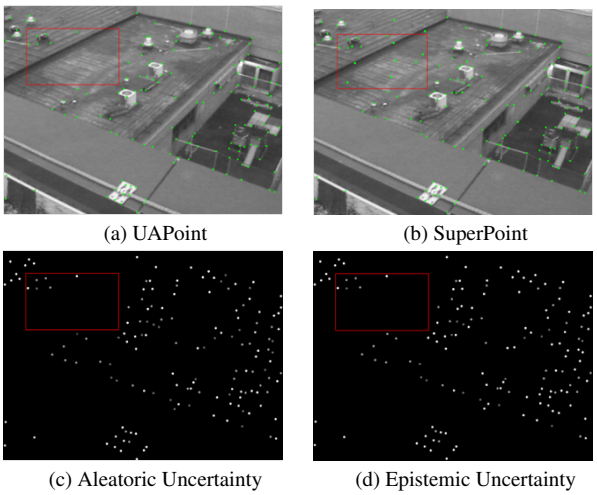
Uncertainty Aware Interest Point Detection and Description

Jingbo Zeng[1,2], Zaiwang Gu[2], Weide Liu[3], Lile Cai[2], Jun Cheng[2*]
[1]School of Electrical and Electronic Engineering, Nanyang Technological University
[2]Institute for Infocomm Research (I^{2}R), A*STAR, Singapore
[3]Boston Children's Hospital and Harvard Medical School, Boston, MA
{ZENG0143, weide001}@e.ntu.edu.sg {Gu_Zaiwang, caill, cheng_jun}@i2r.a-star.edu.sg

Abstract

Interest point detection and description play an important role in many visual tasks, including image registration, pose estimation, 3D reconstruction, and more. State-of-the-art interest point detection techniques are based on deep neural networks (NNs), which are prone to produce overconfident predictions. However, calibrated and robust uncertainty measurement is crucial when deploying deep NN models in safety critical applications. In this work, we propose a novel Uncertainty-Aware interest Point (UAPoint) detection method to address this problem. Our method leverages evidential learning to learn both aleatoric and epistemic uncertainty. We further propose a constrained sampling scheme to construct more efficient training pairs for the descriptor decoder. We evaluate our method on a wide range of benchmarks and show that our method achieves state-of-the-art performance. Code will be released in https://github.com/JingboZeng/UAPoint.



(a) UAPoint (b) SuperPoint (c) Aleatoric Uncertainty (d) Epistemic Uncertainty

Figure 1. Comparison between the proposed UAPoint and SuperPoint [11]. UAPoint computes the uncertainty of each interest point detected. (a) interest point by UAPoint. (b) interest point by SuperPoint. (c) aleatoric uncertainty of UAPoint (d) epistemic uncertainty of UAPoint. Higher intensity in (c) and (d) indicates interest points with less uncertainties.

1. Introduction

Interest point detection aims to identify and locate distinctive points in images. These interest points, a.k.a keypoints, should be invariant to noise, geometric transformation, and viewpoint and illumination changes. They serve as the foundation for various tasks such as image registration [32,49], 3D reconstruction [23], structure-from-motion [51], and simultaneous localization and mapping [10,42].

Classical methods for interest point detection perform keypoint localization and descriptor extraction in a stage-wise manner [1,5,9,20,24,29,38,47]. State-of-the-art methods are based on deep neural networks (NNs) [4, 11, 14, 18, 25,28,33–36,45,46,57,58,60,61,63,64]. These NN models have separate branches for keypoint detection and descriptor prediction, and perform both tasks simultaneously. The output of the keypoint detection branch is a heatmap that predicts the probability of each location being a keypoint, while the descriptor branch outputs dense local descriptors (one for each pixel).

Deep learning-based keypoint detection methods suffer from a common limitation of deterministic neural nets: the trained model remains ignorant to its prediction confidence. As revealed in [19], modern neural networks tend to produce overconfident results. This can pose problems when being deployed in safety critical domains, where the model may predict confident keypoints at unreliable locations, making accurate matching difficult.

Motivated by the need for calibrated and robust uncertainty measurement, we propose a novel Uncertainty-Aware interest Point (UAPoint) detection method which computes uncertainties for each interest point detected as shown in

*corresponding author

Fig. 1. Our method leverages evidential learning [2] to learn both aleatoric and epistemic uncertainty. In particular, the former measures the uncertainty caused by the inherent noise in the data, while the latter measures the uncertainty caused by the lack of knowledge of the model. We demonstrate on a wide range of benchmarks that the incorporation of evidential learning into keypoint detection leads to improved model performance.

Our main contributions can be summarized as follows:

1. We introduce an uncertainty aware keypoint detection and description method, which incorporates deep evidential learning into keypoint detection.

2. We propose a constrained sampling scheme to improve the self-supervised descriptor training. By selecting hard negative samples, the model learns better descriptors to distinguish detected keypoints.

3. We evaluate our method extensively and demonstrate that it outperforms state-of-the-art methods in various tasks, including homography estimation, outdoor localization, and relative pose estimation.

2. Related Works

2.1. Interest point detection and description

Early works such as SIFT [38], SURF [5], and others [9, 20, 24, 47, 48] have demonstrated their effectiveness in practical applications. These techniques use prior knowledge from designers to explicit geometric concept like corners and gradients. Some of them are robust to deal with viewpoint change and illumination variance. Handcrafted detectors have laid the groundwork for subsequent research. With the advent of learning-based algorithms, these methods are less popular, but still exhibit comparable advantages in certain scenarios [16, 42].

Recent work like SuperPoint [11] adopts self-supervised learning to detect points with obvious geometrical features as keypoints from synthetic dataset with adaptation on real-world images. GLAMpoint [56] exploits reinforcement learning methods to find correctly matched keypoints for homography estimation between image pairs. Some other methods [6, 12, 43, 46, 57, 62, 63] also provide end-to-end pipelines. NeSS-ST [44] combines the classical handcrafted Shi-Tomasi detector [24] with a neural network to select stable keypoints. SuperGlue [49] and LightGlue [32] focus on introducing improved and more productive methods for feature matching. SiLK [18] aims at simplicity on structure and training. Recently, some researchers work on dense matching without keypoints [13, 15]. Although our approach can be applied to these methods, this paper focuses on keypoint based methods.

2.2. Uncertainty Estimation

With the increasing application of deep learning techniques, prediction differentiation has become a growing demand. One possible solution is to measure uncertainty of each forecast. Bayesian neural network [26] provides a creative inspiration, which introduces uncertainty by substituting the deterministic weight parameters with distributions. However, the expensive computational cost to optimize a large number of parameters is unaffordable. Monte Carlo dropout [17] is widely used to reduce this negative influence. It formulates the dropout process as Bernoulli distributed random variables to view the training process as variational inference. Deep evidential regression [2] extend uncertainty estimation to regression tasks, assuming all predictions are normally distributed. It is followed by normal inverse gamma distribution estimation to ensure an explicit representation of the aleatoric and epistemic uncertainties.

Previously, uncertainty estimation has been explored for many regression tasks such as classification, object detection [22, 59], stereo matching [7, 37, 53], model generalization [39]. [41, 55] provide plug-and-play use spatial covariance to model uncertainty. They focus on selecting available detections with less spatial covariance in existing results. In this work, we introduce evidential learning into keypoint detection by computing uncertainties on keypoint heatmap as an alternative target, which means uncertainty is an extra criterion to excavate potential relationship.

3. Methodology

The system diagram of our method is presented in Fig. 2. The proposed UAPoint framework consists of three components: feature encoder, uncertainty aware interest point decoder, and constrained descriptor decoder. The feature encoder adopts the pretrained ResNet-34 [21] as backbone. The uncertainty aware interest point decoder computes interest point heatmap and predicts both aleatoric uncertainty and epistemic uncertainty (Sec. 3.1). The constrained descriptor decoder computes descriptor for each interest point. We use the same structure as that used in SuperPoint [11], but compute a different loss via constrained sampling (Sec. 3.2). The loss of each module are detailed in Sec. 3.3.

3.1. Uncertainty Aware Interest Point Decoder

As depicted in Fig. 3, the uncertainty aware keypoint decoder computes over feature map with size $H/8 \times W/8 \times 256$ from feature encoder and outputs a tensor sized $H/8 \times W/8 \times 65$. The tensor is further transformed through Softmax and Reshape to obtain a $H \times W$ heatmap $\mathcal{D}$. $\mathcal{D}$ represents the probability of each pixel in the original input being an interest point. At the same time, $\mathcal{D}$ is also used by the uncertainty estimation head to compute the aleatoric

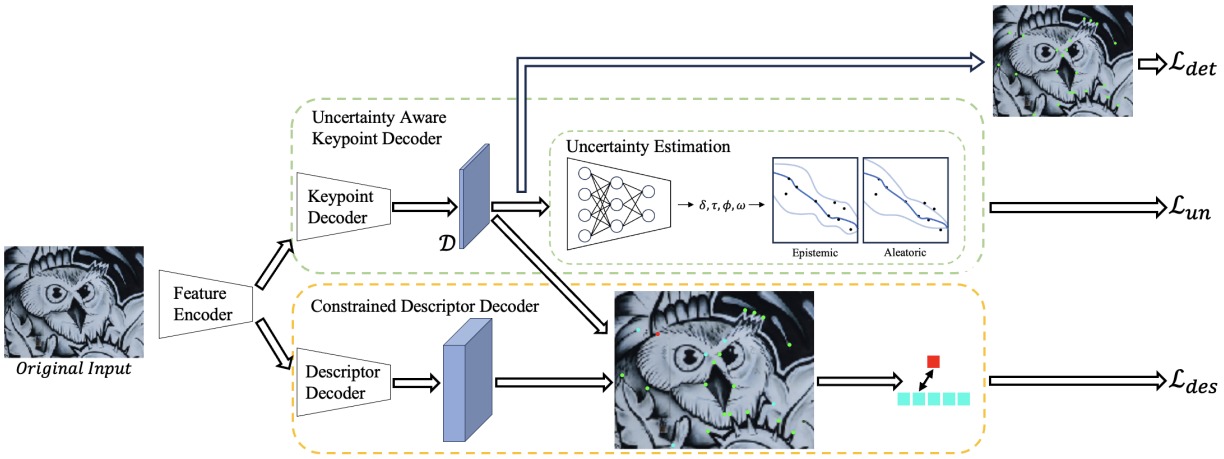


Figure 2. System diagram of the proposed UAPoint framework. It comprises three major components: feature encoder, uncertainty aware interest point decoder, and constrained descriptor decoder. We adopt a multi-task learning approach to train the network.

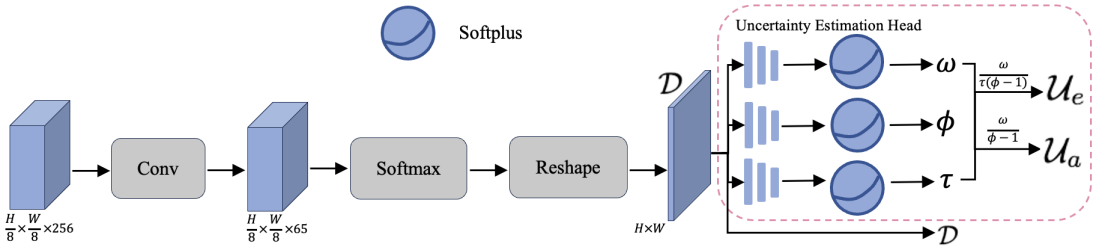


Figure 3. The structure of uncertainty aware interest point decoder. It transforms the tensor sized $H/8 \times W/8 \times 65$ to $H \times W$. It predicts key point heatmap $\mathcal{D}$ and three distribution parameters τ, ϕ, ω to measure aleatoric and epistemic uncertainties.

and epistemic uncertainties.

To achieve uncertainty aware keypoint detection, we incorporate evidential learning into the network. Following deep evidential learning [2], the heatmap $\mathcal{D}$ for keypoint detection should follow a normal distribution with unknown parameter (μ, σ^2), where μ is the mean and σ is the variance. Assuming μ and σ can be drawn through normal distribution and inverse-gamma distribution respectively, these two can be represented by:

$$\mathcal{D} \sim \mathcal{N}(\mu, \sigma^2), \mu \sim \mathcal{N}(\mathcal{D}, \sigma^2 \tau^{-1}), \sigma^2 \sim \Gamma^{-1}(\phi, \omega), \quad (1)$$

where $\mathcal{N}(\cdot)$ represents normal distribution, $\Gamma^{-1}(\cdot)$ represents inverse-gamma distribution, $\mathcal{D} \in \mathbb{R}$, $\phi > 1$, $\omega > 0$, and $\tau > 0$.

To estimate the three parameters τ, ϕ, ω, we employ three branches. As shown in Fig. 3, the uncertainty estimation head employs a 2D convolution layer with softplus activation in each branch, and outputs three tensors $V_\tau, V_\phi, V_\omega \in \mathbb{R}^{H \times W}$. The distribution parameters can be obtained as follows:

$$o = \text{Softplus}\left(\sum V_o \cdot \mathcal{D}\right), \quad (2)$$

where $o \in \{\tau, \phi, \omega\}$ and Softplus$(\cdot)$ denotes the softplus activation function.

The aleatoric uncertainty $\mathcal{U}_a$ and epistemic uncertainty $\mathcal{U}_e$ can be computed as:

$$\begin{aligned} \mathcal{U}_a &= \mathbb{E}(\sigma^2) = \frac{\omega}{\phi - 1}, \\ \mathcal{U}_e &= Var(\mu) = \frac{\omega}{\tau(\phi - 1)}. \end{aligned} \quad (3)$$

3.2. Constrained Sampling

The descriptor decoder has the same structure as that in SuperPoint [11]. It is trained in a self-supervised manner under the assumption that each interest point shall be invariant to homography transform. Previously Schroff et al. [52] proposed to compute a triplet loss by comparing each point with its paired positive point and one randomly selected negative point. SuperPoint [11] follows this scheme to compute descriptor loss. However, as most detected interest points come from distinctive locations like edges, corners, and junctions, they are often easily differentiable from points that are randomly sampled from other parts of the image, making model training inefficient. Intuitively, selecting

hard negative points, *i.e.*, keypoints that are most similar to the anchor, would help the network learn more distinctive descriptors.

We propose a constrained sampling approach. In addition to one randomly selected negative point N_r, we further include a second point as constrained negative point via constrained sampling, where we choose another point N_c from other keypoints that are most similar to the anchor (see Fig. 2 for illustration). We rank keypoints by their Euclidean distance to the anchor keypoint and choose one randomly from the top k most similar ones. We use $k = 5$ in our paper based on the ablation study in Fig. 5d

3.3. Loss Functions

3.3.1 Detector Loss

The detector loss of an image is computed as the mean cross-entropy loss between the predicted heatmap and the ground truth:

$$\mathcal{L}_p(\mathcal{D}) = \frac{1}{HW} \sum_{i=1,j=1}^{H,W} CE(\mathcal{D}(i,j), \mathcal{D}^{gt}(i,j)), \quad (4)$$

where $CE(\cdot)$ denotes cross entropy loss, the subscripts (i, j) denote pixel coordinates.

The detector loss $\mathcal{L}_{det}$ is computed from pairs of synthetically warped images:

$$\mathcal{L}_{det} = \mathcal{L}_p(\mathcal{D}_X) + \mathcal{L}_p(\mathcal{D}_{\mathcal{H}(X)}), \quad (5)$$

where $\mathcal{D}_X$ denotes the heatmap for input image X and $\mathcal{H}(\cdot)$ denotes the homography warping.

3.3.2 Uncertainty Loss

With the definition of different distribution parameters, we define uncertainty loss $\mathcal{L}_{un}$ inspired by [2, 37], which is a combination of two terms $\mathcal{L}_{neg}$ and $\mathcal{L}_{reg}$. $\mathcal{L}_{neg}$ is computed during training process, represented by:

$$\begin{aligned} \mathcal{L}_{neg} = (\phi + \frac{1}{2}) \log(\mathcal{D}^{gt} - \mathcal{D})\tau - \phi \log(\Psi) \\ + \log(\frac{\Gamma(\phi)}{\Gamma(\phi + \frac{1}{2})}) + \frac{1}{2} \log(\frac{\pi}{\tau}), \end{aligned} \quad (6)$$

where $\Psi = 2\omega(\tau + 1)$.

To reduce the evidence of incorrect detection, a regularization loss $\mathcal{L}_{reg}$ is introduced:

$$\mathcal{L}_{reg} = \left|\mathcal{D}^{gt} - \mathcal{D}\right| \cdot (2\tau + \phi). \quad (7)$$

The overall uncertainty loss $\mathcal{L}_{un}$ is computed as:

$$\mathcal{L}_{un} = \mathcal{L}_{neg} + \alpha \mathcal{L}_{reg}, \quad (8)$$

where $\alpha > 0$ controls the weight of the regularization loss. We use $\alpha = 0.5$ based on the ablation study in Fig. 5a.

3.3.3 Descriptor Loss

The descriptor loss $\mathcal{L}_{des}$ is computed as:

$$\mathcal{L}_{des} = \frac{1}{M} \sum_{i=1}^{M} \left[d(v_i, p_i) - \frac{1}{2} d(v_i, n_{r,i}) - \frac{1}{2} d(v_i, n_{c,i}) \right], \quad (9)$$

where M denotes the number of keypoints, $d(\cdot)$ denotes Euclidean distance function, v_i denotes the descriptor of i^{th} selected interest point P_i, p_i denotes the descriptor of its corresponding positive point in the warped image, and $n_{r,i}$ and $n_{c,i}$ denote the descriptors for the selected negative points $N_{r,i}$ and $N_{c,i}$ for the i^{th} keypoint respectively.

3.3.4 Overall Loss

The overall loss is a combination of the above three losses:

$$\mathcal{L} = \mathcal{L}_{det} + \lambda_1 \mathcal{L}_{un} + \lambda_2 \mathcal{L}_{des}, \quad (10)$$

where λ_1 and λ_2 are parameters to control the balance of the three items. We use $\lambda_1 = 1$ and $\lambda_2 = 0.1$ based on ablation studies in Fig. 5b and Fig. 5c, respectively.

4. Experiments

Implementation Details We follow the original pipeline in [11] to train UAPoint model. We use pretrained ResNet-34 to initialize the backbone in UAPoint. Similarly to that in SuperPoint, we retrain MagicPoint in synthetic data and then adapt it to MS-COCO [31] through homography adaptation. MS-COCO is a large-scale object detection dataset with more than 330K images, providing annotations for different applications such as object detection, interest point detection, and image segmentation. We use COCO2014 to train our model. The generated pseudo-ground truth is used to train UAPoint in a self-supervised manner. We set the training batch size as 64, initial learning rate as 0.0001, each training process needs 200,000 iterations. The learning rate is decayed by a factor of 2 after $50,000$, $100,000$, and $150,000$. Regarding ablation studies, all experiments are in variants of UAPoint. We maintain all other settings consistent with the original work for each model.

Evaluation of interest point detectors and descriptors has been well studied and we follow the protocol proposed by Mikołajczyk *et al*. [40] to evaluate UAPoint. We compute two detector metrics, mean localization error (MLE) and repeatability (Rep.), two descriptor metrics mean average precision (mAP) and matching score (MS) to measure the performance of interest point detector and descriptor, similar to that in [11, 18]. Our model is trained on MS-COCO [31] images and evaluated on HPatches dataset.

4.1. Homography Estimation and Repeatability

We first evaluate the homography estimation of UAPoint on HPatches [3] dataset. HPatches dataset is a public set for

Table 1. HPatches Detector and Descriptor Evaluation. UAPoint achieves competitive performance to other methods. **Bold** represents the best result and Underline shows the second best.

Method	MLE↓		Rep.↑		mAP↑		MS↑		Homo. Esti.↑	
	$\epsilon = 1$	$\epsilon = 3$	$\epsilon = 1$	$\epsilon = 3$	$\epsilon = 1$	$\epsilon = 3$	$\epsilon = 1$	$\epsilon = 3$	$\epsilon = 1$	$\epsilon = 3$
SIFT [38]	**0.751**	**0.833**	0.323	0.495	0.372	0.694	0.269	0.313	**0.424**	0.676
ORB [48]	1.016	1.157	0.482	0.641	0.515	0.735	0.197	0.266	0.150	0.395
LIFT [62]	0.928	1.102	0.241	0.449	0.416	0.664	0.275	0.315	0.284	0.598
SuperPoint [11]	0.974	1.158	0.352	0.581	0.537	0.821	0.331	0.470	0.310	0.684
R2D2 [46]	0.915	1.043	0.363	0.718	0.342	0.754	0.375	0.488	0.204	0.497
DISK [57]	0.897	0.961	0.384	0.686	0.223	0.519	0.380	0.474	0.221	0.520
SiLK [18]	0.891	0.879	0.622	**0.807**	0.615	**0.873**	0.404	0.513	0.398	0.664
Ness-ST	0.887	0.845	0.628	0.784	0.608	0.842	0.411	0.509	0.385	**0.718**
UAPoint	0.885	0.925	**0.636**	0.796	**0.630**	0.861	**0.423**	**0.527**	0.407	0.706

Table 2. Comparison with LoFTR [54]. LoFTR is trained on MegaDepth and ScanNet, respectively.

Method	mAP↑		Homo. Esti↑	
	$\epsilon = 1$	$\epsilon = 3$	$\epsilon = 1$	$\epsilon = 3$
LoFTR (MegaDepth)	**0.645**	**0.871**	0.373	0.648
LoFTR (ScanNet)	0.236	0.568	0.066	0.331
UAPoint	0.630	0.861	**0.407**	**0.706**

Table 3. Evaluation of repeatability on HPatches dataset.

Methods	Illumination mAA↑($\epsilon = 5$)	Viewpoint mAA↑($\epsilon = 5$)
SuperPoint [11]	0.883	0.541
R2D2 [46]	**0.888**	0.506
Ness-ST [44]	0.883	0.556
UAPoint	0.881	**0.564**

evaluating the local descriptor, which contains 116 scenes and 696 different images. These 116 scenes are divided into two parts; the first 57 scenes perform large illumination changes, and the other 59 record large viewpoint changes.

We compare UAPoint with seven different methods: SIFT [38], ORB [48], LIFT [62], SuperPoint [11], R2D2 [46], DISK [57], and SiLK [18]. All pairs of original inputs are resized to 1080×1080, with 3000 interest points. NMS is still implemented to ensure an even distribution. We use different thresholds $\epsilon = 1, 3$ respectively, to introduce different homographies.

Quantitative homography estimation results are given in Tab. 1. UAPoint outperforms the existing detector-based methods on repeatability and mean average precision when $\epsilon = 1$, while comparable for $\epsilon = 3$. In particular, our approach has a strong margin when the error threshold is small. These prove UAPoint has a great accuracy when facing pixel-level localization. SIFT provides great performance on MLE and homography estimation ($\epsilon = 1$), because it utilizes additional sub-pixel localization, while other methods do not perform this step. Fig. 4 shows a visualized comparison. Although each method has similar performance facing pixels with distinct geometric features, our proposed UAPoint provide more precise detection and less error when tackling repeatable or low-texture region. Even compared to the detector-free method [54], as the results shown in Tab. 2, our method still has competitive performance.

The repeatability is computed as the percentage of keypoints that are both detected in image pairs. We evaluate the repeatability of our model on HPatches compared to NeSS-ST [44], R2D2 [46], SuperPoint [11]. We use mean Average Accuracy for both illumination and viewpoint change. The original inputs are resized to 1024×1024, and each image has no more than 1000 points. The results are shown in Tab. 3. Our method has an improvement when facing viewpoint changes and shows similar performance under illumination changes compared with other methods.

4.2. Ablation Studies

4.2.1 Effectiveness of Components in UAPoint

To evaluate the effectiveness of different modules in our framework, we train different variants using MS-COCO [31], each training requires 200,000 iterations and 10 GPU days. Our baseline is a retrained SuperPoint [11], but we utilizes ResNet-34 [21] as the backbone. Quantitative results are given in Tab. 4. We validate these variants with the same metrics. Ablation studies show that these two modules are effective and indispensable. The overall framework has excellent performance on all metrics. The constrained sampling provides better improvement on matching score and homography estimation with $\epsilon = 1$ and $\epsilon = 3$, while uncertainty estimation contributes more on repeatability, mean average precision, and homography estimation when $\epsilon = 5$.

(a) UAPoint (b) SuperPoint (c) LIFT (d) SIFT

Figure 4. Visualization of keypoints detected by LIFT [62], SIFT [38], SuperPoint [11], and UAPoint. UAPoint is able to suppress keypoint detected from region with low textures or high repeatable patterns (highlighted by red boxes).

Table 4. Ablation study on the proposed components. UN represents the uncertainty loss and CS represents the constrained sampling scheme.

Method		Detector&Descriptor			Homo. Estimation		
UN	CS	Rep.↑	mAP↑	MS↑	$\epsilon = 1 \uparrow$	$\epsilon = 3 \uparrow$	$\epsilon = 5 \uparrow$
		0.654	0.806	0.502	0.337	0.686	0.749
	✓	0.717	0.822	0.515	0.364	0.695	0.763
✓		0.732	0.825	0.513	0.382	0.697	0.785
✓	✓	**0.796**	**0.861**	**0.527**	**0.407**	**0.706**	**0.851**

Table 5. Comparison of models sizes and running-time.

	UAPoint	SuperPoint	NeSS-ST	R2D2
Size (MB)	7.52	4.96	3.54	1.85
Inference time (ms)	5.1	4.0	8.2	24.4

4.2.2 Ablation studies for different parameters

We also evaluate the performance using different parameters α in Eq. (8), λ_1 and λ_2 in Eq. (10), and top-k selection in Sec. 3.2. The results are provided in Fig. 5. Our framework provides the best performance when $\alpha = 0.5$, $\lambda_1 = 1$, $\lambda_2 = 0.1$, and $k = 5$. We follow these results to compare the performance in downstream tasks.

We also provide comparison on computational cost and inference time between UAPoint and recent learning-based competitors like SuperPoint, R2D2, and NeSS-ST in Tab. 5.

4.3. Performance in Other Tasks

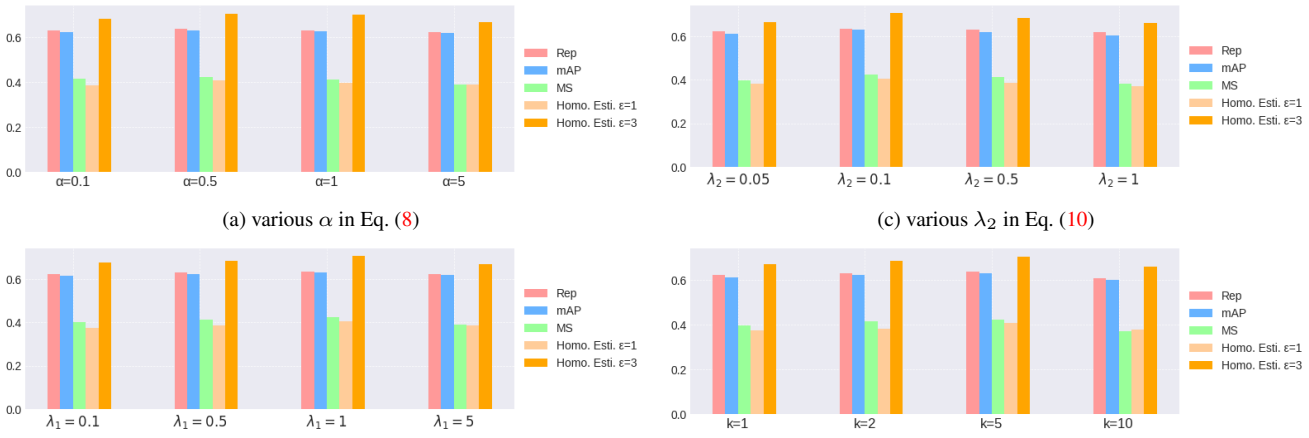
We validate our UAPoint by combining it with dense matchers for downstream tasks, including outdoor localization, outdoor pose estimation, and indoor pose estimation. Some competitive matching algorithms and detector-free methods are used. We follow the setup in LightGlue [32] to design this comparison.

4.3.1 Outdoor Localization

We evaluate visual localization using Aachen Day-Night benchmark [50], which is a dataset designed for benchmarking outdoor visual localization in changing conditions. It has 4328 database images and 922 query images with changing conditions of weather, season, and day-night cycles. It focuses on localizing high-quality night-time images. We measure the percentage of query images localized in three accuracy intervals: $(0.25m, 2°)$, $(0.5m, 5°)$, and $(5m, 10°)$. The second column in Tab. 6 shows in day-time scenarios, UAPoint performs better on $(0.25m, 2°)$ but on par with SuperPoint on $(0.5m, 5°)$. In night-time images, UAPoint performs better on both intervals. For the loose criteria $(5m, 10°)$, the accuracy is saturated.

4.3.2 Outdoor Pose Estimation

We use image pairs from MegaDepth [30] for outdoor pose estimation. MegaDepth serves as a comprehensive resource created to tackle the difficulties of perceiving depth in images for depth estimation. We select RANSAC [27] as the pose estimator. To measure the performance, we predict pose accuracy and record AUC at 5°, 10°, and 20° in rotations and translations to compare the angular error between different feature extractors and matchers. The forth column in Tab. 6 provides the results.



(a) various α in Eq. (8)

(c) various λ_2 in Eq. (10)

(b) various λ_1 in Eq. (10)

(d) various k in top-k selection in Sec. 3.2

Figure 5. Sensitivity of model performance for various parameters.

Table 6. Comparison for downstream tasks: outdoor localization performance on Aachen Day-Night dataset, outdoor pose estimation performance on MegaDepth1500 dataset, and indoor pose estimation on ScanNet dataset. The advantage of UAPoint is most significant for smaller error tolerance.

Methods	Aachen Day-Night		Pairs per Second	MegaDepth	ScanNet
	Day↑	Night↑		RANSAC AUC↑	Pose Esti. AUC↑
	(0.25m, 2°) / (0.5m, 5°) / (5m, 10°)			5° / 10° / 20°	5° / 10° / 20°
LoFTR	0.887 / **0.956** / 0.990	0.785 / 0.906 / 0.990	-	0.503 / 0.671 / 0.799	**0.221** / **0.408** / **0.576**
Alike [64]	0.857 / 0.924 / 0.967	0.816 / 0.888 / 0.990	-	0.494 / 0.618 / 0.714	0.08 / 0.164 / 0.259
XFeat [45]	0.847 / 0.915 / 0.965	0.776 / 0.989 / 0.980	-	0.416 / 0.564 / 0.677	0.167 / 0.326 / 0.478
ZippyPoint [25]	0.807 / 0.886 / 0.937	0.612 / 0.704 / 0.796	-	0.236 / 0.349 0.463	0.165 / 0.331 / 0.488
SuperPoint+SGMNet	0.868 / 0.942 / 0.977	0.837 / 0.918 / 0.990	10.2	0.432 / 0.616 / 0.756	0.154 / 0.321 / 0.483
SuperPoint+SuperGlue	0.882 / 0.955 / 1.000	0.867 / 0.929 / 1.000	6.5	0.497 / 0.671 / 0.787	0.162 / 0.338 / 0.518
SuperPoint+LightGlue	0.892 / 0.954 / 1.000	0.877 / 0.939 / 1.000	17.2	0.499 / 0.670 / 0.805	0.164 / 0.336 / 0.502
UAPoint+SGMNet	-	-	-	-	-
UAPoint+SuperGlue	0.887 / 0.954 / 1.000	0.871 / 0.931 / 1.000	5.3	0.509 / 0.673 / 0.804	0.165 / 0.327 / 0.524
UAPoint+LightGlue	**0.894** / 0.954 / 1.000	**0.881** / **0.940** / 1.000	15.8	**0.514** / **0.675** / **0.811**	0.178 / 0.351 / 0.536

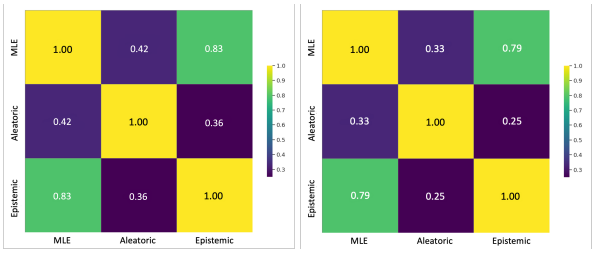
4.3.3 Indoor Pose Estimation

As indoor images are lacking of texture, indoor pose estimation is a challenging issue. We use ScanNet [8] as the test set, and report AUC at 5°, 10°, and 20°. The last colunm in Tab. 6 presents the results.

4.4. Uncertainty Analysis

Aleatoric uncertainty refers to the inherent randomness of the data, while epistemic uncertainty represents a lack of information during training. The former is introduced by the inevitable noise when collecting data, and it cannot be reduced by optimization. The latter relates to model capacity, measuring the uncertainty of model parameters estimated by the training process. We conduct Pearson correlation analysis between MLE and two uncertainties on MS-COCO [31] training set and HPatches [3].

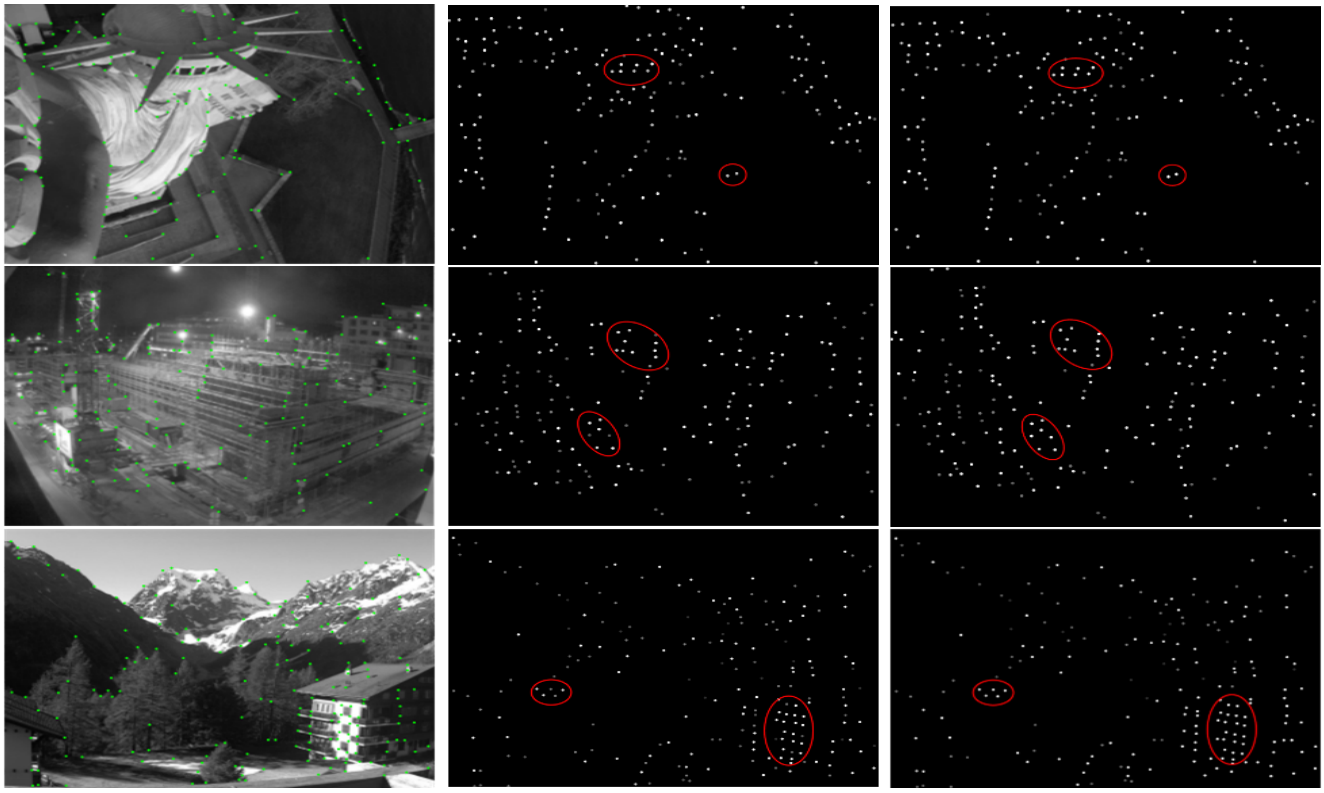
For both Fig. 6a and Fig. 6b, the heatmap show that MLE and its uncertainties have a positive correlation, which provides a correctness of our motivation to introduce uncertainty to rule interest point detection. The value between MLE and epistemic uncertainty is higher than the value between MLE and aleatoric uncertainty. On the MS-COCO training set, the former is 0.83 and the latter is 0.42. On HPatches, these values are 0.79 and 0.33. It is worth not-



(a) MLE on MS-COCO

(b) MLE on HPatches

Figure 6. Uncertainty analysis. (a) The Pearson correlation coefficient between MLE and the learned uncertainties on MS-COCO; (b) The Pearson correlation coefficient between MLE on HPatches and the learned uncertainties.



(a) Keypoint detection (b) Aleatoric uncertainty (c) Epistemic uncertainty

Figure 7. Visualization of keypoints detected by UAPoint and the learned uncertainties. Higher intensity indicates higher confidence (less uncertainty). UAPoint detects the most confident keypoints at regions with distinctive patterns (indicated by red circles).

Table 7. Effect of discarding uncertain keypoints.

Points	Det. & Des.			Homo. Esti.	
	Rep.↑	mAP↑	MS↑	$\epsilon = 1$↑	$\epsilon = 3$↑
0%	0.796	0.861	0.527	0.407	0.706
5%	0.803	0.868	0.534	0.415	0.707
10%	**0.805**	**0.872**	**0.536**	**0.418**	**0.713**
15%	0.794	0.863	0.522	0.411	0.708

ing that the performance of our framework is more related to epistemic uncertainty. The results also show that lower uncertainty will lead to lower MLE and more accurate detection, and vice versa. Some qualitative results are given in Fig. 7. We observe that pixels such as vertices and corners have lower uncertainties, while high uncertainties are assigned to pixels with less significant geometrical features.

We also evaluate the homography estimation performance when keypoints of top 5%, 10%, and 15% high uncertainties are discarded. The results are given in Tab. 7. The performance improves when 5% and 10% points with high uncertainty are excluded. However, it starts to drop when more points are excluded.

5. Conclusion

In this work, we propose UAPoint for uncertainty aware interest point detection and description. We leverage deep evidential learning to estimate aleatoric and epistemic uncertainties in keypoint detection. We also propose a constrained sampling scheme to construct more efficient training pairs for the descriptor decoder. Experimental results demonstrate that the proposed method leads to improved performance in keypoint localization and descriptor matching. When combined with pretrained feature matchers, our method obtained competitive results compared with state-of-the-art methods. We also show that the learned uncertainty can be utilized to filter uncertain keypoints, which further improves the results for homography estimation.

Acknowledgement

This work was supported in part by the Agency for Science, Technology and Research under its MTC programmatic funds grant No. M23L7b0021 and AI^3 horizontal technology coordinating office grant No. C231118001.

References

[1] Pablo Fernández Alcantarilla, Adrien Bartoli, and Andrew J Davison. Kaze features. In *ECCV*, pages 214–227. Springer, 2012. 1

[2] Alexander Amini, Wilko Schwarting, Ava Soleimany, and Daniela Rus. Deep evidential regression. *NeurIPS*, 33:14927–14937, 2020. 2, 3, 4

[3] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *CVPR*, pages 5173–5182, 2017. 4, 7

[4] Axel Barroso-Laguna, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Key. net: Keypoint detection by handcrafted and learned cnn filters. In *ICCV*, pages 5836–5844, 2019. 1

[5] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *ECCV*, pages 404–417. Springer, 2006. 1, 2

[6] Aritra Bhowmik, Stefan Gumhold, Carsten Rother, and Eric Brachmann. Reinforced feature points: Optimizing feature detection and description for a high-level task. In *CVPR*, pages 4948–4957, 2020. 2

[7] Shuo Cheng, Zexiang Xu, Shilin Zhu, Zhuwen Li, Li Erran Li, Ravi Ramamoorthi, and Hao Su. Deep stereo using adaptive thin volume representation with uncertainty awareness. In *CVPR*, pages 2524–2534, 2020. 2

[8] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, pages 5828–5839, 2017. 7

[9] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 1, pages 886–893, 2005. 1, 2

[10] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE TPAMI*, 29(6):1052–1067, 2007. 1

[11] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPR workshops*, pages 224–236, 2018. 1, 2, 3, 4, 5, 6

[12] 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. 2

[13] Johan Edstedt, Ioannis Athanasiadis, Mårten Wadenbäck, and Michael Felsberg. DKM: Dense kernelized feature matching for geometry estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 2

[14] Johan Edstedt, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. Dedode: Detect, don't describe—describe, don't detect for local feature matching. In *2024 International Conference on 3D Vision (3DV)*, pages 148–157. IEEE, 2024. 1

[15] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. RoMa: Robust Dense Feature Matching. *IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 2

[16] Qiang Fu, Hongshan Yu, Xiaolong Wang, Zhengeng Yang, Yong He, Hong Zhang, and Ajmal Mian. Fast orb-slam without keypoint descriptors. *IEEE TIP*, 31:1433–1446, 2021. 2

[17] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *The International Conference on Machine Learning*, pages 1050–1059, 2016. 2

[18] Pierre Gleize, Weiyao Wang, and Matt Feiszli. Silk: Simple learned keypoints. In *ICCV*, pages 22499–22508, 2023. 1, 2, 4, 5

[19] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017. 1

[20] Chris Harris, Mike Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244, 1988. 1, 2

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 2, 5

[22] Yihui He, Chenchen Zhu, Jianren Wang, Marios Savvides, and Xiangyu Zhang. Bounding box regression with uncertainty for accurate object detection. In *CVPR*, pages 2888–2897, 2019. 2

[23] Jared Heinly, Johannes L. Schönberger, Enrique Dunn, and Jan-Michael Frahm. Reconstructing the world in six days. In *CVPR*, pages 3287–3295, 2015. 1

[24] Shi Jianbo. Good features to track. In *CVPR*, pages 593–600. IEEE, 1994. 1, 2

[25] Menelaos Kanakis, Simon Maurer, Matteo Spallanzani, Ajad Chhatkuli, and Luc Van Gool. Zippypoint: Fast interest point detection, description, and matching through mixed precision discretization. In *CVPR*, pages 6114–6123, 2023. 1, 7

[26] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *NeurIPS*, 30, 2017. 2

[27] Viktor Larsson, Zuzana Kukelova, and Yinqiang Zheng. Making minimal solvers for absolute pose estimation compact and robust. In *ICCV*, pages 2316–2324, 2017. 6

[28] Jongmin Lee, Byungjin Kim, and Minsu Cho. Self-supervised equivariant learning for oriented keypoint detection. In *CVPR*, pages 4847–4857, 2022. 1

[29] Stefan Leutenegger, Margarita Chli, and Roland Y Siegwart. Brisk: Binary robust invariant scalable keypoints. In *ICCV*, pages 2548–2555, 2011. 1

[30] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, pages 2041–2050, 2018. 6

[31] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014. 4, 5, 7

[32] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. In *ICCV*, 2023. 1, 2, 6

[33] Weide Liu, Zhonghua Wu, Yang Zhao, Yuming Fang, Chuan-Sheng Foo, Jun Cheng, and Guosheng Lin. Harmonizing base and novel classes: A class-contrastive approach for generalized few-shot segmentation. *IJCV*, 132(4):1277–1291, 2024. 1

[34] Weide Liu, Chi Zhang, Henghui Ding, Tzu-Yi Hung, and Guosheng Lin. Few-shot segmentation with optimal transport matching and message flow. *IEEE Transactions on Multimedia*, 25:5130–5141, 2022. 1

[35] Weide Liu, Chi Zhang, Guosheng Lin, and Fayao Liu. Crnet: Cross-reference networks for few-shot segmentation. In *CVPR*, pages 4165–4173, 2020. 1

[36] Weide Liu, Chi Zhang, Guosheng Lin, and Fayao Liu. Crcnet: Few-shot segmentation with cross-reference and region–global conditional networks. *IJCV*, 130(12):3140–3157, 2022. 1

[37] Jieming Lou, Weide Liu, Zhuo Chen, Fayao Liu, and Jun Cheng. Elfnet: Evidential local-global fusion for stereo matching. In *ICCV*, pages 17784–17793, 2023. 2, 4

[38] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60:91–110, 2004. 1, 2, 5, 6

[39] Changsheng Lu and Piotr Koniusz. Few-shot keypoint detection with uncertainty learning for unseen species. In *CVPR*, pages 19416–19426, 2022. 2

[40] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE TPAMI*, 27(10):1615–1630, 2005. 4

[41] Dominik Muhle, Lukas Koestler, Krishna Murthy Jatavallabhula, and Daniel Cremers. Learning correspondence uncertainty via differentiable nonlinear least squares. In *CVPR*, pages 13102–13112, 2023. 2

[42] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. 1, 2

[43] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. Lf-net: Learning local features from images. *NeurIPS*, 31, 2018. 2

[44] Konstantin Pakulev, Alexander Vakhitov, and Gonzalo Ferrer. Ness-st: Detecting good and stable keypoints with a neural stability score and the shi-tomasi detector. In *ICCV*, pages 9578–9588, 2023. 2, 5

[45] Guilherme Potje, Felipe Cadar, André Araujo, Renato Martins, and Erickson R Nascimento. Xfeat: Accelerated features for lightweight image matching. In *CVPR*, pages 2682–2691, 2024. 1, 7

[46] Jerome Revaud, Philippe Weinzaepfel, César De Souza, Noe Pion, Gabriela Csurka, Yohann Cabon, and Martin Humenberger. R2d2: Repeatable and reliable detector and descriptor. *NeurIPS*, 2019. 1, 2, 5

[47] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *ECCV*, pages 430–443. Springer, 2006. 1, 2

[48] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, pages 2564–2571. IEEE, 2011. 2, 5

[49] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, pages 4938–4947, 2020. 1, 2

[50] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. Benchmarking 6dof outdoor visual localization in changing conditions. In *CVPR*, pages 8601–8610, 2018. 6

[51] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, pages 4104–4113, 2016. 1

[52] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, pages 815–823, 2015. 3

[53] Zhelun Shen, Yuchao Dai, and Zhibo Rao. Cfnet: Cascade and fused cost volume for robust stereo matching. In *CVPR*, pages 13906–13915, 2021. 2

[54] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *CVPR*, pages 8922–8931, 2021. 5

[55] Javier Tirado-Garín, Frederik Warburg, and Javier Civera. Dac: Detector-agnostic spatial covariances for deep local features. *arXiv preprint arXiv:2305.12250*, 2023. 2

[56] Prune Truong, Stefanos Apostolopoulos, Agata Mosinska, Samuel Stucky, Carlos Ciller, and Sandro De Zanet. Glampoints: Greedily learned accurate match points. In *ICCV*, pages 10732–10741, 2019. 2

[57] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. Disk: Learning local features with policy gradient. *NeurIPS*, 33:14254–14265, 2020. 1, 2, 5

[58] Thomas Wimmer, Peter Wonka, and Maks Ovsjanikov. Back to 3d: Few-shot 3d keypoint detection with back-projected 2d features. In *CVPR*, pages 4154–4164, 2024. 1

[59] Heng Yang and Marco Pavone. Object pose estimation with statistical guarantees: Conformal keypoint detection and geometric uncertainty propagation. In *CVPR*, pages 8947–8958, 2023. 2

[60] Jie Yang, Ailing Zeng, Feng Li, Shilong Liu, Ruimao Zhang, and Lei Zhang. Neural interactive keypoint detection. In *ICCV*, pages 15122–15132, 2023. 1

[61] Jie Yang, Ailing Zeng, Ruimao Zhang, and Lei Zhang. Unipose: Detecting any keypoints. *arXiv preprint arXiv:2310.08530*, 2023. 1

[62] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *ECCV*, pages 467–483. Springer, 2016. 2, 5, 6

[63] 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

[64] 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. 1, 7