypoints, and ii) a covariance head that outputs the 2D spatial uncertainty in pixels, both sharing a lightweight backbone. The iii) ranker module outputs soft keypoint scores which maximize the repeatability at different keypoint budgets.

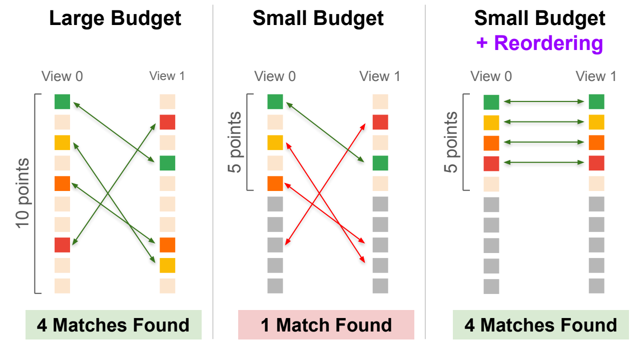

Figure 3. **Keypoint ranking.** Inconsistent keypoint ranking between images (left) results in excessive match filtering when the amount of keypoints per image is restricted (*small budget*, middle). Our ranking module keeps repeatable points at the top of the list and yields similar ranks for corresponding points (right).

maximize repeatability after filtering, and ii) predicting metric, spatial covariance of our detections from homography adaptation, and their impact on downstream tasks.

3. Method

We first provide an overview of our method, shown in Fig. 2 and discussed in Sec. 3.1, illustrating our main components: a **detector** (Sec. 3.2) to find repeatable keypoints, a differentiable **ranker** (Sec. 3.3) to maximize repeatability at smaller keypoint budgets, and a **covariance estimator** (Sec. 3.4) to quantify their spatial uncertainty.

3.1. Overview

Training Setup: RaCois trained on arbitrary image collections without any labels or auxiliary data. In particular, we simulate two-view matching by sampling two crops from an image using synthetic homographies and strong photometric augmentations [15, 47]. Let $\boldsymbol{I}_A, \boldsymbol{I}_B \in \mathbb{R}^{H\times W\times 3}$ be two views of an image related by a known ground-truth homography $\mathbf{H}_{A\rightarrow B}$. The i^{th} keypoint detected in view $v \in \{A, B\}$ is denoted by $\mathbf{x}_v^i$ or simply $\mathbf{x}^i$ and the set of detected keypoints by $\mathbf{x}_v \in \mathbb{R}^{N\times 2}$.

Detector: The detector module identifies keypoints that are repeatable, *i.e.*, that can be reliably and accurately detected from multiple views and varying appearance conditions, usually located on corners or blobs. Repeatability is a critical requirement for robust feature matching. Given an input image $\boldsymbol{I}$, the detector estimates a heatmap $\boldsymbol{P} \in \mathbb{R}^{H\times W}$ that represents, for each pixel, the probability to be selected as keypoint. We select keypoints $\mathbf{x}_v$ as local maxima in this probability score map with non-maxima suppression (NMS), which prevents clustering of points [15, 42, 58, 67].

Prior studies [29, 44] have highlighted the lack of rotational equivariance in modern deep keypoint detectors and proposed architectural remedies, usually based on computationally expensive equivariant architectures [10, 62]. We instead demonstrate that extensive data augmentation during training is sufficient to effectively enforce rotation equivariance. This approach enables our model to achieve strong robustness to image rotations while preserving a lightweight and efficient design. Specifically, we generate training pairs using synthetic homographies with full 360° rotations combined with strong photometric transformations.

Ranker: Modern, learned keypoint detectors typically order keypoints based on their detection score [15, 58], which indicates the likelihood of a keypoint's presence. However, we want this ordering to maintain a maximum of matches, which requires awareness of the keypoints spatial distribution and matchability in each image. Patch-based detectors [15, 58], in particular, tend to ignore this in their scores. This can lead to a significant loss in matches and repeatability when the keypoint budget is limited, that is, the amount of sampled and retained keypoints $n \leq N$ is small. Fig. 3 demonstrates this problem: when considering fewer keypoints than trained for, a suboptimal ordering of the keypoints, where corresponding keypoints have vastly different ranks, leads to significantly fewer correspondences.

Our ranking module addresses this by providing an alternative ordering specifically designed to maximize matching performance over varying keypoint budgets. This outputs a separate ranking score map $\boldsymbol{R} \in \mathbb{R}^{H\times W}$, from which we obtain a ranking score r^i for each keypoint $\mathbf{x}^i$.

Our goal is to learn ranking scores for views A and B, such that, when ordering the keypoints in descending order of ranking scores, the number of matches is maximized over *all* keypoint budgets $n \in \{1, \ldots, N\}$, where N is the total number of detected keypoints. Let $\mathcal{R}^n(\mathbf{r}_v) \in \mathbb{N}^n$ be the ordered set of n keypoint indices in view v with the highest ranking scores $\mathbf{r}_v^i$. The ranking problem can then be expressed as finding the optimal ranking scores $\mathbf{r}_A, \mathbf{r}_B$ that maximize

$$\max_{\mathbf{r}_A,\mathbf{r}_B} \sum_{n=1}^{N} |\mathbf{M}(\mathcal{R}^n(\mathbf{r}_A), \mathcal{R}^n(\mathbf{r}_B))| \ . \tag{1}$$

Here, $\mathbf{M} : \mathbb{N}^n \times \mathbb{N}^n \rightarrow \mathbb{N}^{k\times 2} \subseteq \mathcal{M}_{A\rightarrow B}$ extracts the subset of all ground-truth matches $\mathcal{M}_{A\rightarrow B}$ between the ranked and truncated keypoints in both views, and $|\mathbf{M}| = k$:

$$\mathbf{M}(\mathbf{x}, \mathbf{y}) = \{(i, j) \ \forall \ i, j \in \mathbf{x}, \mathbf{y} \text{ if } (i, j) \in \mathcal{M}_{A\rightarrow B}\} \tag{2}$$

Covariance Estimator: Accurately modeling the spatial uncertainty of detected keypoints is critical for robust 3D vision. While largely overlooked in recent years, propagating this uncertainty is necessary to obtain reliable covariance estimates for downstream tasks such as triangulation and pose estimation. Keypoints exhibit localization errors due to detector inaccuracies, image noise, and discretization effects, leading to reprojection errors across views. We characterize this error by estimating the 2D spatial uncertainty of keypoints in metric scale (pixels), providing a measure of confidence and local uniqueness of the keypoint. Concretely, we want to estimate the symmetric, positive definite covariance matrix for each pixel, $\mathbf{\Sigma} \in \mathbb{R}^{H\times W\times 2\times 2}$.

3.2. Keypoint Detector

Following recent work in self-supervised keypoint learning [20, 44], we adopt a *policy-gradient* [55] approach to train a detector that produces repeatable keypoints.

The detector takes a normalized RGB image $\boldsymbol{I}$ and outputs a keypoint score map $\boldsymbol{S} \in \mathbb{R}^{H\times W}$. This score map is normalized with a global softmax operation over the whole image to obtain a probability score map $\boldsymbol{P} \in \mathbb{R}^{H\times W}$. Keypoints are selected by applying Non-Maximum Suppression (NMS) [15, 58, 67] followed by top-N selection on this score map, which is a standard technique for keypoint sampling.

The training objective maximizes a reward signal which encourages repeatable keypoints to be detected in $\boldsymbol{I}_A$ and $\boldsymbol{I}_B$. Following [20, 44], the reward directly maximizes repeatability. For a keypoint $\mathbf{x}^i_A$, it is defined as $\rho(\mathbf{x}^i_A) = \{\rho_{\text{pos}} \text{ if } d(\mathbf{x}^i_A) \leq d_{\max}; \quad \rho(\mathbf{x}^i_A) = \rho_{\text{neg}} \text{ otherwise}\}$, and ρ_{pos} and ρ_{neg} are the positive and negative reward, respectively. $d(\mathbf{x}^i_A)$ is the distance between the reprojected keypoint $\mathbf{H}_{A\rightarrow B}(\mathbf{x}^i_A)$ and its closest neighbor in view B, and $d_{\max}$ is a predefined radius for a successful match.

We minimize the negative log-likelihood of the sampled keypoints, weighted by their normalized reward

$$\mathcal{L}_{\text{detector}} = -\sum_{v\in\{A,B\}} \sum_{i=1}^{K} \rho'(\mathbf{x}^i_v) \ \log p^i_v \ , \tag{3}$$

where $\mathbf{x}^i_v$ is the i-th sampled keypoint from view $v \in \{A, B\}$, the term $p^i_v = \boldsymbol{P}_v[\mathbf{x}^i_v]$, where $[\cdot]$ is the lookup operator, and p^i_v is the sampling probability of keypoint $\mathbf{x}^i_v$ according to the probability score map $\boldsymbol{P}_v$. The normalized reward $\rho(\mathbf{x}^i_v)$ is calculated following DaD [20] as $\rho'(\mathbf{x}^i_v) = \frac{\rho(\mathbf{x}^i_v)}{\mathbb{E}_v[\rho(\mathbf{x}^i_{\mathbf{v}})]+\epsilon}$, where $\rho(\mathbf{x}^i_v)$ is the un-normalized reward for that keypoint and $\mathbb{E}_v[\rho(\mathbf{x}^i_v)]$ is the average reward over all keypoints in view v.

3.3. Ranker

The discrete ranking objective in Eq. (1) is non-differentiable. We use a differentiable approximation [8] to supervise our network via soft ranks. Let $h_{\text{soft}} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ denote the *soft ranking* operator, which maps a vector of ranking scores $\mathbf{r} = [r^1, \ldots, r^n]$ to their differentiable ranks $\mathbf{r}_{\text{soft}} = h_{\text{soft}}(\mathbf{r})$. We train the ranker module in a self-supervised manner using two loss terms.

Spearman Loss: Maximizing Spearman's rank correlation coefficient [53] of our ordered keypoints encourages corresponding keypoints (matches) to have similar ranks within their respective lists. Maximizing this metric can be accomplished by minimizing the Euclidean distance between their soft ranks [8]. We compute the soft ranks for the subset of matched keypoints in both views, denoted by the vectors $\mathbf{r}^{\text{matched}}_A$ and $\mathbf{r}^{\text{matched}}_B$. The loss is then simply:

$$\mathcal{L}_{\text{spearman}} = \frac{1}{N} \sum_{i=1}^{N} \left(h_{\text{soft}}(\mathbf{r}^{\text{matched}}_{A,i}) - h_{\text{soft}}(\mathbf{r}^{\text{matched}}_{B,i})\right)^2 \tag{4}$$

By minimizing this loss, we are encouraging that corresponding keypoints have similar ranks in both views. This ensures that either both keypoints are present in the keypoint list or none are truncating.

Pull Loss: In Fig. 3 it is evident that the ranker needs to place the matched keypoints at the beginning of the list, and the unmatched points ranked at the end of the list. The *pull* loss encourages this placement of keypoints by pulling the matched points towards the first rank (#1) and the unmatched points towards the last rank (#N). For each keypoint $\mathbf{x}^i_v$ with soft rank $h_{\text{soft}}(r^i_v)$, the per-keypoint loss is:

$$\mathcal{L}^i_{\text{pull}} = \begin{cases} |h_{\text{soft}}(r^i_v) - 1| & \text{if } \mathbf{x}^i_v \text{ is matched} \\ |h_{\text{soft}}(r^i_v) - N| & \text{otherwise} \end{cases} \tag{5}$$

The final ranker loss is a weighted combination of these two terms: $\mathcal{L}_{\text{ranker}} = \mathcal{L}_{\text{spearman}} + \lambda_{\text{ranker}} \cdot \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}^i_{\text{pull}}$

3.4. Covariance Estimator

We treat each detected keypoint $\mathbf{x}^i_v$ as a noisy observation with associated uncertainty $\mathbf{\Sigma}^i_v$. The reprojection error $\mathbf{e}^i_{B\rightarrow A}$ between corresponding keypoints accounts for uncertainties from both views: $\mathbf{e}^i_{B\rightarrow A} = \mathbf{x}^i_A - \mathbf{H}_{B\rightarrow A}(\mathbf{x}^i_B) \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}^i_{\text{error}})$, where the combined error covariance is:

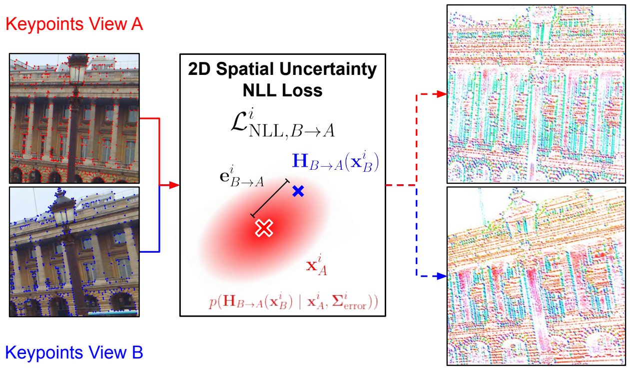

Figure 4. **Covariance supervision.** We train our covariance estimator by maximizing the log-likelihood of the reprojection error between corresponding keypoints. For corresponding keypoints $\mathbf{x}^i_A$ in view A and $\mathbf{x}^i_B$ in view B, the reprojection error $\mathbf{e}^i_{B\to A} = \mathbf{H}_{B\to A}(\mathbf{x}^i_B) - \mathbf{x}^i_A$ is modeled as a zero-mean Gaussian: $\mathbf{e}^i_{B\to A} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}^i_A + \mathbf{J}^i_{B\to A}\boldsymbol{\Sigma}^i_B(\mathbf{J}^i_{B\to A})^\top)$, where $\mathbf{J}^i_{B\to A}$ is the Jacobian of the homography evaluated at $\mathbf{x}^i_B$. The resulting covariance estimates (right) are strongly anisotropic (colored by the angle of the covariance's first eigenvector) and are large in areas with low texture (illustrated by opacity).

$\boldsymbol{\Sigma}^i_{\mathrm{error}} = \boldsymbol{\Sigma}^i_A + \mathbf{J}^i_{B\to A}\boldsymbol{\Sigma}^i_B(\mathbf{J}^i_{B\to A})^\top$, and $\mathbf{J}^i_{B\to A}$ is the Jacobian of the homography transformation evaluated at $\mathbf{x}^i_B$, propagating the uncertainty from view B to view A.

Rather than predicting $\boldsymbol{\Sigma}^i_v$ directly, the network outputs the three non-zero elements of its Cholesky decomposition $\mathbf{L}^i_v$, where $\boldsymbol{\Sigma}^i_v = \mathbf{L}^i_v(\mathbf{L}^i_v)^\top$. This parameterization guarantees symmetry and positive semi-definiteness. To ensure strict positive definiteness, the diagonal entries of $\mathbf{L}^i_v$ are passed through a Softplus activation. The network produces three scoremaps corresponding to the entries of $\mathbf{L}$, from which $\mathbf{L}^i_v$ is sampled at each detected keypoint location.

We compute the negative log-likelihood (NLL) of the reprojection error as

$$\begin{aligned}\mathcal{L}^i_{\mathrm{NLL},B\to A} =& \frac{1}{2}\log\det(\boldsymbol{\Sigma}^i_{\mathrm{error}}) \\ &+ \frac{1}{2}(\mathbf{e}^i_{B\to A})^\top(\boldsymbol{\Sigma}^i_{\mathrm{error}})^{-1}\mathbf{e}^i_{B\to A} \quad ,\end{aligned} \tag{6}$$

where the matrix inverse and determinant are computed efficiently via Cholesky decomposition of $\boldsymbol{\Sigma}^i_{\mathrm{error}}$. We also apply the same loss in the reverse direction ($A \to B$) and define the total covariance loss as the average bidirectional NLL over all $|\mathbf{M}|$ matched keypoints:

$$\mathcal{L}_{\mathrm{covariance}} = \frac{1}{2|\mathbf{M}|}\sum_{i=1}^{|\mathbf{M}|}\left(\mathcal{L}^i_{\mathrm{NLL},B\to A} + \mathcal{L}^i_{\mathrm{NLL},A\to B}\right). \tag{7}$$

4. Experiments

We first report important implementation details in Sec. 4.1. In Sec. 4.2, we study the detected keypoints on four two-view estimation benchmarks, followed by detailed evaluations on rotational equivariance in Sec. 4.3. Finally, we ablate the proposed keypoint ranking and covariance estimation modules in Sec. 4.4 and Sec. 4.5, respectively.

4.1. Implementation

The architecture of RaCo is based on that of ALIKED-N(16) [67], which is lightweight, multi-scale, and sufficiently expressive. It predicts both the detection heatmap and the pixelwise covariance. The ranker is a separate ResNet [24] backbone which takes as input the normalized RGB image and outputs the ranker score map $\boldsymbol{R}$. We first train the backbone and detection head, and later train the covariance head and the ranker separately in a second stage. The model is trained on synthetic pairs sampled from the Oxford-Paris 1M distractors dataset [41]. It is trained using the AdamW [34] optimizer in PyTorch [39] with an NVIDIA 2080 Ti GPU. The detector head is trained using 400k samples center cropped at 768×768 and then resized to 640×640 after sampling the homographies, for 1 epoch with a batch size of 2 with an initial learning rate of 2×10^{-4} decaying according to a cosine annealing schedule to a terminal learning rate of 10^{-6}. We sample 512 keypoints per image with an NMS radius of 3px following the training sampling of [20].

For the detector we set $d_{\max}$=1.2px, ρ_{pos}=1, and $\rho_{\mathrm{neg}} = -\min\{10^{-2}, t\cdot 10^{-6}\}$ where t is the number of optimizer steps taken through training. At inference time we use subpixel sampling based on the soft-argmax over the patch around the selected keypoint [20, 67].

For the training of the other two modules, the inference setting of the detector is used to capture the distribution of keypoints at inference. The covariance estimator head is trained for 20k steps and the ranker module is separately trained for 1 epoch with the identical setup as the detector training. For both training runs we freeze all model layers except those of the pertinent module/head. More details are provided in the supplementary.

4.2. Two-View Keypoint Matching

Setup: We present a comprehensive evaluation of our keypoint detector in the two-view setting. We use 4 different evaluation datasets and use the ground truth transformations to project keypoints across views. Keypoint matching is subsequently performed by identifying mutual nearest neighbors within a specified reprojection radius in both views. We force all detectors to detect the same number of keypoints.

HPatches [4] comprises over 500 real-world image pairs under a homography transformation. The views are subject to either illumination or viewpoint changes. DNIM [68] consists of 1722 images grouped into 17 sequences per webcam. This data set exhibits strong illumination changes, and we sample 428 random image pairs augmented with random homographies to also evaluate the robustness to perspective changes. MegaDepth [31] is a large-scale dataset of photo-tourism internet images. We use the subset MegaDepth1800 [32] of 4 scenes from the dataset's test set. We introduce the ETH3D-Two-View dataset, covisible image pairs sampled from indoor and outdoor scenes in ETH3D [51]. ETH3D and MegaDepth provide ground truth camera poses, intrinsic parameters, and depth images, using which we can map points across images.

Baselines: We compare our model against SIFT [35] and several learned keypoint detectors. SuperPoint [15] is trained on homographies in a supervised way to explicitly detect corners. ALIKED-N(16) [67] is trained in an unsupervised manner using depth and homography data. DISK [58] and DaD [20] are both trained on depth data. We report the number of ground-truth matches within a reprojection threshold, the fraction of repeatable points within two thresholds and the localization error in pixels. We also estimate homographies ($\mathbf{H}$) or relative poses ($\mathbf{T}$) and report the Area Under the recall Curve (AUC).

Results: Tab. 1 and Tab. 2 summarize the results of the matching evaluation on both types of datasets. On all datasets, our model obtains the highest repetability at 3px. Our model is competitive with ALIKED [67] and DaD [20], which are trained with depth supervision, on relative pose estimation and our localization error is slightly higher than DaD [20] and equal to ALIKED [67] on ETH3D. DISK [58], which is trained on MegaDepth [31], exhibits strong performance in terms of number of matches, however, at the cost of repeatability and weaker generalization to ETH3D or HPatches, which have a slight domain gap compared to MegaDepth. SuperPoint [15], despite not having competitive repeatability scores, is still able to estimate the relative pose very well, which shows that corners are high quality keypoints and that depth data is not necessarily required to train a good detector. On DNIM [68], RaCo outperforms all other methods, showing the model's robustness to both illumination and perspective changes.

	detector	#matches @3px	rep. [%] 1px	rep. [%] 3px	loc. [px]	AUC $\mathbf{H}$ 1px	AUC $\mathbf{H}$ 3px
HPatches [4]	SIFT [33]	282	24.8	44.0	0.99	32.7	68.8
	SuperPoint [15]	503	22.1	54.0	1.19	41.5	74.7
	DISK [58]	484	23.9	49.9	1.05	27.5	63.7
	ALIKED [67]	530	32.8	56.3	0.96	34.1	69.5
	DaD [20]	510	30.2	55.2	1.01	40.0	73.5
	Ours	544	33.9	58.5	0.98	41.2	74.2
DNIM [68]	SuperPoint [15]	56	11.9	30.2	1.26	2.4	23.1
	ALIKED [67]	57	11.6	26.6	1.13	1.7	18.0
	DaD [20]	58	13.9	29.8	1.16	1.9	23.2
	Ours	72	20.4	35.8	1.00	4.4	25.9

Table 1. **Homography estimation.** We evaluate the repeatability and correspondences of keypoints paired with ground-truth matches on HPatches [4] and DNIM [68]. Our method achieves competitive repeatability and pose estimation performance on HPatches, and exhibits superior robustness on image pairs from DNIM with large illumination changes. We color the best and second best results for each metric are colored.

	detector	#matches @3px	rep. [%] 3px	rep. [%] 5px	loc. [px]	AUC $\mathbf{T}$ 5°	AUC $\mathbf{T}$ 10°
MD1800 [32]	SIFT [33]	317	34.8	39.0	1.47	66.0	78.7
	SuperPoint [15]	549	42.5	46.8	1.61	71.1	82.3
	DISK [58]	647	41.0	43.7	1.39	67.9	80.2
	ALIKED [67]	579	45.9	48.6	1.30	71.2	82.5
	DaD [20]	588	45.9	49.9	1.51	72.4	83.3
	Ours	595	46.1	49.9	1.44	71.8	82.8
ETH3D [51]	SIFT [33]	298	34.6	41.5	1.62	85.2	90.2
	SuperPoint [15]	498	42.6	47.0	1.57	92.9	96.1
	DISK [58]	559	43.1	48.2	1.50	85.0	89.6
	ALIKED [67]	523	43.9	47.7	1.38	90.3	94.0
	DaD [20]	534	46.5	50.1	1.21	94.5	96.8
	Ours	562	47.2	52.7	1.37	92.5	95.6

Table 2. **Relative pose estimation.** We evaluate two-view matching performance and keypoint repeatability on two datasets: MegaDepth1800 [31, 32] and ETH3D-Two-View [51]. The colors indicate the best and second best results for each metric. On both benchmarks, our method achieves the highest repeatability despite only being trained on homographies.

4.3. Rotation Equivariance

Setup: Following [43], we evaluate the rotation equivariance of keypoint detectors using in-plane rotations. We take the first 20 images from HPatches [4] and take the largest square crop that can be rotated through 360°. We rotate this original view in increments of 10°, and resize it to 512×512. We add zero-mean Gaussian noise ($\sigma = 10$, pixel intensity scale $[0, 255]$) to each image pair to suppress interpolation artifacts. We run the keypoint detectors on this image pair and extract 200 keypoints per view.

Baselines: We evaluate ALIKED-N(16) [67] (trained with and without extra rotation augmentation), DaD [20], SuperPoint [15], DISK [58], SIFT [35], and our model. We report the repeatability at thresholds of $\{1, 2, 3\}$ px. We summarize the rotation equivariance as the area under the repeatability vs rotation angle curve (rotation AUC) after normalizing the angles to the range $[0, 1]$. We also report the average runtime of each detector.

Results: Fig. 5 illustrates the repeatability at various rotation angles and Tab. 3 reports the AUC of the repeatability up to different thresholds. Our model retains the highest repeatability by a large margin when the image pair is subject to in plane rotation. We achieve a consistent repeatability around

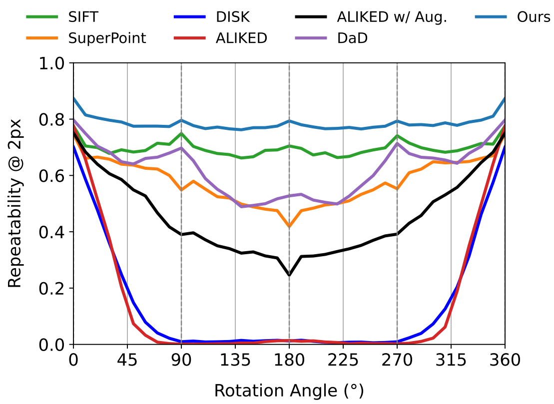

Figure 5. **Rotation evaluation on HPatches [4].** We plot the repeatability@2px over the rotation angle between image pairs. SIFT [35] is more robust than any learned keypoint detectors, but our improved rotation augmentations result in state-of-the-art rotational robustness without requiring specialized model architectures.

detector	repeatability AUC [%]			run time [ms]
	1px	2px	3px	
SIFT [35]	52.3	69.6	71.9	34.8
SuperPoint [15]	25.3	57.7	67.7	6.8
DISK [58]	7.3	13.0	18.1	31.6
ALIKED [67]	8.4	12.8	16.7	5.1
ALIKED w/ Aug. [67]	24.9	44.8	49.7	5.1
DaD [20]	27.1	62.0	70.0	15.2
Ours	71.5	78.3	79.5	4.8
Ours w/ ReCONV[10]	74.8	81.9	83.1	42.3
Ours w/o Aug.	13.8	24.9	33.1	4.8

Table 3. **Rotation ablation and run time.** We report the area under the repeatability curve up to different error thresholds under in-plane rotations up to 360°. The upper section compares our model against state-of-the-art keypoint detectors and the lower section ablates key design decisions like rotation augmentations and equivariant convolutions. We color the best and second best results for each metric. Our method significantly outperforms state-of-the-art models because of strong rotation augmentations, even without expensive equivariant convolutions.

80% over all angles while other detectors with the exception of SIFT exhibit a degradation in repeatability. In contrast to DaD [20], which uses rotation augmentations at 90° intervals, we sample rotations over the entire circle, resulting in more stable and generally increased equivariance. Training RaCo with rotationally equivariant convolutions [10] results in 3% increased repeatability, but is 10× slower at inference and 3.5× slower to train, consuming 2.5× more memory. Removing rotation augmentations drastically impairs the accuracy of the models, demonstrating the importance of smooth rotation augmentations.

4.4. Keypoint Ranking

Setup: We consider the setup of the matching evaluation on HPatches [4] and MegaDepth1800 [32] from Sec. 4.2. We vary the number of keypoints extracted per view (keypoint budget), and compute the repeatability at that budget. We provide more details in the supplementary material.

Baselines: We compare our model in two settings, the first when the keypoint probability scores from the detector score map $\boldsymbol{P}$ are used to order the points, and secondly when the ranking scores from $\boldsymbol{R}$ are used to order (+*Ranker*). To demonstrate that our ranker model is applicable to any detector, we retrain the ranker module with SuperPoint's keypoints and include it in the comparison.

Results: For both our model and SuperPoint, ordering and truncating keypoints according to our ranking score provides a significant boost to the repeatability at all keypoint budgets on both datasets, see Fig. 6. Note how the terminal repeatability is identical for models with and without the ranker, because it is only performing a reordering operation. We observed that keypoint detectors that select keypoints over a grid such as SuperPoint [15] or DISK [58] suffer more from a bad ordering of points. Especially there, our ranker is able to recover repeatability at restricted budgets by ranking points over the entire image. Keypoint detectors that produce globally normalized heatmaps [20, 44, 67] (also our method) are typically trained with some sort of top-K selection, and hence, an implicit global ordering. At train time, they learn to prioritize matchable keypoints, similar to our ranker. However, they do not enforce the correlation between matched points. This being a different objective than detection probabilities motivates our dedicated ranker module, and results in further improved truncation robustness.

4.5. Multiview Triangulation

Setup: We evaluate the covariances for the task of 3D triangulation on the ETH3D dataset [51]. We reproject keypoints using the ground truth depth and poses across views to form matches, and subsequently triangulate them to multi-view tracks [46, 50]. We then refine the 3D points with non-linear least squares optimization, implemented with PyCeres and COLMAP [3, 50], in which reprojection error residuals are weighted by the estimated covariances. Post-convergence, we compute 3D marginal covariances, defining precision as the reciprocal of the covariance ellipsoid volume. Points are sorted by precision and pruned from lowest to highest to reach the target size. This filters for high triangulation stability relative to input 2D uncertainties.

Baselines: We compare our metric 2D spatial uncertainties against the following baselines. Detector Agnostic Covariances: DAC [56] provides isometric (**DAC-iso**) and full (**DAC-full**) covariances up to scale by operating on the de-

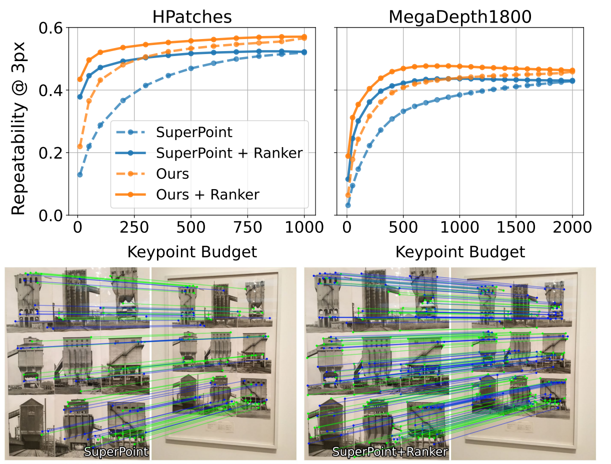

Figure 6. **Keypoint ranking evaluation.** We evaluate the repeatability at 3px against the number of keypoints per image on HPatches [4] and MegaDepth1800 [32] (top). For both SuperPoint [15] and RaCo, our ranker module is more effective in ordering keypoints than the original detection scores. We also visualize the repeatable points as matches at a budget of 128 (●) and 256 (●) keypoints (bottom). SuperPoint+Ranker (right) almost doubles repeatable points here over SuperPoint [15] scores (bottom left).

tector score maps. The next baseline involves assigning a **Constant** isotropic covariance to every keypoint in the image. Finally we weight the keypoints by the reprojection error (**Reproj. Err.**) from the 3D reconstruction.

Results: Fig. 7 shows that across all point cloud sizes, our metric 2D covariances consistently attains the highest accuracy and completeness. Interestingly, assuming a **Constant** isotropic keypoint covariance remains surprisingly competitive, likely because of large track lengths. **DAC-full**'s anisotropic covariances perform better than **DAC-iso**, but the lack of scale in both methods limit the informativeness of their covariances in this multi-view setting. Our method achieves highest performance even at 99%, suggesting that our estimated 2D covariances also increase the accuracy of 3D points by downweighting noisy observations.

4.6. Covariance Metric Consistency

Setup: Following Sec. 4.5, we bin points by the square root of the trace of their predicted marginal covariances into 20 equally populated bins. We plot the mean predicted uncertainty against the mean observed Euclidean distance to the ground truth mesh for each bin.

Baselines: We evaluate the same baselines as in Sec. 4.5. We compute the slope β via log-log linear regression, where a slope of $\beta = 1$ indicates perfect metric consistency.

Results: Fig. 8 shows our method ($\beta = 0.94$) which learns anisotropic 2D uncertainties of each keypoint achieves the closest agreement to the ideal unit slope, demonstrating that

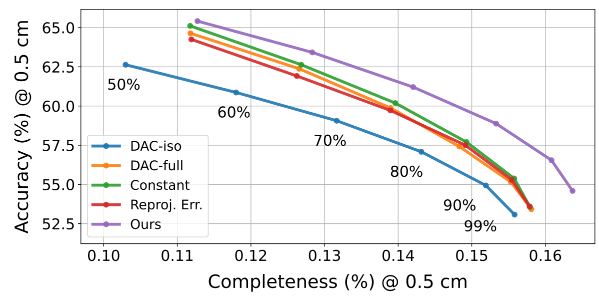

Figure 7. **3D triangulation on ETH3D [51].** We plot triangulated point cloud accuracy against completeness (within 0.5cm). Each point represents the fraction of the original cloud retained after filtering by spatial uncertainty metrics. Our 2D covariances i) improve both accuracy and completeness even without filtering, and ii) yield superior 3D covariance estimates for more effective filtering.

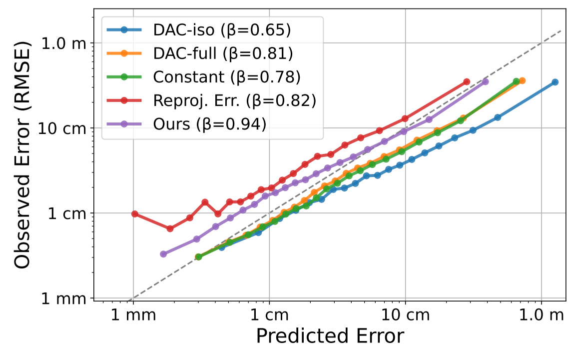

Figure 8. **Covariance Calibration.** We plot observed error against the ground truth vs. predicted uncertainty ($\sqrt{\mathrm{Tr}(\Sigma)}$). Our method is the closest to ideal calibration line, shown as the dashed gray line.

our covariances preserve a physical metric scale.

5. Conclusion

We introduce RaCo, a lightweight neural network trained on perspective image crops only, addressing several challenges in keypoint detection. Our method integrates a repeatable interest point detector with a differentiable ranker and a metric covariance estimator. A ranker, trained to maximize repeatability, efficiently identifies the most valuable keypoints. The covariance estimator provides a metric-scale measure of spatial uncertainty, valuable for downstream tasks. Our approach achieves strong rotational robustness through a simple data augmentation strategy, and experimental results validate our model's effectiveness in keypoint repeatability and two-view matching, particularly in scenes with large in-plane rotations. Ultimately, RaCo provides a simple yet effective strategy to detect robust interest points, rank keypoints, and quantify their metric covariance, making it a valuable building block for various computer vision systems.

References

[1] Sameer Agarwal, Noah Snavely, Steven M Seitz, and Richard Szeliski. Bundle adjustment in the large. In *ECCV*, 2010. 2

[2] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building Rome in a day. *Communications of the ACM*, 54 (10):105–112, 2011. 1

[3] Sameer Agarwal, Keir Mierle, and Others. Ceres solver. `http://ceres-solver.org`, 2025. 7

[4] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *CVPR*, 2017. 5, 6, 7, 8

[5] Axel Barroso-Laguna, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Key.Net: Keypoint Detection by Handcrafted and Learned CNN Filters. In *ICCV*, 2019. 2

[6] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. In *ECCV*, 2006. 2

[7] Aritra Bhowmik, Stefan Gumhold, Carsten Rother, and Eric Brachmann. Reinforced feature points: Optimizing feature detection and description for a high-level task. In *CVPR*, 2020. 2

[8] Mathieu Blondel, Olivier Teboul, Quentin Berthet, and Josip Djolonga. Fast differentiable sorting and ranking. In *ICML*, 2020. 4

[9] Georg Bökman, Johan Edstedt, Michael Felsberg, and Fredrik Kahl. Steerers: A framework for rotation equivariant keypoint descriptors. In *CVPR*, 2024. 1, 2

[10] Gabriele Cesa, Leon Lang, and Maurice Weiler. A program to build E (N)-equivariant steerable CNNs. In *ICLR*, 2022. 2, 3, 7

[11] Gonglin Chen, Tianwen Fu, Haiwei Chen, Wenbin Teng, Hanyuan Xiao, and Yajie Zhao. RDD: Robust Feature Detector and Descriptor using Deformable Transformer. In *CVPR*, 2025. 2

[12] Ashley Chow, Eduard Trulls, HCL-Jevster, Kwang Moo Yi, lcmrll, old ufo, Sohier Dane, tanjigou, WastedCode, and Weiwei Sun. Image matching challenge 2023, 2023. 2

[13] Peter Hviid Christiansen, Mikkel Fly Kragh, Yury Brodskiy, and Henrik Karstoft. UnsuperPoint: End-to-end Unsupervised Interest Point Detector and Descriptor. *arXiv:1907.04011*, 2019. 2

[14] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *CVPR*, 2017. 1

[15] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-Supervised Interest Point Detection and Description. In *CVPR Workshop on Deep Learning for Visual SLAM*, 2018. 1, 2, 3, 4, 6, 7, 8

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

[17] Johan Edstedt, Ioannis Athanasiadis, Mårten Wadenbäck, and Michael Felsberg. DKM: Dense Kernelized Feature Matching for Geometry Estimation. In *CVPR*, 2023. 1, 2

[18] Johan Edstedt, Georg Bökman, and Zhenjun Zhao. DeDoDe v2: Analyzing and Improving the DeDoDe Keypoint Detector. In *CVPR*, 2024. 2

[19] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. RoMa: Revisiting Robust Lossses for Dense Feature Matching. In *CVPR*, 2024. 1, 2

[20] Johan Edstedt, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. DaD: Distilled Reinforcement Learning for Diverse Keypoint Detection. *CoRR*, 2025. 2, 4, 5, 6, 7

[21] Hugo Germain, Guillaume Bourmaud, and Vincent Lepetit. S2DNet: Learning accurate correspondences for sparse-to-dense feature matching. In *ECCV*, 2020. 2

[22] Pierre Gleize, Weiyao Wang, and Matt Feiszli. Silk: Simple learned keypoints. In *ICCV*, 2023. 2

[23] Christopher G Harris, Mike Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, 1988. 1, 2

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016. 5

[25] IMC 2021. CVPR 2021 Image Matching Challenge. `https://www.cs.ubc.ca/research/image-matching-challenge/`, 2021. 1

[26] 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. *IJCV*, 2020. 2

[27] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *NeurIPS*, 2017. 2

[28] Johannes Künzel, Anna Hilsmann, and Peter Eisert. RIPE: Reinforcement Learning on Unlabeled Image Pairs for Robust Keypoint Extraction. *ICCV*, 2025. 2

[29] Jongmin Lee, Byungjin Kim, and Minsu Cho. Self-supervised equivariant learning for oriented keypoint detection. In *CVPR*, 2022. 2, 3

[30] Kunhong Li, Longguang Wang, Li Liu, Qing Ran, Kai Xu, and Yulan Guo. Decoupling makes weakly supervised local feature better. In *CVPR*, 2022. 2

[31] Zhengqi Li and Noah Snavely. MegaDepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018. 1, 2, 6

[32] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local Feature Matching at Light Speed. In *ICCV*, 2023. 2, 6, 7, 8

[33] Ce Liu, Jenny Yuen, and Antonio Torralba. SIFT Flow: Dense correspondence across scenes and its applications. *IEEE TPAMI*, 2010. 6

[34] I. Loshchilov and F. Hutter. Fixing Weight Decay Regularization in Adam. *arXiv:1711.05101*, 2017. 5

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

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

[37] Dominik Muhle, Lukas Koestler, Krishna Murthy Jatavallabhula, and Daniel Cremers. Learning correspondence uncertainty via differentiable nonlinear least squares. In *CVPR*, 2023. 2

[38] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE TPAMI*, 26(6):756–770, 2004. 1

[39] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 5

[40] Guilherme Potje, Felipe Cadar, André Araujo, Renato Martins, and Erickson R Nascimento. Xfeat: Accelerated features for lightweight image matching. In *CVPR*, 2024. 2

[41] Filip Radenović, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Revisiting Oxford and Paris: Large-scale image retrieval benchmarking. In *CVPR*, 2018. 5

[42] Jerome Revaud, Philippe Weinzaepfel, César De Souza, Noe Pion, Gabriela Csurka, Yohann Cabon, and Martin Humenberger. R2D2: Repeatable and reliable detector and descriptor. In *NeurIPS*, 2019. 1, 2, 3

[43] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary R Bradski. ORB: An efficient alternative to SIFT or SURF. In *ICCV*, 2011. 1, 2, 6

[44] Emanuele Santellani, Christian Sormann, Mattia Rossi, Andreas Kuhn, and Friedrich Fraundorfer. S-TREK: Sequential Translation and Rotation Equivariant Keypoints for local feature extraction. In *ICCV*, 2023. 1, 2, 3, 4, 7

[45] Emanuele Santellani, Martin Zach, Christian Sormann, Mattia Rossi, Andreas Kuhn, and Friedrich Fraundorfer. GMM-IKRS: Gaussian mixture models for interpretable keypoint refinement and scoring. In *ECCV*, 2024. 2

[46] 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, 7

[47] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 2, 3

[48] Paul-Edouard Sarlin, Philipp Lindenberger, Viktor Larsson, and Marc Pollefeys. Pixel-Perfect Structure-From-Motion With Featuremetric Refinement. *IEEE TPAMI*, 2023. 1

[49] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomas Pajdla. Benchmarking 6DOF outdoor visual localization in changing conditions. In *CVPR*, 2018. 1

[50] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1, 2, 7

[51] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *CVPR*, 2017. 6, 7, 8

[52] Jianbo Shi et al. Good features to track. In *CVPR*, 1994. 2

[53] C. Spearman. The proof and measurement of association between two things. *American Journal of Psychology*, 15(1): 72–101, 1904. 4

[54] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-Free Local Feature Matching with Transformers. *CVPR*, 2021. 2

[55] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In *NeurIPS*, 1999. 4

[56] Javier Tirado-Garín, Frederik Warburg, and Javier Civera. DAC: Detector-Agnostic Spatial Covariances for Deep Local Features. In *3DV*, 2024. 2, 7

[57] Prune Truong, Martin Danelljan, Radu Timofte, and Luc Van Gool. PDC-Net+: Enhanced Probabilistic Dense Correspondence Network. *IEEE TPAMI*, 2023. 1

[58] Michał J Tyszkiewicz, Pascal Fua, and Eduard Trulls. DISK: Learning local features with policy gradient. In *NeurIPS*, 2020. 1, 2, 3, 4, 6, 7

[59] Changhao Wang, Guanwen Zhang, Zhengyun Cheng, and Wei Zhou. Rethinking low-level features for interest point detection and description. In *ACCV*, 2022. 2

[60] Qianqian Wang, Xiaowei Zhou, Bharath Hariharan, and Noah Snavely. Learning Feature Descriptors using Camera Pose Supervision. In *ECCV*, 2020. 2

[61] Xinjiang Wang, Zeyu Liu, Yu Hu, Wei Xi, Wenxian Yu, and Danping Zou. Featurebooster: Boosting feature descriptors with a lightweight neural network. In *CVPR*, 2023. 2

[62] Maurice Weiler and Gabriele Cesa. General e (2)-equivariant steerable cnns. In *NeurIPS*, 2019. 2, 3

[63] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. LIFT: Learned invariant feature transform. In *ECCV*, 2016. 2

[64] Bernhard Zeisl, Pierre Fite Georgel, Florian Schweiger, Eckehard G Steinbach, Nassir Navab, and G Munich. Estimation of Location Uncertainty for Scale Invariant Features Points. In *BMVC*, 2009. 2

[65] Jingbo Zeng, Zaiwang Gu, Weide Liu, Lile Cai, and Jun Cheng. Uncertainty aware interest point detection and description. In *WACV*, 2025. 2

[66] Xiaoming Zhao, Xingming Wu, Jinyu Miao, Weihai Chen, Peter CY Chen, and Zhengguo Li. Alike: Accurate and lightweight keypoint detection and descriptor extraction. *IEEE Transactions on Multimedia*, 25:3101–3112, 2022. 2

[67] Xiaoming Zhao, Xingming Wu, Weihai Chen, Peter CY Chen, Qingsong Xu, and Zhengguo Li. Aliked: A lighter keypoint and descriptor extraction network via deformable transformation. *IEEE Transactions on Instrumentation and Measurement*, 72:1–16, 2023. 1, 2, 3, 4, 5, 6, 7

[68] Hao Zhou, Torsten Sattler, and David W Jacobs. Evaluating local features for day-night matching. In *ECCV*, 2016. 5, 6