[Supplementary Material]
XFeat: Accelerated Features for Lightweight Image Matching

Guilherme Potje[1] Felipe Cadar[1,2] André Araujo[3]
Renato Martins[2,4] Erickson R. Nascimento[1,5]

[1]Universidade Federal de Minas Gerais [2]Université de Bourgogne, ICB UMR 6303 CNRS
[3]Google Research [4]Université de Lorraine, LORIA, Inria [5]Microsoft

{guipotje,cadar,erickson}@dcc.ufmg.br, renato.martins@u-bourgogne.fr, andrearaujo@google.com

In this supplementary material accompanying the main paper, we present a more detailed overview of the architecture of our proposed CNN backbone and the practices employed in the training process. Moreover, we provide an expanded set of qualitative results and extended discussion, providing additional contextualization with the current state-of-the-art methods. Code and weights are available at verlab.dcc.ufmg.br/descriptors/xfeat_cvpr24.

1. Backbone details

To maintain the backbone's structural simplicity, we employ a primary unit termed the basic layer. This unit is structured with a 2D convolution with square kernel sizes $k = 1$ or $k = 3$, complemented by ReLU activation and Batch Normalization. A stride of 2 in the convolution is applied for halving the spatial resolution as needed. The network's architecture is modular, comprising several basic layers as a basic block, as depicted in Fig. 1. Each block consists of two or three basic layers. The backbone of our network comprises six of these basic blocks, designed to halve the spatial resolution in each step while progressively augmenting the depth using the approach detailed in Sec. 3.1 of the main paper. The first basic layer on each block performs the spatial downsampling. Two additional basic blocks, in the end, are employed to perform the fusion of multi-resolution features and reliability map prediction, respectively. Preliminary experiments revealed that adding a single skip connection to the model as shown in Fig. 1 slightly increased performance, which has led to its incorporation in the final backbone design.

2. Training description

We trained the network on a mix of Megadepth [5] scenes using the training split provided by [8] and synthetically warped pairs using raw images (without labels) from COCO [6] in the proportion of $6:4$ respectively. All image pairs were resized to $(W = 800, H = 600)$, and ground-truth correspondences were scaled accordingly. Our ablations show that hybrid training significantly improves generalization for small CNNs, as observed in high-capacity models [7]. The network was trained on batches of 10 image pairs using the Adam optimizer [4] with an initial learning rate of 3×10^{-4}, applying an exponential decay of 0.5 at every $30{,}000$ gradient updates. Convergence is attained after $160{,}000$ iterations, within 36 hours on a single NVIDIA RTX 4090 GPU, consuming 6.5 GB of VRAM in total, considering both training and synthetic warps done on the fly on GPU. Disk I/O is the predominant speed bottleneck due to the overhead of loading images and depth maps from the Megadepth dataset in their original resolution, which

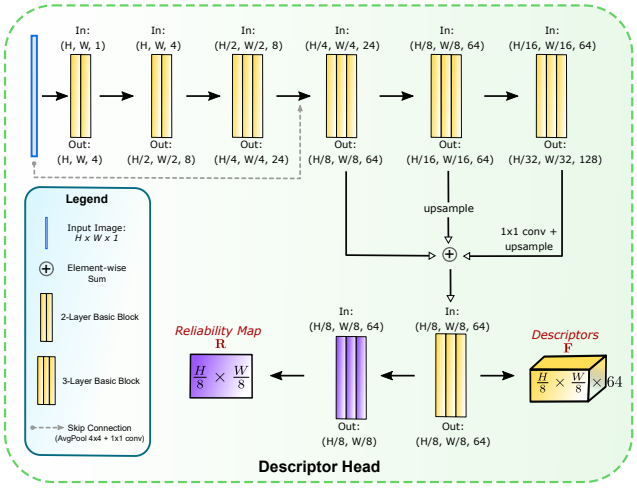

Figure 1. **Detailed descriptor backbone.** Our backbone is comprised of 23 convolutional layers, following the downsampling strategy described in Sec. 3.1 of the main paper. Our network is deeper compared to ALIKE [10] and SuperPoint [2] backbones in terms of layers, but due to the efficient downsampling strategy adopted, our network's inference is much faster.

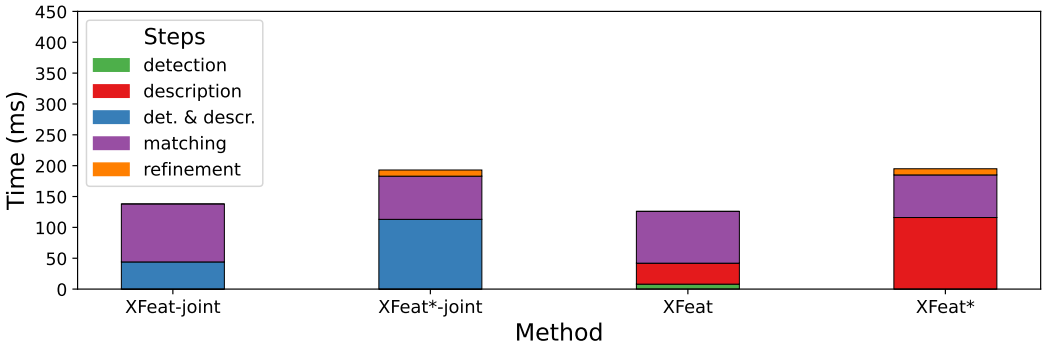

Figure 2. **Detailed timing analysis on i7-6700K CPU.** Required time by each step of our ablated methods.

can be easily solved with a more careful data preparation scheme. The low memory usage of our method enables training on entry-level hardware, facilitating the fine-tuning or full training of our network for specific tasks and scene types.

3. Detailed timing analysis

This section reports a detailed timing analysis of our proposed solutions in sparse and semi-dense matching settings. Regarding XFeat*'s match refinement step, we show in Fig. 2 that the match refinement cost is negligible. More notably, even with the refinement step included, XFeat* achieves a similar matching time compared to XFeat with the same number of keypoints because refinement is performed after the nearest neighbor search. Additionally, we present the extraction running times for the most efficient methods available on an **Orange Pi Zero 3** equipped with a Cortex-A53 ARM processor. This device stands out as one of the **smallest and most affordable consumer-grade embedded computers** ($28). Considering its limited processing power, we adjusted the input resolution to 480×360 for all methods and used their standard PyTorch implementation without any deployment optimization. Our findings show that XFeat operates at an average of 1.8 FPS, SuperPoint at 0.16 FPS, and ALIKE at 0.58 FPS, respectively. This experiment shows that XFeat is the only learned method capable of running over one FPS on a highly constrained embedded device that is not optimized for neural network inference.

4. Megadepth-1500 qualitative results

Fig. 3 shows more qualitative results of our two proposed approaches compared to the baseline methods used in the main paper. For more challenging cases such as strong viewpoint and illumination changes, XFeat and XFeat* exhibit exceptional robustness even compared to DISK [9] – the largest CNN architecture regarding floating point operations. We hypothesize that this robustness is attributed to our network's large receptive field and depth compared to shallower models such as SuperPoint, ALIKE, and SiLK [3], demonstrating the effectiveness of our featherweight backbone in the compute-accuracy trade-off.

5. ScanNet-1500 extended discussion

Recalling the results obtained in Tab. 2 of the main paper, XFeat and XFeat* surpass both fast and standard local feature extractors in pose accuracy while being significantly faster for indoor relative pose estimation. DISK and ALIKE, which were trained in the same Megadepth scenes as XFeat, display signs of overfitting in landmark imagery: they perform exceptionally well in strict thresholds (AUC@5°) on Megadepth-1500 test set, but their relative performance are similar or worse in tasks such as homography estimation and visual localization compared to XFeat and SuperPoint, as one can observe in Tab. 3 and Tab. 4 of the main paper.

We conjecture that XFeat produces less biased local descriptors due to our hybrid training with synthetic warps on COCO. SuperPoint also demonstrate increased generalization accross different downstream tasks and datasets due to its inherent self-supervised training strategy on synthetic warps. Hybrid training can encourage local feature representations to focus less on distinctive textures often present in landmark outdoor imagery that could bias the CNN training. In addition, the large receptive field of our network, as well as its increased layer depth compared to the other approaches, helps XFeat in indoor imagery (which often lacks distinctiveness at the local level), resulting in more consistent matches compared to DISK and ALIKE in ScanNet-1500, even though XFeat and the competitors were not trained on ScanNet data.

6. Comparison with learned matchers

Since XFeat* uses paired inputs when performing the refinement step, we provide additional comparisons of XFeat* (semi-dense matching) with popular learned matchers such

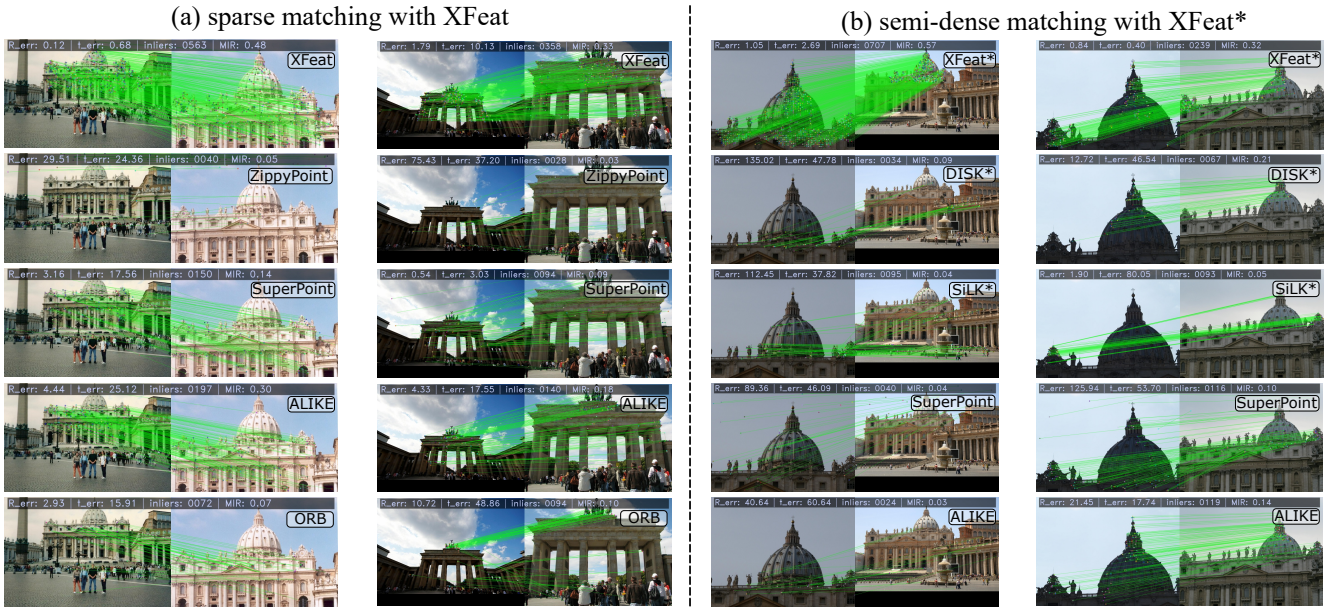

Figure 3. **Additional qualitative results on Megadepth-1500 [5, 8] landmark dataset.** XFeat and XFeat* are robust in demanding scenarios with significant viewpoint and illumination variations, outperforming even the more computationally intensive DISK model in semi-dense matching with 10,000 local features at a striking 16× speedup. In a sparse setting with 4,096 keypoints, our method, which is many times faster than ALIKE (5×) and SuperPoint (9×), demonstrates more robustness to wide baseline transformations due to the effective re-formulation of XFeat's backbone CNN.

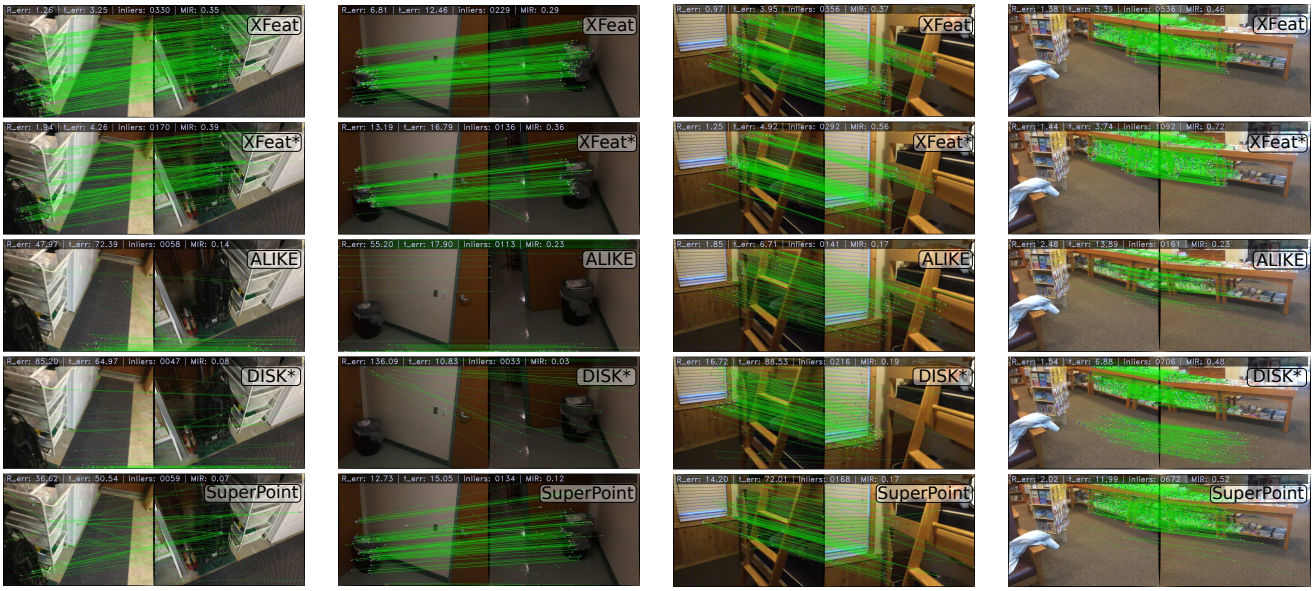

Figure 4. **Additional qualitative results on ScanNet-1500 [1, 8] indoor dataset.** Our proposed approaches consistently outperform state-of-the-art methods such as DISK and ALIKE in indoor imagery, both in terms of camera pose and inlier ratio. Notice that SuperPoint also often outperforms DISK and ALIKE. Sec. 5 provides a detailed discussion on the reasons behind our method's superiority.

as LoFTR [8] and LightGlue [7], and coarse-to-fine strategies as Patch2Pix [11], to elucidate the key differences. The results for these new approaches are shown in Tab. 1. Although XFeat* needs paired inputs for refinement, it fundamentally differs in its methodology from learned matchers, being only comparable to Patch2Pix, as we rely on traditional nearest neighbor search for matching, followed by a lightweight refinement of matches, incurring a negligible

Table 1. **Matchers comparison on Megadepth-1500.** Inference speed in pairs per second (PPS) @ 1,200 px. (i7-6700K CPU).

Method	Type	AUC@5°	@10°	@20°	Acc@10°	MIR	#inliers	PPS
LoFTR	learned matcher	68.3	80.0	88.0	93.9	0.93	3009	0.06
LightGlue	learned matcher	61.4	75.0	84.8	91.8	0.92	475	0.31
Patch2Pix	coarse-fine	47.8	61.0	71.0	77.8	0.59	536	0.05
XFeat*	coarse-fine	**50.2**	**65.4**	**77.1**	**85.1**	**0.74**	**1885**	**1.33**

computational load (see Fig. 2). The requirement for paired inputs does not change the usual pipeline for SfM and visual localization tasks because XFeat*'s features can be stored for each image independently, as usually done for sparse settings. For instance, high-resolution feature maps are not required, unlike LoFTR, to produce refined matches.

Our techniques are, in fact, complementary to learned matchers; for example, LightGlue can be trained using both XFeat and XFeat* features. Learned matchers are more data hungry and much more expensive to train, *e.g.*, LoFTR uses 64 GPUs for 24 hours to be trained. XFeat*, for its turn, can be trained on a single 8 GB GPU. Furthermore, XFeat* offers up to 22× speedup over existing semi-dense solutions as shown in Tab. 1 and surpasses coarse-to-fine approaches such as Patch2Pix in accuracy, while being faster and delivering many more matches than sparse learned matchers as LightGlue. Naturally, XFeat, as a local descriptor, offers limited robustness to aggressive viewpoint changes and highly ambiguous image pairs compared to transformer-based feature matchers. Coupling a lightweight transformer such as LightGlue or LoFTR's linear transformer with XFeat's local features can open new directions in scalable, high-performance image matching tasks, facilitating advancements in both efficiency and accuracy that are pivotal for pushing the boundaries in visual navigation, augmented reality, and real-time visual SLAM.

References

[1] 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. 3

[2] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPRW*, pages 224–236, 2018. 1

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

[4] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 1

[5] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, pages 2041–2050, 2018. 1, 3

[6] 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. Springer, 2014. 1

[7] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local Feature Matching at Light Speed. In *ICCV*, 2023. 1, 3

[8] 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. 1, 3

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

[10] Xiaoming Zhao, Xingming Wu, Jinyu Miao, Weihai Chen, Peter CY Chen, and Zhengguo Li. Alike: Accurate and lightweight keypoint detection and descriptor extraction. *IEEE TMM*, 2022. 1

[11] Qunjie Zhou, Torsten Sattler, and Laura Leal-Taixe. Patch2pix: Epipolar-guided pixel-level correspondences. In *CVPR*, pages 4669–4678, 2021. 3