LoMa: Local Feature Matching Revisited

David Nordström[1⋆], Johan Edstedt[2⋆], Georg Bökman[3]
Jonathan Astermark[4], Anders Heyden[4], Viktor Larsson[4]
Mårten Wadenbäck[2], Michael Felsberg[2], and Fredrik Kahl[1]

[1]Chalmers University of Technology [2]Linköping University [3]University of Amsterdam [4]Centre for Mathematical Sciences, Lund University

Abstract. Local feature matching has long been a fundamental component of 3D vision systems such as Structure-from-Motion (SfM), yet progress has lagged behind the rapid advances of modern data-driven approaches. The newer approaches, such as feed-forward reconstruction models, have benefited extensively from scaling dataset sizes, whereas local feature matching models are still only trained on a few mid-sized datasets. In this paper, we revisit local feature matching from a data-driven perspective. In our approach, which we call LoMa, we combine large and diverse data mixtures, modern training recipes, scaled model capacity, and scaled compute, resulting in remarkable gains in performance. Since current standard benchmarks mainly rely on collecting sparse views from successful 3D reconstructions, the evaluation of progress in feature matching has been limited to relatively easy image pairs. To address the resulting saturation of benchmarks, we collect 1000 highly challenging image pairs from internet data into a new dataset called HardMatch. Ground truth correspondences for HardMatch are obtained via manual annotation by the authors. In our extensive benchmarking suite, we find that LoMa makes outstanding progress across the board, outperforming the state-of-the-art method ALIKED+LightGlue by +18.6 mAA on HardMatch, +29.5 mAA on WxBS, +21.4 (1m, 10°) on InLoc, +24.2 AUC on RUBIK, and +12.4 mAA on IMC 2022. We release our code and models publicly at https://github.com/davnords/LoMa.

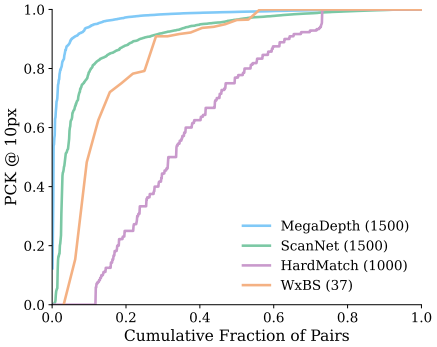
Keywords: Feature Matching · Structure-from-Motion · 3D Vision

1 Introduction

Structure-from-Motion (SfM) [23] aims to reconstruct the 3D world from unordered images and has long been a central problem in computer vision. A crucial part of SfM pipelines, typically referred to as *local feature matching*, is image matching through detection of sparse keypoints and description of their local appearance using high-dimensional representations, traditionally with *e.g.* SIFT [33], where correspondences are found by correlating the descriptions. To improve robustness and accuracy, neural network models have been introduced, both for detection and description, such as SuperPoint [12], ALIKED [68],

⋆ Equal contribution. Listing order is random.

(a) Matches on a pair from HardMatch.



(b) Distribution of pairs for different benchmarks successfully matched by SotA models.

Fig. 1: Revisiting local feature matching. We introduce HardMatch, a challenging hand-annotated matching benchmark, and LoMa, a fast and accurate family of local feature-based models. (a) LoMa successfully matches pairs from HardMatch where LightGlue fails, (b) HardMatch is significantly harder than previous benchmarks.

and DeDoDe [16], and for sparse matching with models such as SuperGlue [46] and LightGlue [32]. This paradigm yields fast and accurate matches and remains widely popular. While still heavily used in practice, local feature matching has recently been overshadowed in the literature by the advent of detector-free methods such as LoFTR [50] and RoMa [19], and feed-forward reconstruction models such as MASt3R [29] and VGGT [60] that are typically trained on orders of magnitude more data than their local feature matching counterparts. In the context of detector-free methods, it is often argued that detector-based local feature matching is fundamentally limited [50], and a significant amount of research has gone into how to scale detector-free SfM [13, 20, 24, 28], in order to overcome these supposed limitations. We argue that *the reports of the death of the local feature matcher are greatly exaggerated*.

In this paper, we revisit local feature matching from a data-driven perspective. In particular, we focus on (i) curating a large and diverse training data mixture together with scalable training recipes for both descriptors and matchers, and (ii) increasing training compute along two axes: data scale (the number and diversity of image pairs) and model capacity (the number of parameters). As we demonstrate through extensive experiments and ablations, these changes lead to substantial improvements in matching performance across a wide range of benchmarks. Our models outperform prior local feature methods by large margins and, in several settings, are competitive with or even surpass recent dense matching and feed-forward reconstruction pipelines. Figure 1a shows a qualitative example of a very challenging case that our matcher solves.

To meaningfully assess progress in matching capabilities and guide future research, well-designed evaluations and benchmarks are essential. Historically, improvements in feature matching have been measured on datasets derived from

SfM reconstructions, such as MegaDepth [30]. However, as we show in Fig. 1b, many of these benchmarks are now close to saturation: for a large fraction of image pairs, modern state-of-the-art matchers already recover a high percentage of correct correspondences. When benchmarks saturate, further improvements become difficult to observe, even when models meaningfully improve in robustness or generalization. This obscures remaining failure modes and risks encouraging overfitting to benchmark-specific artifacts, such as particular geometric verification settings, rather than advancing fundamental matching capability. To clearly measure progress, more challenging and diverse benchmarks are required. However, existing difficult image matching benchmarks, such as WxBS [38], are too small to reliably measure model improvements.

To address these limitations, we manually annotate image correspondences for a collection of 1000 pairs from 100 different categories, which we call **HardMatch**. The dataset is organized into 9 challenging groups spanning diverse and extreme matching scenarios. We find that feed-forward reconstruction methods largely fail on this benchmark, and even SotA dense matchers struggle. In a *return of the local feature matcher*, we demonstrate that our family of models, **LoMa**, can achieve performance even surpassing dense methods (and greatly outperforming sparse methods) by training on more diverse data with modern training recipes and increased compute. Our models, LoMa-{B(ase), L(arge), G(igantic)}, set a strong baseline for future progress in feature matching.

Our main contributions can be summarized as:

1. We revisit local feature matching from a modern, data-driven perspective (Sec. 3), introducing new training datasets with MVS-generated ground-truth and training recipes that we will make publicly available.
2. We introduce **HardMatch**, a challenging benchmark of 1000 hand-labeled image pairs that is lightweight yet large and difficult enough to provide meaningful signal for future research. We additionally report a human baseline based on independent annotators (Sec. 4).
3. We release a fast and accurate family of descriptor-matcher models that achieve SotA performance on HardMatch (+18.6mAA over LightGlue) and strong results across more than ten established matching and visual localization benchmarks. Extensive evaluations and ablations are provided in Sec. 5.

2 Related Work

Feature Matching. Finding pixel correspondences between two images is a fundamental task in 3D computer vision. Traditionally, image matching has been done in three stages: (i) keypoint detection, (ii) local feature description, and (iii) nearest neighbor matching in feature space. Learning-based approaches for keypoint detection [8, 15, 17, 39, 56], description [5, 16, 22, 52], as well as joint detection and description [12, 42, 55, 58, 68, 69], have been proposed to replace handcrafted methods such as SIFT [36] and ORB [44]. SuperGlue [46] proposed replacing the nearest neighbor matcher with a graph attention network, allowing global reasoning on local keypoint descriptors. Subsequently, LightGlue [32]

introduced a layer-wise loss and improved speed through pruning and early-stopping. Detector-free matching, first introduced in LoFTR [50], in contrast to sparse matching, eliminates keypoints. Matching benchmarks [11, 30, 38] have, since DKM [14], been topped by dense matchers [18, 19, 67], which match every pixel. Learning-based SfM methods, commonly referred to as *feed-forward reconstruction*, often include matching objectives. Notably, VGGT [60] uses a tracking head and MASt3R [29] combines pointmap regression with detector-free local features. In this work, our contribution is not the development of a novel matcher or descriptor. Instead, we use the existing DeDoDe descriptor, DaD [17] keypoints, and LightGlue matcher, with our proposed modern training recipe and our large-scale curated datasets. We show that using our approach, we can greatly surpass the performance of the original models.

Matching Evaluation. Feature matchers are commonly evaluated through relative pose estimation on sparse views from successful 3D reconstructions, such as MegaDepth [30] and ScanNet [11], or visual localization [3, 26, 47, 51]. These methods generally use pre-existing 3D reconstructions with localized query images to construct the benchmark. While this enables directly evaluating the estimated pose, the requirement for successful localization means that matching the query images is already solvable with existing systems. Thus, both categories are mostly saturated. In a similar vein, the Image Matching Challenge (IMC) [25] is a yearly challenge that aims to test the limits of reconstruction methods, with a hidden test set of ground-truth (GT) reconstructions. While IMC challenges are typically less saturated, evaluating matchers on them is typically a complex task, as there is no standardization, any reconstruction method is allowed, and SfM-pipelines involve a large number of hyperparameters.

In contrast, some previous benchmarks forgo mapping, and instead evaluate using only GT correspondences. One such work is WxBS [38], which consists of manually collected and labeled challenging pairs. Instead of evaluating the error in estimated relative pose, they use the epipolar error of the GT correspondences under a Fundamental matrix, which is estimated using correspondences from the matcher. However, its small size (37 pairs) limits its usefulness in model comparison. Besides this, we also find signs of this benchmark being near saturation (*cf.* Table 2 and Fig. 7). Our work takes a similar approach to WxBS, but includes more than 25 times as many pairs, with much higher diversity and difficulty.

3 Training the LoMa Descriptor and Matcher

In this section, we detail the training of the LoMa descriptor and matcher (*cf.* Fig. 2), based on DeDoDe [16] and LightGlue [32], respectively.

3.1 Problem Formulation

The aim in two-view matching is to obtain correct keypoint correspondences between two images $I^{\mathcal{A}}$ and $I^{\mathcal{B}}$. We follow a common three-stage approach, where

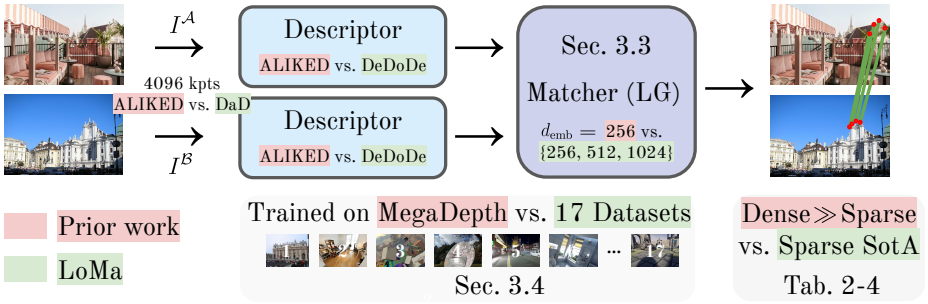


Fig. 2: The LoMa pipeline. By replacing ALIKED [68] with DaD [17]+DeDoDe [16] and training the descriptor and matcher on a large collection of datasets we achieve SotA results, even surpassing dense matchers on some tasks (*e.g.* HardMatch).

first keypoints $x_i^{\mathcal{A}}$ and $x_j^{\mathcal{B}}$ are detected in the images, second the keypoints are assigned descriptions $f_i^{\mathcal{A}}$ and $f_j^{\mathcal{B}}$, and third the descriptions are matched between the two images. The keypoints are assigned descriptions using a neural network called the *descriptor* g_θ, after which the descriptions are matched using a second neural network called the *matcher* h_ϕ.

3.2 Learning Objective

In this work, we do not train a detector. Instead, we use DaD to supervise both the descriptor (g_θ) and matcher (h_ϕ). We compare DaD to other detectors and an ensemble in Tab. 6 in the supplementary. We first train g_θ, followed by h_ϕ with g_θ frozen. The number of keypoints in each image during training is $N = 2048$.

Ground truth (GT) correspondences are obtained via known relative poses and depth maps in the training datasets. We denote the GT matches by $\mathcal{M}^{\mathcal{A},\mathcal{B}} = \{(i,j) \mid x_i^{\mathcal{A}} \text{ and } x_j^{\mathcal{B}} \text{ match}\}$.

Description. For training the descriptor g_θ, we follow DeDoDe [16] and use a dual-softmax based loss. Descriptions of each keypoint are obtained as $f_i^{\mathcal{A}} = g_\theta(x_i^{\mathcal{A}}, I^{\mathcal{A}}), f_j^{\mathcal{B}} = g_\theta(x_j^{\mathcal{B}}, I^{\mathcal{B}}) \in \mathbb{R}^{d_{\text{desc}}}$, and a description similarity matrix is defined per image pair as

$$S_{ij} = {f_i^{\mathcal{A}}}^\top f_j^{\mathcal{B}}. \tag{1}$$

The loss per image pair is given by

$$\mathcal{L}_{\text{desc}} = -\left(\sum_{(i,j)\in\mathcal{M}^{\mathcal{A},\mathcal{B}}} \log \text{softmax}_i(\tau^{-1} S_{ij}) + \log \text{softmax}_j(\tau^{-1} S_{ij}) \right), \tag{2}$$

where τ^{-1} is the inverse temperature, a hyperparameter. The loss encourages the dual-softmax matrix (also called soft assignment matrix)

$$\mathcal{P}_{ij} = \text{softmax}_i(\tau^{-1} S_{ij}) \odot \text{softmax}_j(\tau^{-1} S_{ij}) \tag{3}$$

to have the GT matches as maxima along both i and j. As noted in [29], (2) can be viewed as a form of infoNCE-loss applied over the GT correspondences.

Matching. For training the matcher, we follow LightGlue [32]. The matcher h_ϕ takes keypoints and descriptions for two images as input and outputs refined descriptions $\tilde{f}$ for each keypoint, which now depend on both images:

$$\left(\left(\tilde{f}_i^{\mathcal{A}}\right)_{i=1}^{N}, \left(\tilde{f}_j^{\mathcal{B}}\right)_{j=1}^{N}\right) = h_\phi\left(\left(x_i^{\mathcal{A}}, f_i^{\mathcal{A}}\right)_{i=1}^{N}, \left(x_j^{\mathcal{B}}, f_j^{\mathcal{B}}\right)_{j=1}^{N}\right). \tag{4}$$

In each layer of h_ϕ, we use the dual-softmax loss (2) on the refined features, passed through a linear head, along with a separate matchability loss. A separate linear head with softmax activation predicts a matchability score for each keypoint. We supervise this prediction using a binary cross-entropy loss with the ground-truth matchability. A keypoint is defined as matchable if it has a match in the ground truth set $\mathcal{M}^{\mathcal{A},\mathcal{B}}$. Layer-wise supervision allows trading performance for speed at inference time. We study this trade-off in Sec. 5.5.

3.3 Architecture

Our descriptor follows the DeDoDe [16] architecture, while the matcher is based on LightGlue [32]. Input keypoints and descriptors are processed through L identical blocks of self- and cross-attention, progressively refining the descriptors. When the descriptor dimension (d_{desc}) differs from the matcher embedding dimension (d_{emb}), we apply a learned linear projection.

Self-attention is applied by each point attending to all points of the same image, while in cross-attention each point attends to all points of the other image. Rotary position embeddings (RoPE) [49] are used in the self-attention computation, making the attention scores dependent on the relative positions $x_i - x_{i'}$. Positional embeddings are not used in the cross-attention computation.

At inference, we use the descriptions output from the last layer to define a dual-softmax matrix $\mathcal{P}_{ij}$ as in (3). A correspondence (i, j) is registered when $\mathcal{P}_{ij}$ represents a maximum along both the rows and columns, *i.e.* the match is mutual. We discard matches for which $\mathcal{P}_{ij} < \mu$ with $\mu = 0.1$.

We release three main variants of the LoMa matcher, B, L, and G, with progressively increasing size. All variants share the same architecture, consisting of $L = 9$ transformer blocks that alternate between self-attention and cross-attention layers, and use attention heads of dimension 64 throughout. They differ only in their embedding dimensionality, which is 256, 512, and 1024, respectively. We also release B^{128}, using the lighter descriptor DeDoDe-B, instead of G, with $d_{\text{desc}} = 128$, providing a lightweight set of features for *e.g.* visual localization.

3.4 Training Data

Our data mixture, presented in Tab. 1, is inspired by RoMa v2 [18] and UFM [67] by incorporating both wide baseline and optical flow datasets. Compared to prior

matchers such as LightGlue, which was pretrained on synthetic homographies and fine-tuned on MegaDepth, our training data is significantly more diverse. In addition to the datasets used in RoMa v2, we add Aria Synthetic Environments [4], CO3Dv2 [41], MPSD [1], MegaDepth (Re-MVS), MegaScenes [54], MegaSynth [27], and SpatialVID [59]. The large data collection leads to a near 10-point improvement on HardMatch (*cf*. Tab. 5). For three of these datasets we compute 3D ground truth beyond the original data. We will make the data and code for these datasets, which we provide further details on below, public.

MegaDepth (Re-MVS): We run COLMAP [48] MVS (photometric+geometric, default settings) on all scenes, additionally including reconstructions skipped in MegaDepth (all sparse models beyond the first reconstruction).

MegaScenes: MegaScenes contains a large number of scenes. However, we find that many of these are not of sufficient quality or size to constitute good training data. We select a subset of reconstructions and from these filter out a total of 303 scenes with a sufficiently large number of cameras and 3D points. For these scenes we run standard COLMAP MVS, similarly as for MegaDepth (Re-MVS).

SpatialVID: While SpatialVID provides 3D annotations, we find them to be insufficiently accurate for feature matching. We therefore select a subset comprising 59 scenes, and run COLMAP SfM (using SIFT+DaD keypoints with RoMa v2 correspondences) with shared intrinsics. As most scenes contain dominant forward motion, we do not filter initial pairs on forward motion, as this commonly led to reconstructions failing. We implement a custom MVS pipeline using RoMa v2 correspondences with a simple native PyTorch [40] PatchMatch implementation to compute depth maps.

3.5 Training

For all training, we use the AdamW [35] optimizer, the data mix outlined in Tab. 1, and a fixed resolution of 560×560. We use a cosine annealing learning rate with a peak learning rate of 2×10^{-4} and a global batch size of 64. We use a slight weight decay of 5×10^{-5} and use Exponential Moving Average (EMA) with a decay factor of $\alpha = 0.999$. We train the descriptor for 50K steps, which takes approximately one day on $8 \times$ A100:40GB. We show in the supplementary (*cf*. Fig. 8a) that training the descriptor for longer does not help. We train the matcher for 250K steps. Sizes B and L are trained on $8 \times$ A100:40GB while G is trained on $16 \times$ A100:40GB, each taking around two days.

4 HardMatch

In this section, we introduce HardMatch, an extremely challenging image matching benchmark divided into 9 groups (*cf*. Fig. 3). We detail data collection

Table 1: Training data. Unlike previous local feature matching methods [16,55] typically trained on MegaDepth [30], we scale our training to 17 3D datasets, approaching the data volume used in feedforward reconstruction.

Datasets	Type / GT Source	Weight
ScanNet++ v2 [66]	Indoor / Mesh	1
BlendedMVS [65]	Aerial / Mesh	1
Map-Free [3]	Object-centric / MVS	1
Hypersim [43]	Indoor / Graphics	1
MegaScenes [54]	Outdoor / MVS	1
MegaDepth [30]	Outdoor / MVS	1
MegaDepth (Re-MVS)	Outdoor / MVS	1
AerialMD [57]	Aerial / MVS	1
TartanAir v2 [62]	Outdoor / Graphics	1
Mapillary Planet-scale Depth [1]	Driving / MVS	0.1
Aria Synthetic Environments [4]	Indoor / Graphics	0.1
CO3Dv2 [41]	Object-centric / MVS	0.1
MegaSynth [27]	Indoor / Graphics	0.1
SpatialVID [59]	Forward-motion / MVS	0.01
FlyingThings3D [37]	Outdoor / Graphics	0.5
UnrealStereo4k [53]	Outdoor / Graphics	0.01
Virtual KITTI 2 [9,21]	Outdoor / Graphics	0.01

(Sec. 4.1), evaluation (Sec. 4.2), and qualitative characteristics (Sec. 4.3). We will release the benchmark publicly under a permissive license.

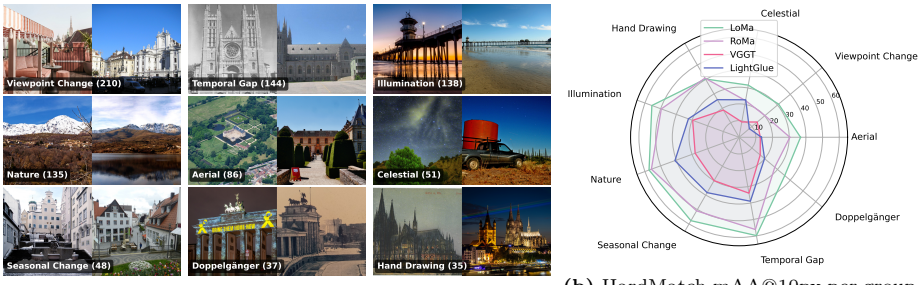
4.1 Data Collection

The main steps in the data collection process are: (i) identifying candidate images online, (ii) selecting difficult pairs using a matching model with manual annotation, and (iii) manual keypoint annotation. We detail the steps below.

Finding Images. We begin by identifying a large corpus of candidate images by scraping 100 categories of Wikimedia Commons under permissive licenses. The categories were chosen to provide high diversity. We illustrate categories in the test set in Fig. 12. The methodology is inspired by MegaScenes [54].

Identifying Difficult Pairs. To filter the large image collection into difficult matching pairs, we randomly sample 100 pairs per scene and use the confidence map of RoMa v2 [18] to identify difficult pairs. We select pairs where RoMa v2 is uncertain by thresholding the maximum confidence between 0.3 and 0.9, which provides a good balance of difficult pairs while still having some overlap. We manually inspect each identified pair and classify it as matchable or unmatchable, proceeding until we identify 10 pairs per category. The categories are randomly split into a validation set (10 categories) and a test set (90 categories).

Annotating Keypoints. We manually annotate corresponding keypoints in each pair, identifying as many salient matches as possible. Each pair contains between 8 and 28 annotated correspondences. The resulting dataset, which we



(a) Example pairs per group (#pairs in parentheses). **(b)** HardMatch mAA@10px per group.

Fig. 3: HardMatch groups. The dataset contains image pairs from a wide range of challenging scenarios, organized into 9 groups. (a) Example pairs illustrating each group. (b) HardMatch mAA@10px performance per group.

call HardMatch, consists of 1000 image pairs from all over the world. We illustrate the geographic and temporal distribution in Fig. 9 in the supplementary. To provide more granular insights, we group pairs into the labels shown qualitatively in Fig. 3a. The smallest group is roughly equal to WxBS in size.

To verify our keypoint annotations, we provide a human baseline and estimate the ground truth error using independent annotators. Eight independent annotators are each assigned 20 pairs to verify. Annotators are asked to match a random keypoint in the first image with an arbitrary pixel location in the second image. We record the pixel error distribution and include a curve in Fig. 4.

4.2 Evaluation

Following WxBS, we evaluate by estimating a Fundamental matrix F using correspondences from the matcher and computing the epipolar error of the GT correspondences. We compute the percentage correct keypoints (PCK) under different pixel error thresholds. Each pair contributes equally to the PCK, regardless of number of keypoints. More details on the evaluation can be found in the supplementary (Sec. C.1) as well as an alternative evaluation methodology directly using the GT keypoints (Sec. C.2).

4.3 Qualitative Characteristics

HardMatch is an image matching dataset specifically designed to capture extreme appearance variations. Many pairs consist of images taken under substantially different conditions. The dataset includes examples such as aerial versus ground views, images captured over a century apart, hand-drawn sketches paired with natural photographs, night–day transitions, seasonal changes, and viewpoint differences of up to 180 degrees. Selected examples illustrating this diversity are shown in Fig. 3a. More pairs are found in the supplementary.

5 Experiments

We compare the LoMa family of models to a wide range of sparse, dense, and feed-forward reconstruction methods on a large collection of benchmarks for extreme matching (Sec. 5.1), relative pose estimation (Sec. 5.2), visual localization (Sec. 5.3), and more (Sec. 5.4). Finally, we give insights into ablations, throughput, and scaling (Sec. 5.5). All evaluations use $N = 4096$ keypoints. See supplementary Tab. 7 for varying number of keypoints. Additional experiments (Sec. A) and details (Sec. B) can be found in the supplementary. We use the abbreviations SG (SuperGlue), LG (LightGlue), and SP (SuperPoint) throughout.

5.1 Extreme Matching

WxBS. We evaluate on the challenging matching benchmark WxBS [38]. The benchmark features 37 pairs of hand-labeled correspondences that display a mix of extreme changes in viewpoint, illumination, and modality. We report the mean accuracy in Tab. 2, and the accuracy as a function of the threshold in the supplementary (*cf*. Fig. 7). LoMa-G achieves SotA results on WxBS, barely beating RoMa (73.4 vs. 72.6) while handily beating other sparse matchers.

HardMatch. We report the performance on HardMatch for: (i) groups (Fig. 3b), (ii) pixel thresholds (Fig. 4), and (iii) a wide range of matchers (Tab. 2). The complete results are in the supplementary (Tab. 14). We include a human baseline as a reference (described in Sec. 4.1). However, the numbers are not directly comparable, as the matchers are evaluated through their estimated Fundamental matrix F, while the human baseline is directly evaluated on the correspondences. We find HardMatch to be challenging for SotA matchers. LoMa-G achieves the best result of 54.3 mAA@10px, approximately 20 points below its performance on WxBS. Doppelgängers [10, 64], large viewpoint changes, aerial photographs, and star constellations are particularly challenging for all matchers. We illustrate some qualitative examples in Fig. 10 in the supplementary.

5.2 Relative Pose Estimation

We compare LoMa to SotA matchers, detection+description with mutual nearest neighbor matching, and feed-forward reconstruction methods on relative pose estimation. We report the results on MegaDepth-1500 [30, 50] and ScanNet-1500 [11, 46] in Tab. 2. LoMa significantly outperforms other sparse matchers on both datasets. In particular, LoMa-L achieves gains of 8.4 and 12.9 AUC@5° compared to other sparse matchers on MegaDepth and ScanNet, respectively.

5.3 Visual Localization

Map-free. The map-free relocalization benchmark [3] tests the ability to localize the camera in metric space given a single reference image and no map. To

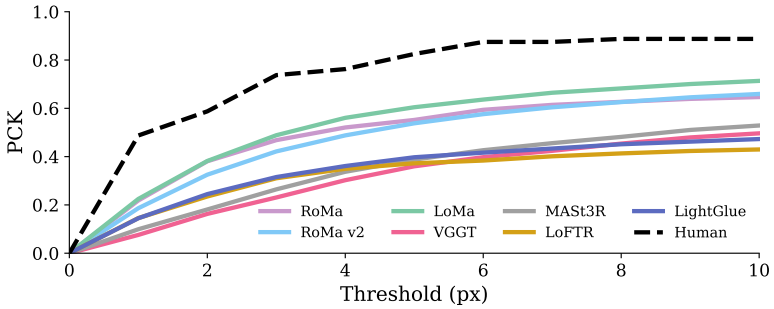


Fig. 4: HardMatch accuracy at different thresholds. LoMa performs slightly better than the best dense matchers and significantly outperforms LightGlue.

obtain monocular metric depth, we use DA3 [31]. Following the benchmark, we use the Virtual Correspondence Reprojection Error (VCRE<90px) and report the results for the validation set in Tab. 3. LoMa-G achieves a ≈20-point increase in precision against other sparse matchers.

InLoc. We evaluate visual localization on InLoc [51] using the HLoc [45] pipeline and report the results in Tab. 3. We find that LoMa significantly outperforms other sparse matchers. Most notably, LoMa-G achieves a more than 20-point increase over the second best matcher on the most narrow threshold for DUC2.

Oxford Day-and-Night. We evaluate visual localization under challenging lighting conditions on the Oxford Day-and-Night [63] dataset. In contrast to InLoc, the evaluation requires the feature matcher to construct an SfM model using the daytime database image. We use the HLoc pipeline and report the median result for night queries in Tab. 3 and individual scenes in Tab. 10 in the supplementary. For the most narrow threshold, LoMa-G achieves a more than 14-point increase in accuracy compared to other sparse matchers.

5.4 Additional Matching Evaluations

RUBIK. We further evaluate on the newly released RUBIK [34] benchmark and report the results in Tab. 4. We find that LoMa-G outperforms other sparse matchers, notably improving both AUC at 10° and 20° by ≈ 24 points.

Image Matching Challenge 2022. The Image Matching Challenge (IMC) is a yearly competition held at CVPR. The 2022 version [25] consists of a hidden test-set of Google street-view images with the task to estimate the Fundamental matrix between them. Table 4 presents the results of our submission. LoMa sets a new SoTA, handily beating other sparse matchers. LoMa also beats the competition winner from 2022 that used an ensemble of LoFTR, DKM, and SuperGlue as well as RoMa which achieved a score of 86.3 and 88.0, respectively.

Table 2: SotA matching comparison. Relative pose estimation on MegaDepth-1500 [30, 50] and ScanNet-1500 [11, 46] and accuracy on WxBS [38] and HardMatch.

Method	MegaDepth			ScanNet				WxBS	HM
AUC@ →	5°	10°	20°	5°	10°	20°	mAA@ →	10px	10px
Feed-forward Reconstruction									
MASt3R [29] ECCV'24	**42.4**	**61.5**	**76.9**	33.6	**56.8**	**74.1**		34.5	**33.6**
VGGT [60] CVPR'25	33.5	52.9	70.0	**33.9**	55.2	73.4		**36.3**	28.4
Dense Matchers									
LoFTR [50] CVPR'21	52.8	69.2	81.2	22.1	40.8	57.6		50.7	33.1
RoMa [19] CVPR'24	62.6	76.7	86.3	31.8	53.4	70.9		**72.6**	**48.1**
UFM [67] NeurIPS'25	41.5	57.9	72.4	31.3	54.1	72.0		53.3	33.9
RoMa v2 [18]	**62.8**	**77.0**	**86.6**	**33.6**	**56.2**	**73.8**		64.8	46.5
Detect+Describe, 4096 Keypoints									
DISK [55] NeurIPS'20	35.0	51.4	64.9	6.4	13.9	23.2		21.9	22.0
ALIKED [68] TIM'23	41.9	58.4	71.7	6.7	14.6	25.0		35.1	26.6
DeDoDe-G [16] 3DV'24	44.6	61.8	75.7	13.5	27.3	41.9		46.4	30.3
LoMa Desc. (ours)	**51.7**	**68.3**	**80.9**	**18.7**	**37.2**	**55.6**		**63.0**	**39.5**
Sparse Matchers, 4096 Keypoints									
SP+SG [12, 46]	43.7	61.8	76.5	16.4	32.5	49.0		45.6	36.0
SP+LG [12, 32]	43.8	61.8	76.4	15.9	32.1	48.9		40.4	34.8
DISK+LG [32, 55]	47.8	65.3	79.0	9.3	19.3	30.8		39.2	30.3
ALIKED+LG [32, 68]	48.1	65.7	79.3	14.5	28.9	43.5		43.9	35.7
LoMa-B^{128} (ours)	55.1	71.2	83.2	24.8	45.5	63.7		61.5	48.2
LoMa-B (ours)	55.7	71.8	83.6	27.5	49.7	68.2		68.7	51.1
LoMa-L (ours)	**56.5**	**72.7**	**84.3**	**29.3**	**51.9**	**70.3**		70.6	53.5
LoMa-G (ours)	56.1	72.2	84.0	**29.3**	51.7	70.0		**73.4**	**54.3**

5.5 Analysis

Ablations. We evaluate our design choices in Tab. 5 by performance on the validation set of HardMatch. Changing from ALIKED to DaD+DeDoDe and retraining on MegaDepth gives a moderate boost (+5.7). Extending the training data of the matcher and subsequently also the descriptor from MegaDepth to the full dataset gives further improvements in generalization (+9.1). We then extend the training from 50K steps to 250K steps (+0.4). Scaling the matcher embedding dimension d_{emb} from 256 to 1024 yields further improvements (+1.4).

Throughput. In SfM and visual localization, the matcher is the main bottleneck because it must process each pair of images, whereas detection and description are performed only once per image. The layer-wise loss allows the matcher to trade accuracy for speed via early stopping. We analyze this trade-off in Fig. 5 by evaluating different stopping layers ($L = \{3, 5, 9\}$). The LoMa-B matcher has the same runtime as LG while producing significantly more accurate matches.

Table 3: Visual localization. Comparison on Map-free [3], InLoc [51], and Oxford Day-and-Night [63]. On the two latter, we report the percentage of query images correctly localized within (0.25m, 2°) / (0.5m, 5°) / (1m, 10°).

Method	Map-free		InLoc		Oxford
	Prec.	AUC	DUC1	DUC2	Night
SP+SG	46.3	74.1	46.5/65.7/78.3	52.7/72.5/79.4	44.3/54.4/58.0
SP+LG	45.5	76.2	43.9/64.6/76.8	42.7/68.7/74.0	43.4/53.5/57.7
DISK+LG	43.2	60.7	43.4/60.6/74.2	36.6/53.4/67.2	14.8/17.5/20.1
ALIKED+LG	47.2	79.5	41.4/64.6/79.8	35.9/64.1/67.9	42.8/53.9/58.9
LoMa-B^{128} (ours)	60.8	87.0	54.5/76.8/87.9	64.1/82.4/84.7	54.8/62.2/67.1
LoMa-B (ours)	65.6	89.0	57.1/**80.8**/**91.9**	71.0/87.0/88.5	54.7/62.4/66.2
LoMa-L (ours)	67.6	89.4	**59.1**/**80.8**/**91.9**	71.0/84.0/87.8	56.0/64.6/69.2
LoMa-G (ours)	**68.9**	**90.3**	55.6/80.3/91.4	**73.3**/**87.8**/**89.3**	**58.9**/**66.0**/**69.7**

Table 4: Additional evaluations. Comparison by AUC on RUBIK [34] and mAA on Image Matching Challenge 2022 [25].

Method	RUBIK		IMC 2022
	@10°	@20°	@10°
SP+SG	46.2	54.1	72.4
SP+LG	44.8	52.1	69.2
DISK+LG	40.8	46.0	74.2
ALIKED+LG	49.0	55.2	76.9
LoMa-B^{128} (ours)	67.7	75.2	85.5
LoMa-B (ours)	65.7	72.2	87.4
LoMa-L (ours)	69.1	76.1	89.0
LoMa-G (ours)	**73.2**	**79.9**	**89.3**

On an A100, LoMa-B can hit hundreds of pairs per second with 2048 keypoints and 16 pairs in each batch. We show the results for a single pair in each batch in Fig. 8b in the supplementary, which results in different model sizes being more similar in speed on modern GPUs due to poor utilization. For a single image pair, the LoMa-B matcher with $L = 3$ runs at almost 300 pairs per second.

Scaling Local Feature Matching. We find that local feature matchers benefit significantly from training on additional data (*cf*. Fig. 6a) and increasing the model size (*cf*. Fig. 6b). This is also illustrated through ablations in Tab. 5.

6 Limitations

Despite the strong empirical performance, several limitations remain.

– Scaling the sparse matcher works well in our experiments, but large-scale descriptor training tends to overfit, see Suppl. Fig. 8a.

Table 5: Ablations. Performance on the validation set of HardMatch (HM).

Method	HM
mAA@ →	10px
I: ALIKED+LG (Baseline)	36.3
II: DaD+DeDoDe+LG	42.0
III: Matcher → All data	48.9
IV: Descriptor → All data	51.1
V: Longer training (LoMa-B)	51.5
VI: LoMa-L	52.8
VII: LoMa-G	**52.9**

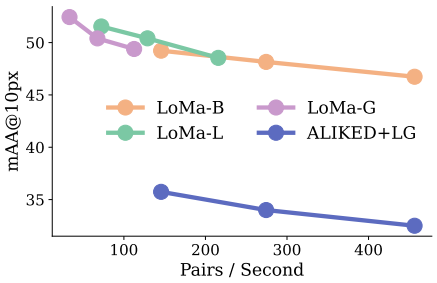
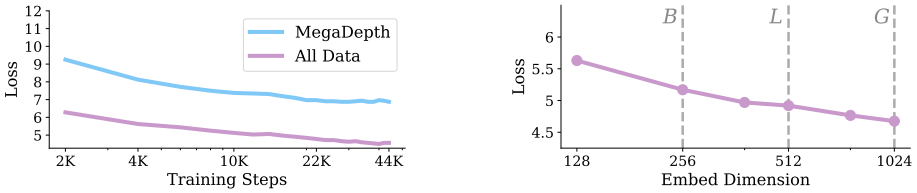


Fig. 5: Pareto curve. HardMatch performance as a function of inference speed (A100) for different stopping layers.



(a) Data scale. **(b)** Model capacity.

Fig. 6: Increased data scale and model capacity. Both axes of scaling, (a) data and (b) model size, lead to significant reductions in validation loss on HardMatch.

- LoMa outperforms previous methods, but still struggles on challenging HardMatch subgroups, such as Doppelgängers and extreme viewpoint changes.
- HardMatch, similarly to WxBS, relies on human-annotated keypoints. The evaluation protocol based on F estimation alleviates this issue, but it requires static scenes and perspective cameras.
- Although HardMatch is more diverse than previous benchmarks, it still contains geographic and temporal biases, see Suppl. Figs. 9a and 9b.

7 Conclusion

We revisit the classical problem of local feature matching and show that combining large-scale data with modern practices yields substantial performance gains. To support this, we introduce (i) HardMatch, a highly challenging evaluation dataset consisting of 1000 hand-labeled image pairs, and (ii) LoMa, a family of models achieving SotA performance on this new benchmark as well as on the established benchmarks IMC 2022 and WxBS, surpassing even dense matchers.

Acknowledgements

This work was supported by the Wallenberg Artificial Intelligence, Autonomous Systems and Software Program (WASP), funded by the Knut and Alice Wallenberg Foundation, and by the strategic research environment ELLIIT, funded by the Swedish government. The computational resources were provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) at C3SE, partially funded by the Swedish Research Council through grant agreement no. 2022-06725, and by the Berzelius resource, provided by the Knut and Alice Wallenberg Foundation at the National Supercomputer Centre.

References

1. Antequera, M.L., Gargallo, P., Hofinger, M., Bulo, S.R., Kuang, Y., Kontschieder, P.: Mapillary planet-scale depth dataset. In: ECCV (2020) 7, 8
2. Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: Netvlad: Cnn architecture for weakly supervised place recognition. In: CVPR (2016) 2
3. Arnold, E., Wynn, J., Vicente, S., Garcia-Hernando, G., Monszpart, A., Prisacariu, V., Turmukhambetov, D., Brachmann, E.: Map-free visual relocalization: Metric pose relative to a single image. In: ECCV (2022) 4, 8, 10, 13
4. Avetisyan, A., Xie, C., Howard-Jenkins, H., Yang, T.Y., Aroudj, S., Patra, S., Zhang, F., Frost, D., Holland, L., Orme, C., Engel, J., Miller, E., Newcombe, R., Balntas, V.: Scenescript: Reconstructing scenes with an autoregressive structured language model. In: Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., Varol, G. (eds.) ECCV (2025) 7, 8
5. Balntas, V., Lenc, K., Vedaldi, A., Mikolajczyk, K.: HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In: CVPR (2017) 3, 1, 2
6. Barath, D., Matas, J., Noskova, J.: MAGSAC: marginalizing sample consensus. In: CVPR (2019) 5, 8
7. Barath, D., Noskova, J., Ivashechkin, M., Matas, J.: Magsac++, a fast, reliable and accurate robust estimator. In: CVPR (2020) 5, 8
8. Barroso-Laguna, A., Riba, E., Ponsa, D., Mikolajczyk, K.: Key.Net: Keypoint detection by handcrafted and learned cnn filters. In: ICCV (2019) 3
9. Cabon, Y., Murray, N., Humenberger, M.: Virtual kitti 2. arXiv preprint arXiv:2001.10773 (2020) 8
10. Cai, R., Tung, J., Wang, Q., Averbuch-Elor, H., Hariharan, B., Snavely, N.: Doppelgangers: Learning to disambiguate images of similar structures. In: ICCV (2023) 10
11. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: CVPR (2017) 4, 10, 12
12. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-supervised interest point detection and description. In: CVPR (2018) 1, 3, 12, 9
13. Duisterhof, B.P., Zust, L., Weinzaepfel, P., Leroy, V., Cabon, Y., Revaud, J.: Mast3r-sfm: a fully-integrated solution for unconstrained structure-from-motion. In: 3dv (2025) 2
14. Edstedt, J., Athanasiadis, I., Wadenbäck, M., Felsberg, M.: DKM: Dense kernelized feature matching for geometry estimation. In: CVPR (2023) 4
15. Edstedt, J., Bökman, G., Zhao, Z.: Dedode v2: Analyzing and improving the dedode keypoint detector. In: CVPRW (2024) 3, 1

16. Edstedt, J., Bökman, G., Wadenbäck, M., Felsberg, M.: DeDoDe: Detect, Don't Describe – Describe, Don't Detect for Local Feature Matching. In: 3dv (2024) 2, 3, 4, 5, 6, 8, 12
17. Edstedt, J., Bökman, G., Wadenbäck, M., Felsberg, M.: Dad: Distilled reinforcement learning for diverse keypoint detection. arXiv preprint arXiv:2503.07347 (2025) 3, 4, 5, 1
18. Edstedt, J., Nordström, D., Zhang, Y., Bökman, G., Astermark, J., Larsson, V., Heyden, A., Kahl, F., Wadenbäck, M., Felsberg, M.: Roma v2: Harder better faster denser feature matching (2025), https://arxiv.org/abs/2511.15706 4, 6, 8, 12, 2, 3, 9
19. Edstedt, J., Sun, Q., Bökman, G., Wadenbäck, M., Felsberg, M.: RoMa: Robust dense feature matching. In: CVPR (2024) 2, 4, 12, 9
20. Elflein, S., Zhou, Q., Leal-Taixé, L.: Light3r-sfm: Towards feed-forward structure-from-motion. In: CVPR (2025) 2
21. Gaidon, A., Wang, Q., Cabon, Y., Vig, E.: Virtual worlds as proxy for multi-object tracking analysis. In: CVPR (2016) 8
22. Germain, H., Bourmaud, G., Lepetit, V.: S2DNet: learning image features for accurate sparse-to-dense matching. In: ECCV (2020) 3
23. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003) 1
24. He, X., Sun, J., Wang, Y., Peng, S., Huang, Q., Bao, H., Zhou, X.: Detector-free structure from motion. In: CVPR (2024) 2
25. Howard, A., Trulls, E., Yi, K.M., Mishkin, D., Dane, S., Jin, Y.: Image matching challenge 2022 (2022), https://kaggle.com/competitions/image-matching-challenge-2022 4, 11, 13
26. Jafarzadeh, A., Antequera, M.L., Gargallo, P., Kuang, Y., Toft, C., Kahl, F., Sattler, T.: Crowddriven: A new challenging dataset for outdoor visual localization. In: ICCV (2021) 4
27. Jiang, H., Xu, Z., Xie, D., Chen, Z., Jin, H., Luan, F., Shu, Z., Zhang, K., Bi, S., Sun, X., Gu, J., Huang, Q., Pavlakos, G., Tan, H.: Megasynth: Scaling up 3d scene reconstruction with synthesized data. In: CVPR (2025) 7, 8
28. Lee, J., Yoo, S.: Dense-sfm: Structure from motion with dense consistent matching. In: CVPR (2025) 2
29. Leroy, V., Cabon, Y., Revaud, J.: Grounding image matching in 3d with mast3r. In: ECCV (2024) 2, 4, 6, 12, 9
30. Li, Z., Snavely, N.: Megadepth: Learning single-view depth prediction from internet photos. In: CVPR (2018) 3, 4, 8, 10, 12
31. Lin, H., Chen, S., Liew, J.H., Chen, D.Y., Li, Z., Shi, G., Feng, J., Kang, B.: Depth anything 3: Recovering the visual space from any views. arXiv preprint arXiv:2511.10647 (2025) 11
32. Lindenberger, P., Sarlin, P.E., Pollefeys, M.: LightGlue: Local Feature Matching at Light Speed. In: ICCV (2023) 2, 3, 4, 6, 12, 9
33. Liu, C., Yuen, J., Torralba, A.: Sift flow: Dense correspondence across scenes and its applications. tpami **33**(5) (2010) 1
34. Loiseau, T., Bourmaud, G.: Rubik: A structured benchmark for image matching across geometric challenges. In: CVPR (2025) 11, 13
35. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019) 7
36. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV (2004) 3

37. Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: CVPR (2016) 8
38. Mishkin, D., Matas, J., Perdoch, M., Lenc, K.: WxBS: Wide baseline stereo generalizations. In: BMVC (2015) 3, 4, 10, 12, 5
39. Mishkin, D., Radenović, F., Matas, J.: Repeatability Is Not Enough: Learning Affine Regions via Discriminability. In: ECCV (2018) 3
40. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. In: NeurIPS (2019) 7
41. Reizenstein, J., Shapovalov, R., Henzler, P., Sbordone, L., Labatut, P., Novotny, D.: Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In: ICCV (2021) 7, 8
42. Revaud, J., De Souza, C., Humenberger, M., Weinzaepfel, P.: R2d2: Reliable and Repeatable Detector and Descriptor. In: NeurIPS (2019) 3
43. Roberts, M., Ramapuram, J., Ranjan, A., Kumar, A., Bautista, M.A., Paczan, N., Webb, R., Susskind, J.M.: Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In: ICCV (2021) 8
44. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: An efficient alternative to SIFT or SURF. In: ICCV (2011) 3
45. Sarlin, P.E., Cadena, C., Siegwart, R., Dymczyk, M.: From coarse to fine: Robust hierarchical localization at large scale. In: CVPR (2019) 11, 2
46. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: Superglue: Learning feature matching with graph neural networks. In: CVPR (2020) 2, 3, 10, 12, 9
47. Sattler, T., Maddern, W., Toft, C., Torii, A., Hammarstrand, L., Stenborg, E., Safari, D., Okutomi, M., Pollefeys, M., Sivic, J., Kahl, F., Pajdla, T.: Benchmarking 6dof outdoor visual localization in changing conditions. In: CVPR (2018) 4
48. Schönberger, J.L., Frahm, J.M.: Structure-from-Motion Revisited. In: CVPR (2016) 7
49. Su, J., Lu, Y., Pan, S., Murtadha, A., Wen, B., Liu, Y.: Roformer: Enhanced transformer with rotary position embedding (2023), https://arxiv.org/abs/2104.09864 6
50. Sun, J., Shen, Z., Wang, Y., Bao, H., Zhou, X.: LoFTR: Detector-free local feature matching with transformers. In: CVPR (2021) 2, 4, 10, 12, 1, 9
51. Taira, H., Okutomi, M., Sattler, T., Cimpoi, M., Pollefeys, M., Sivic, J., Pajdla, T., Torii, A.: Inloc: Indoor visual localization with dense matching and view synthesis. In: CVPR (2018) 4, 11, 13
52. Tian, Y., Yu, X., Fan, B., Wu, F., Heijnen, H., Balntas, V.: Sosnet: Second order similarity regularization for local descriptor learning. In: CVPR (2019) 3
53. Tosi, F., Liao, Y., Schmitt, C., Geiger, A.: Smd-nets: Stereo mixture density networks. In: CVPR (2021) 8
54. Tung, J., Chou, G., Cai, R., Yang, G., Zhang, K., Wetzstein, G., Hariharan, B., Snavely, N.: Megascenes: Scene-level view synthesis at scale. In: ECCV (2024) 7, 8
55. Tyszkiewicz, M., Fua, P., Trulls, E.: DISK: Learning local features with policy gradient. In: NeurIPS (2020) 3, 8, 12, 1, 9
56. Verdie, Y., Yi, K., Fua, P., Lepetit, V.: Tilde: A temporally invariant learned detector. In: CVPR (2015) 3
57. Vuong, K., Ghosh, A., Ramanan, D., Narasimhan, S., Tulsiani, S.: Aerialmegadepth: Learning aerial-ground reconstruction and view synthesis. In: CVPR (2025) 8

58. Wang, B., Chen, C., Cui, Z., Qin, J., Lu, C.X., Yu, Z., Zhao, P., Dong, Z., Zhu, F., Trigoni, N., Markham, A.: P2-net: Joint description and detection of local features for pixel and point matching. In: ICCV (2021) 3
59. Wang, J., Yuan, Y., Zheng, R., Lin, Y., Gao, J., Chen, L.Z., Bao, Y., Zhang, Y., Zeng, C., Zhou, Y., et al.: Spatialvid: A large-scale video dataset with spatial annotations. arXiv preprint arXiv:2509.09676 (2025) 7, 8
60. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer. In: CVPR (2025) 2, 4, 12, 9
61. Wang, Q.: Understanding and optimizing attention-based sparse matching for diverse local features (2026), https://arxiv.org/abs/2602.08430 1
62. Wang, W., Zhu, D., Wang, X., Hu, Y., Qiu, Y., Wang, C., Hu, Y., Kapoor, A., Scherer, S.: Tartanair: A dataset to push the limits of visual slam. In: iros (2020) 8
63. Wang, Z., Bian, W., Li, X., Tao, Y., Wang, J., Fallon, M., Prisacariu, V.A.: Seeing in the dark: Benchmarking egocentric 3d vision with the oxford day-and-night dataset. In: NeurIPS (2025) 11, 13, 2
64. Xiangli, Y., Cai, R., Chen, H., Byrne, J., Snavely, N.: Doppelgangers++: Improved visual disambiguation with geometric 3d features. In: CVPR (2025) 10
65. Yao, Y., Luo, Z., Li, S., Zhang, J., Ren, Y., Zhou, L., Fang, T., Quan, L.: Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In: CVPR (2020) 8
66. Yeshwanth, C., Liu, Y.C., Nießner, M., Dai, A.: Scannet++: A high-fidelity dataset of 3d indoor scenes. In: ICCV (2023) 8
67. Zhang, Y., Keetha, N., Lyu, C., Jhamb, B., Chen, Y., Qiu, Y., Karhade, J., Jha, S., Hu, Y., Ramanan, D., Scherer, S., Wang, W.: Ufm: A simple path towards unified dense correspondence with flow. In: NeurIPS (2025) 4, 6, 12, 9
68. Zhao, X., Wu, X., Chen, W., Chen, P.C.Y., Xu, Q., Li, Z.: Aliked: A lighter keypoint and descriptor extraction network via deformable transformation. IEEE Transactions on Instrumentation & Measurement **72** (2023) 1, 3, 5, 12, 9
69. Zhao, X., Wu, X., Miao, J., Chen, W., Chen, P.C., Li, Z.: Alike: Accurate and lightweight keypoint detection and descriptor extraction. IEEE Transactions on Multimedia (2022) 3

LoMa: Local Feature Matching Revisited

Supplementary Material

A Additional Experiments

A.1 Performance using Different Detectors

All the results in the main paper use the DaD [17] detector. We investigate the performance of LoMa using different detectors in Tab. 6. For fair comparison, we retrain the descriptor with an ensemble of detectors and use this for all the subsequent comparisons. To create an ensemble [61], we uniformly sample keypoints from DeDoDe v2 [15], DISK [55], ALIKED [68], and DaD [17] during training. We then train separate matchers (one for each detector) and one ensemble matcher that is jointly trained with keypoints randomly sampled from all detectors. We find DaD to be overall the strongest detector, regardless of setting. Generally, slightly better performance is achieved by specializing the matcher on one detector.

Table 6: Detector ablation. Performance (mAA@10px on the validation set of HardMatch) comparison between training with a single detector and randomly sampling multiple detectors (ensemble).

Method	Ensemble Training	Single Detector
DeDoDe v2 [15]	47.0	47.3
DISK [55]	44.3	45.3
ALIKED [68]	48.9	49.0
DaD [17]	**51.0**	**51.3**

A.2 Dependence on the Number of Keypoints

Throughout the main paper, we evaluate with $N = 4096$ keypoints. We study the performance for fewer keypoints in Tab. 7. The performance difference between 2048 and 4096 keypoints is negligible for other sparse matchers, but LoMa benefits slightly from increasing the number of keypoints beyond 2048. Performance degrades significantly below 2048 keypoints.

A.3 HPatches

We evaluate on HPatches [5] following the LoFTR [50] protocol. The dataset contains planar scenes with homographies. We report the results in Tab. 8. The lightweight LoMa-B^{128} achieves the highest score.

Table 7: Varying max number of keypoints. We report the AUC@20 for different number of maximum keypoints.

Method	MegaDepth-1500				ScanNet-1500			
Max Num. Keypoints →	512	1024	2048	4096	512	1024	2048	4096
SP+SG	70.9	75.4	76.5	76.4	44.5	48.1	49.0	49.1
SP+LG	70.6	74.8	76.4	76.4	43.9	47.6	48.6	48.9
LoMa-B (ours)	**80.3**	**82.7**	**83.1**	**83.6**	**58.3**	**63.3**	**66.4**	**68.2**

Table 8: HPatches. Performance on HPatches [5].

Method	HPatches		
AUC@→	@3px	@5px	@10px
SP+SG	64.4	75.6	86.0
SP+LG	64.2	75.5	85.5
DISK+LG	60.4	72.4	83.5
ALIKED+LG	66.2	76.9	86.3
LoMa-B^{128} (ours)	**67.2**	**78.1**	**87.8**
LoMa-B (ours)	66.5	77.5	87.3
LoMa-L (ours)	66.5	77.8	87.5
LoMa-G (ours)	66.4	77.5	87.3

A.4 SatAst

We evaluate astronaut to satellite matching on SatAst [18]. The dataset features large in-plane rotations and scale changes, making it difficult for most matchers. We report the results in Tab. 4, where we find that, while beating other sparse matchers, LoMa struggles with rotations.

A.5 Oxford Day-and-Night

In the main paper, we report the median performance on the night queries of Oxford Day-and-Night [63]. We use the HLoc [45] pipeline with NetVLAD-50 [2]. The benchmark contains four outdoor scenes (Bodleian Library, H.B. Allen Centre, Keble College, Observatory Quarter) and one indoor scene (Robotics Institute). In Tab. 10, we report the results per scene.

A.6 WxBS

To better understand the relative performance of the matchers, we report the accuracy (PCK) at different pixel thresholds in Fig. 7.

Table 9: SatAst. Astronaut to satellite matching on SatAst [18].

Method	SatAst
AUC@ →	@10px
SP+SG	19.8
SP+LG	12.8
DISK+LG	0.0
ALIKED+LG	12.1
LoMa-B (ours)	18.8
LoMa-L (ours)	21.3
LoMa-G (ours)	**22.9**

Table 10: Oxford Day-and-Night. Full results for night queries. Reporting percentage of correctly localized test images within (0.25m, 2°) / (0.5m, 5°) / (1m, 10°)

	Bodleian Library	H.B. Allen Centre	Keble College	Observatory Quarter	Robotics Institute
SP+SG	21.6/26.5/30.8	44.3/57.5/64.1	10.7/13.6/17.3	48.1/54.4/58.0	71.1/73.6/74.5
SP+LG	20.5/25.3/28.8	43.4/54.1/61.5	10.0/14.2/18.3	47.9/53.5/57.7	70.1/71.8/73.0
DISK+LG	14.8/17.5/20.1	9.6/11.4/14.7	0.5/ 0.8/1.1	16.8/20.4/22.9	53.2/57.5/60.5
ALIKED+LG	22.2/27.3/31.0	42.8/55.5/62.6	10.3/13.5/18.3	45.1/53.9/58.9	57.2/61.4/63.6
LoMa-B_{128}(ours)	**26.1/31.8/36.1**	60.1/71.1/76.6	14.2/17.8/22.0	54.8/62.2/67.1	**73.3/76.1/77.0**
LoMa-B (ours)	24.1/28.9/32.5	57.5/69.7/73.7	15.3/21.0/25.2	54.7/62.4/66.2	70.9/74.1/74.7
LoMa-L (ours)	25.1/30.4/34.3	**62.1**/71.9/76.0	15.4/21.1/26.0	56.0/64.6/69.2	71.5/74.9/76.0
LoMa-G (ours)	25.1/30.4/35.0	61.9/**73.7**/**78.2**	**15.8**/**21.4**/**27.5**	**58.9**/**66.0**/**69.7**	72.2/75.2/76.5

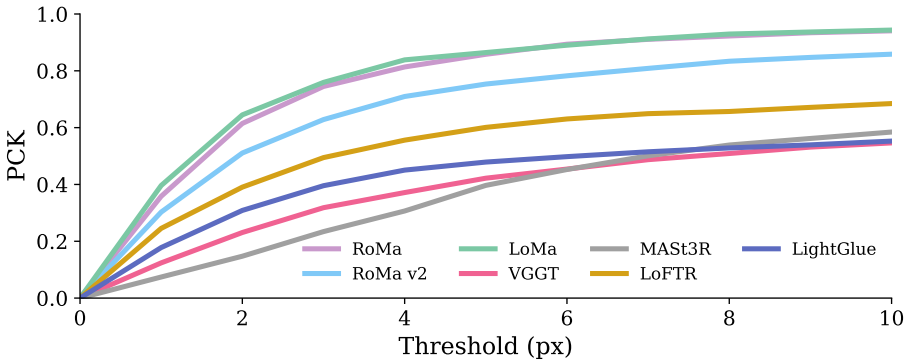
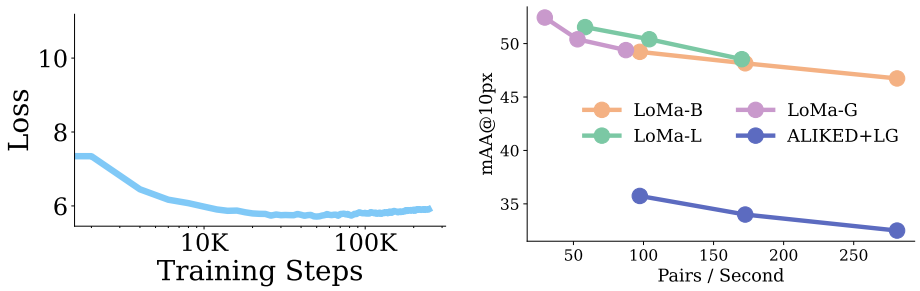


Fig. 7: WxBS accuracy at different thresholds.

A.7 Scaling the Descriptor

As shown in Fig. 8a, the HardMatch validation loss of the descriptor saturates at around 50K steps and then slowly increases. Thus, we limit the descriptor training to only 50K steps (compared to 250K for the matcher).



(a) HardMatch validation loss during descriptor training.

(b) Pareto curve for a single image pair per GPU.

Fig. 8: Descriptor scaling (a) and pareto curve for batch size of 1 (b).

A.8 Inference Speed for a Single Image Pair

In the main paper, we evaluate the speed using a batch size of 16. In many applications, inference will run with a single image pair at a time (batch size of 1). In Fig. 8b, we show the speed for our different model sizes for different stopping layers $L = \{3, 5, 9\}$.

B Details on Evaluation

For all LoMa evaluations, we use an internal resolution of 784×784, $N = 4096$ DaD keypoints, and $L = 9$ layers.

B.1 Relative Pose Estimation

The evaluation protocol follows from LoFTR [50] and is also used in *e.g.* RoMa [19] and RoMa v2 [18]. We use a RANSAC pixel threshold of $\tau = 0.5$. We use the standard AUC metric. The AUC metric evaluates the error of the estimated Essential matrix relative to the ground truth. For each image pair, the error is defined as the maximum of the rotational and translational errors. Since metric scale is unavailable, the translational error is measured using the cosine of the angular difference. The recall at a threshold τ is defined as the fraction of pairs whose error is below τ. The metric AUC@τ° is computed as the normalized integral of the recall curve with respect to the threshold from 0 to τ, divided by τ. In practice, this integral is approximated using the trapezoidal rule over the set of errors produced by the method on the dataset.

B.2 Image Matching Challenge 2022

We run the official Kaggle competition and use 200K iterations of MAGSAC [6, 7] with an inlier threshold of $\tau = 0.2$. We report the mean average accuracy (mAA). The metric evaluates the estimated Fundamental matrix against the hidden ground truth using rotational error (in degrees) and translational error (in meters). A pose is considered correct if both errors fall below specified thresholds. This is evaluated over ten uniformly spaced threshold pairs. The mAA is the average accuracy across all thresholds and images, balanced across scenes.

C HardMatch

C.1 Further Details on Evaluation

Following WxBS [38], our main method for evaluating the hand-labeled correspondences in HardMatch is through the estimation of a Fundamental matrix. Specifically, each method finds matches between the two images and robustly estimates the fundamental matrix using the OpenCV implementation of MAGSAC [6,7] with an inlier threshold of $\tau = 0.25$ pixels. We compute the percent of keypoints (PCK) in the ground truth correspondences consistent with the estimated Fundamental matrix for epipolar pixel thresholds going from 0 to 20. We compute the pixel errors at a resolution of 640×640 and do not evaluate on the approximately 20 pairs we label dynamic.

C.2 Correspondence Evaluation

An alternative methodology involves directly matching the ground truth keypoints. This has the benefit of working even for image pairs where a Fundamental matrix is not well defined, *e.g.* dynamic scenes and non-perspective cameras. For dense matchers,we sample the warp at the ground truth keypoint locations in $I^{\mathcal{A}}$ and find the pixel error to the ground truth correspondences in $I^{\mathcal{B}}$. For sparse matchers, the most straightforward way is to append the ground truth keypoints to the detected keypoints in $I^{\mathcal{A}}$ and record the pixel error between the estimated and true matches in $I^{\mathcal{B}}$. As illustrated in Tab. 11, LoMa performs the best also on this evaluation.

C.3 Dataset statistics

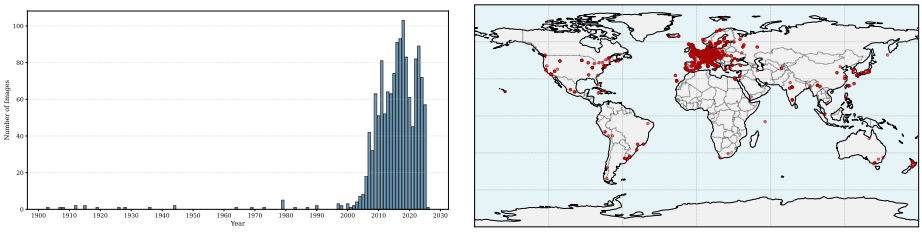
Images in the HardMatch dataset date between the early 20th century until now (*cf*. Fig. 9a) and has a global geographic footprint (*cf*. Fig. 9b). Most of the images are taken at the start of the 21st century in Europe.

C.4 Qualitative Pairs with Matches

In Fig. 10 we illustrate some representative examples for challenging groups in HardMatch. In Fig. 11 we display a random collection of pairs and the matches detected by LoMa-G (inliers during Fundamental matrix estimation with $\tau = 5$ are colored green).

Table 11: Correspondence Evaluation. HardMatch evaluation where, for sparse matchers, ground truth correspondences are appended to detected keypoints.

Method	HardMatch			
PCK@ →	5px	10px	15px	20px
Dense Matchers				
RoMa	52.9	60.9	64.1	66.0
UFM	35.2	47.1	53.4	56.7
RoMa v2	**53.2**	**64.5**	**70.4**	**73.3**
Sparse Matchers, 2048 keypoints				
SP+SG	37.6	39.3	40.2	40.6
SP+LG	39.3	45.1	47.8	49.5
LoMa-B (ours)	64.3	71.4	74.5	76.0
LoMa-L (ours)	65.8	72.7	76.0	77.4
LoMa-G (ours)	**68.0**	**74.0**	**76.8**	**78.2**



(a) Time distribution **(b)** Geographic distribution

Fig. 9: HardMatch statistics. The dataset consists of images taken from all over the world and from over a century apart. The highest concentration is geographically in Europe and temporally in the 21st century.

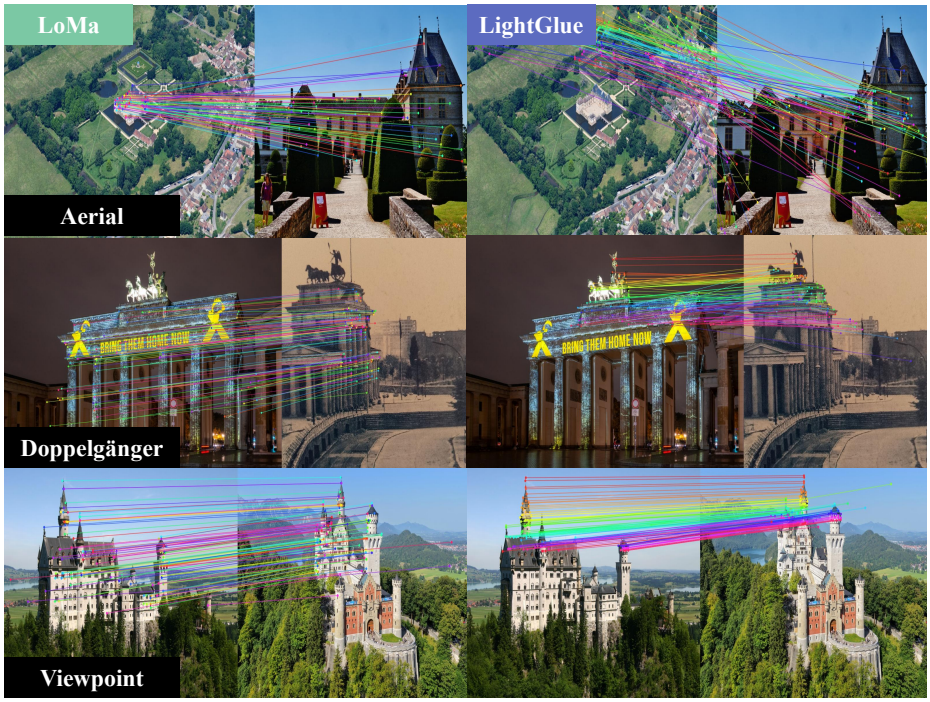


Fig. 10: Hard groups of HardMatch. For hard Doppelgängers in HardMatch, all the matchers fail.

Fig. 11: LoMa-G matches from HardMatch. Inliers at 5px threshold for MAGSAC [6, 7] colored green while outliers are colored red.

C.5 Results by Category and Group

The HardMatch dataset is sourced from 100 Wikimedia Commons categories. We list the categories of the test dataset in Fig. 12. We also report the detailed performance breakdown of different groups in Tab. 14.

Table 14: Detailed HardMatch Performance. Performance (mAA@10px) on different groupings of HardMatch.

Method	Aerial	Celestial	Doppelgänger	Drawing	Illumination	Nature	Seasonal	Temporal	Viewpoint
Feedforward Reconstruction									
MASt3R [29]	**13.0**	**20.2**	**25.4**	**29.1**	**32.3**	**39.5**	**29.7**	**32.3**	**18.2**
VGGT [60]	12.9	8.7	14.8	17.9	32.2	27.7	30.6	32.0	15.6
Dense Matchers									
LoFTR [50]	12.9	25.9	19.3	21.3	33.5	39.5	34.3	38.6	10.5
RoMa [19]	27.2	26.1	28.7	**41.2**	**50.0**	54.2	51.3	**55.0**	20.8
UFM [67]	14.4	**30.7**	22.4	30.4	32.0	33.0	40.3	41.7	15.0
RoMa v2 [18]	**28.7**	28.3	**34.1**	34.3	49.1	**54.4**	**46.5**	50.5	**28.6**
Sparse Matchers, 4096 Keypoints									
SP+SG [12, 46]	16.7	28.1	23.5	27.2	34.5	41.5	42.2	40.8	8.6
SP+LG [12, 32]	12.9	25.1	22.1	26.8	32.8	40.0	37.2	39.2	8.3
DISK+LG [32, 55]	12.0	6.3	16.8	20.0	25.3	30.7	29.2	36.9	9.6
ALIKED+LG [32, 68]	14.0	16.4	21.5	26.0	34.2	40.9	39.1	41.6	10.5
LoMa-B (ours)	31.6	28.1	30.3	41.0	51.7	54.4	54.6	55.6	26.0
LoMa-L (ours)	35.6	**35.8**	35.9	**43.0**	54.5	55.2	56.4	57.6	30.3
LoMa-G (ours)	**36.9**	35.7	**36.3**	40.4	**55.0**	**56.0**	**58.8**	**59.4**	**30.9**

D Progressive Match Refinement

We qualitatively examine the detected matches at different stopping layers in Fig. 13.

E Visualizing a Training Batch

To better understand our training data mix we randomly sample a batch of 32 image pairs and plot them in Fig. 14.

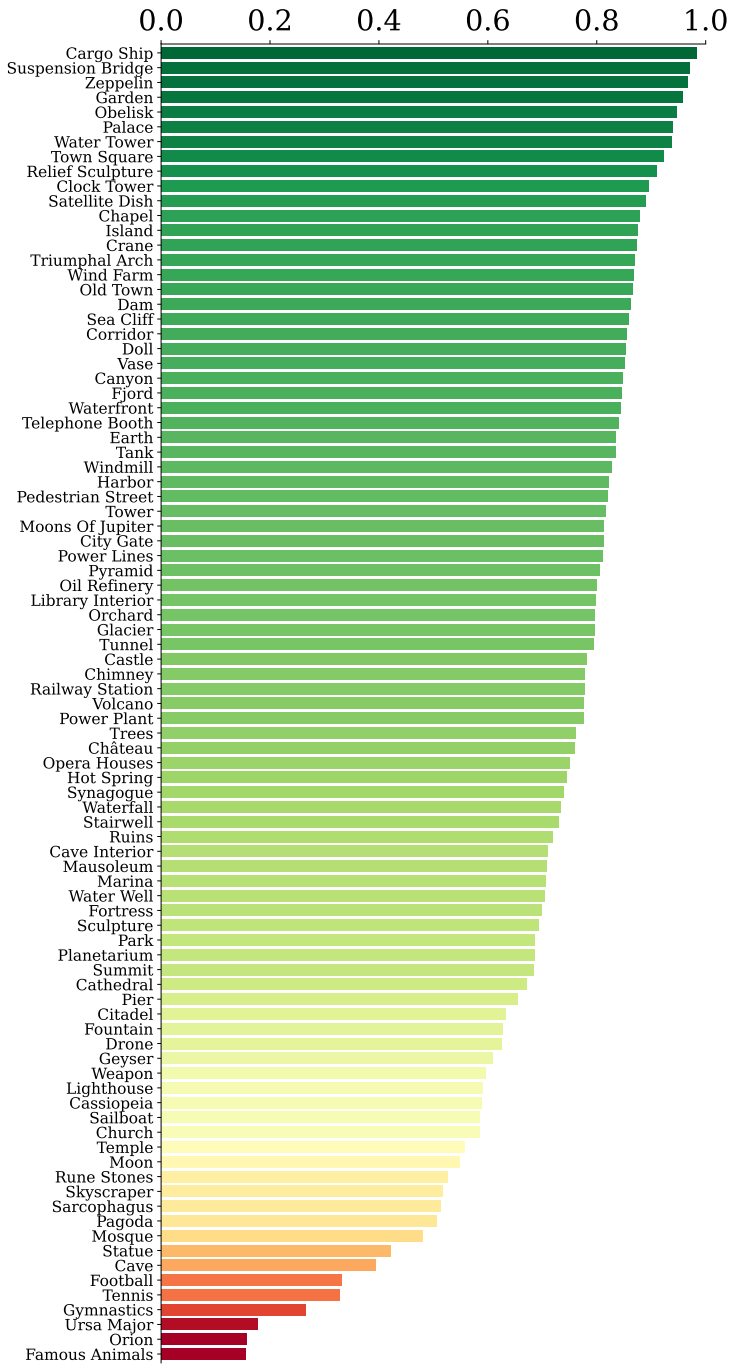
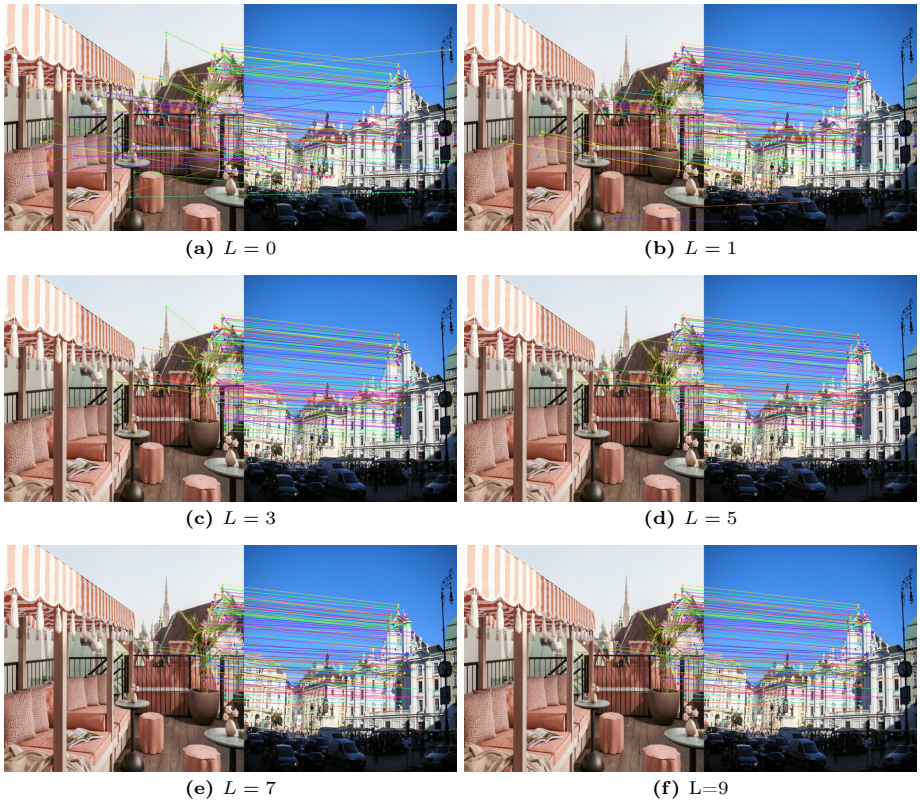


Fig. 12: HardMatch categories from easy to hard. We plot the PCK@10px of LoMa for different categories in the test set.



(a) $L = 0$ **(b)** $L = 1$

(c) $L = 3$ **(d)** $L = 5$

(e) $L = 7$ **(f)** L=9

Fig. 13: Refining matches through depth. The descriptor fails to match the pair ($L = 0$) but as the features are passed through the layers of the matcher, the pair gradually becomes matchable.

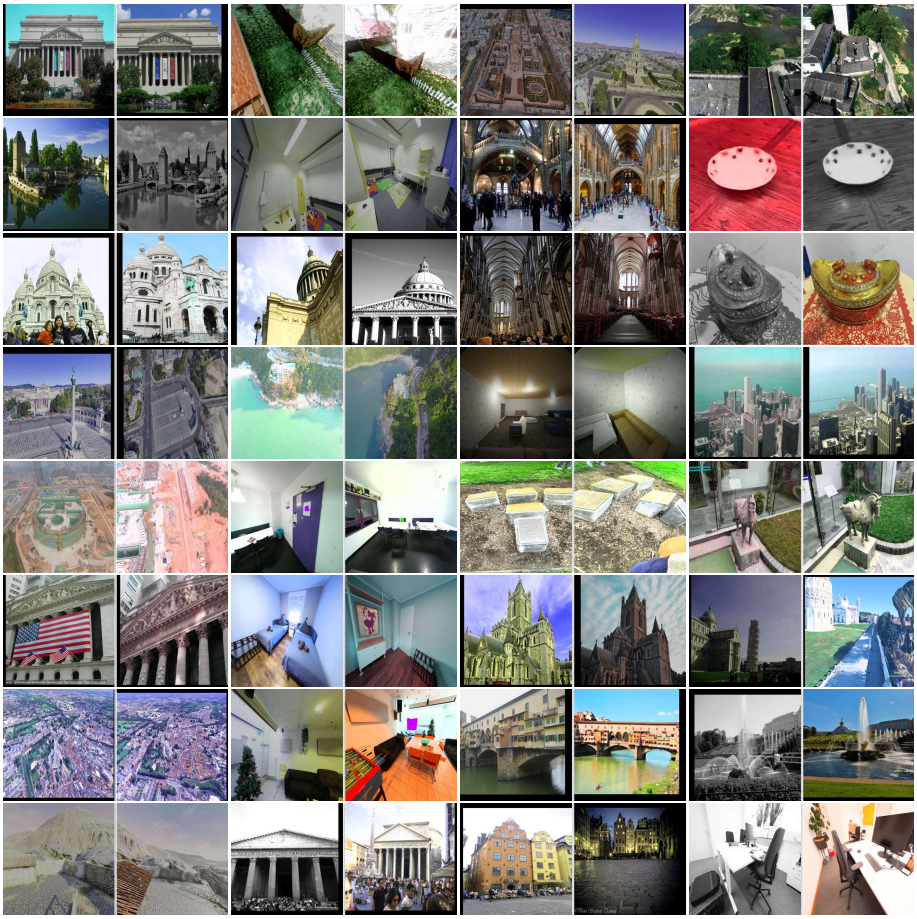


Fig. 14: Visualization of training batch. We visualize a random training batch of 32 image pairs to highlight the diversity in our training data.