# [Re] XFeat: Accelerated Features for Lightweight Image Matching

**Anonymous authors**
**Paper under double-blind review**

## Abstract

We present a reproducibility study of XFeat, a lightweight local feature extractor and matcher designed for efficient visual correspondence on resource-constrained hardware. We re-implement the architecture based on the paper and supplementary material, re-evaluate the authors' released checkpoint alongside our re-implementation, and conduct additional architectural ablations to clarify unmotivated design choices. This distinction between re-evaluation and reproduction is crucial, as the paper, supplement, and public code differ in several important details, including the backbone layout, the fusion block, and the training losses. Empirically, our reproduced models closely match, and in some cases slightly outperform, the re-evaluated original checkpoint on Megadepth-1500 and ScanNet-1500, supporting the main claim that XFeat provides a strong accuracy–efficiency trade-off for real-world use. At the same time, our ablations explore two seemingly crucial architectural arguments from the original paper. In particular, the parallel keypoint branch is important for semi-dense matching, but its benefit is less pronounced than the original paper claims, and the motivation for the single skip-connection is less conclusive than originally implied. Finally, our experiments show that downstream computer vision tasks, such as homography estimation, can be reproduced successfully, whereas visual localization on Aachen remains below the paper's reported numbers even when re-evaluating the authors' own checkpoint, suggesting the gap stems from underspecified evaluation details rather than the model itself.

## 1 Introduction

Extracting and matching local features is a fundamental step in high-level computer vision tasks such as structure-from-motion (Schönberger & Frahm, 2016), visual localization (Sattler et al., 2018), SLAM (Mur-Artal et al., 2015), and homography estimation (Hartley & Zisserman, 2004). Recent learned local feature methods have improved matching robustness considerably, but these gains often come at the cost of heavier backbones, larger descriptors, or more complex matching pipelines (Zhao et al., 2022). Efficient alternatives therefore remain highly relevant, especially for robotic and augmented reality systems that must make use of limited computational resources (Mur-Artal & Tardós, 2017).

This makes XFeat particularly interesting. In *XFeat: Accelerated Features for Lightweight Image Matching* (Potje et al., 2024), the authors propose a compact convolutional architecture designed to maintain competitive matching accuracy while being fast on standard hardware, including CPU-only laptops. The method supports both sparse matching through an explicit keypoint head and semi-dense matching through a coarse-to-fine refinement module, enabling a broader range of downstream applications than is typical for lightweight local feature extractors.

The main paper, supplementary material, and released repository together provide substantial information, but they do not form a perfectly consistent specification. The supplement describes a 23-layer backbone with six blocks and a single specific skip connection, which differs slightly from the released `model.py`. The training code also differs from the losses described in the paper. These differences make XFeat interesting to reproduce and highlight the importance of identifying which conclusions remain valid with the exact implementation described in the paper.

Our study makes three contributions. First, we provide a re-implementation guided by the paper and supplementary material, using the official code only when the paper omits implementation-critical details. Second, we re-evaluate the authors' released model on all datasets used in the original report and reproduce the downstream experiments needed to determine whether the paper's efficiency claims apply to practical real-world tasks. Third, we conduct extensive ablations on skip connections and detector coupling. Together, these analyses support the main efficiency narrative of XFeat while offering a more nuanced view of several architectural claims. All code is available on GitHub at `github.com/ML-anonymous-researcher/xfeat-reproduction`.

## 2 Scope of reproducibility

The main finding of the original paper is that a carefully designed lightweight CNN offers a good balance between matching accuracy and computational cost, without relying on hardware-specific optimizations. The paper attributes this result to three interacting design choices: a "featherweight" descriptor backbone, a parallel keypoint detector, and a lightweight match refinement module for semi-dense matching. Our goal is to reproduce the main empirical findings and test the architectural design choices that were not ablated in the original paper. In particular, we evaluate the following claims:

- **Claim 1:** XFeat performs well in real time on GPU-free settings and resource-constrained devices without hardware-specific optimizations.
- **Claim 2:** A parallel keypoint detector is necessary for robust semi-dense matching.
- **Claim 3:** A single skip connection is sufficient for improved matching performance and additional skip connections do not provide meaningful gains.
- **Claim 4:** XFeat achieves competitive results in downstream vision tasks, including homography estimation and visual localization.

Our approach includes both reproduction and diagnosis. Reproducing the main results tables alone is insufficient to determine whether all the authors' claims are supported by empirical evidence. Additionally, we cannot be certain whether the differences observed in our experiments are due to different implementations or incorrect assumptions. Therefore, whenever possible, we compare three settings: the numbers reported in the original paper, our re-evaluation of the authors' released checkpoint, and our independently trained reproduction. This setup enabled us to better interpret the downstream results from our experiments.

## 3 Methodology

We base our reproduction on three sources: the main XFeat paper, the supplementary material, and the official GitHub repository. The paper describes the method conceptually, the supplement provides backbone and training details missing from the main text, and the repository contains the released implementation, checkpoints, and benchmark scripts for the Megadepth-1500 and ScanNet-1500 benchmarks. We follow the paper and supplement whenever they clearly describe a component; when details are omitted, we defer to the official code. This approach allows us to remain faithful to the paper while making as few arbitrary implementation choices as possible.

The training data used by the authors consists of Megadepth (Li & Snavely, 2018) and a 20K subset of COCO 2017 (Lin et al., 2014), mixed in a 6:4 ratio and resized to approximately $800 \times 600$ during training. We use the same data and train/test split described in the text. For evaluation, we follow the original paper by using Megadepth-1500 and ScanNet-1500 for relative pose estimation. We also evaluate homography estimation on HPatches (Balntas et al., 2017) and visual localization on Aachen Day-Night 1.0 (Sattler et al., 2018), as these are key to downstream performance claims made in the paper. Megadepth is a large Internet photo dataset with camera geometry and depth derived from SfM/MVS; ScanNet is an RGB-D indoor reconstruction dataset; HPatches is the standard planar homography benchmark; and Aachen is a canonical benchmark for visual localization under severe lighting change.

The official repository currently provides documentation for end-to-end training and benchmark scripts for Megadepth-1500 and ScanNet-1500, and explicitly notes that small AUC deviations can result from RANSAC (Fischler & Bolles, 1981). However, it does not include complete scripts for all downstream experiments we reproduce here, particularly HPatches and Aachen. We implemented our own timing analysis, homography estimation, and visual localization scripts, following the evaluation instructions of the corresponding benchmarks as closely as possible. For Megadepth and ScanNet, we use the following settings: sparse XFeat extracts up to 4,096 keypoints scored by keypoint confidence multiplied by reliability, while semi-dense XFeat* processes two image scales, retains the top 10,000 features by reliability, and filters matches through refinement. Uncertainties were quantified using analytically computed standard errors over the evaluation set. For binary accuracy and AUC metrics, we used the binomial standard error $\sqrt{p(1-p)/n}$, where $p$ is the observed proportion of samples satisfying the success criterion (correct classification or error below the given threshold), and $n$ is the number of samples. For continuous metrics, such as the mean inlier ratio and mean number of inliers, we report the sample standard error $s/\sqrt{n}$, where $s$ is the sample standard deviation.

### 3.1 Model and ablation design

The reproduced XFeat model follows the architecture described in the paper and supplementary material rather than the exact structure of the released `model.py`. Each *basic layer* is a convolution followed by BatchNorm and ReLU. The supplement describes a backbone with six convolutional blocks and 23 convolutional layers in total, with channel counts $\{4, 8, 24, 64, 64, 128\}$, progressively deepening as spatial resolution decreases. A single skip connection injects a downsampled projection of the input image into the features before the 1/4 resolution block.

The descriptor head constructs a dense 64-dimensional feature map $F \in \mathbb{R}^{H/8 \times W/8 \times 64}$ by merging features from three scales $\{1/8, 1/16, 1/32\}$ after upsampling and projection to a common dimension. The resulting sum is passed through a fusion block, followed by an additional head that predicts a reliability map $R \in \mathbb{R}^{H/8 \times W/8}$. The keypoint head is separate from the descriptor head. Instead of predicting keypoints from the final encoder features, XFeat unfolds the input image into $8 \times 8$ cells and processes this tensor with lightweight $1 \times 1$ convolutions to predict one of 64 possible cell positions plus a dustbin class (65 channels in total). A more complete diagram of the originally proposed model than that shown in the paper is presented in Figure 1. Finally, the fine matcher is a lightweight MLP that takes concatenated descriptor pairs and predicts cell offsets for coarse-to-fine correspondence.

Several implementation differences required us to make some explicit reimplementation decisions. First, the released encoder omits one of the early convolutions present in the supplement's six-block description. Second, the paper states that the fusion stage uses three basic layers, while the released implementation uses two basic layers followed by a $1 \times 1$ convolution. Third, the paper's keypoint-head description does not mention the final $1 \times 1$ classifier that produces 65 logits per cell. We follow the supplement for the backbone depth, but follow the code for the fusion-block output and the final MLP. We find this implementation to be the most faithful interpretation of the authors' intent.

Our ablation study extends the one presented in the paper. In addition to the default model, we evaluate four variants implemented from the reproduced architecture: `skip_none`, which removes the original skip connection; `skip_input`, which routes input projections to every block; `skip_resnet`, which adds residual connections around each block inspired by ResNet-style skip connections (He et al., 2016); and `non_parallel_keypoints`, which replaces the parallel keypoint branch with a detector block operating on encoder features, similar to other heads. The last ablation is our re-implementation of the paper's *joint keypoint extraction* experiment. The remaining ablations in the paper test design choices already well-studied in efficiency literature, so we focused our ablation budget on the two claims most specific to XFeat's architecture.

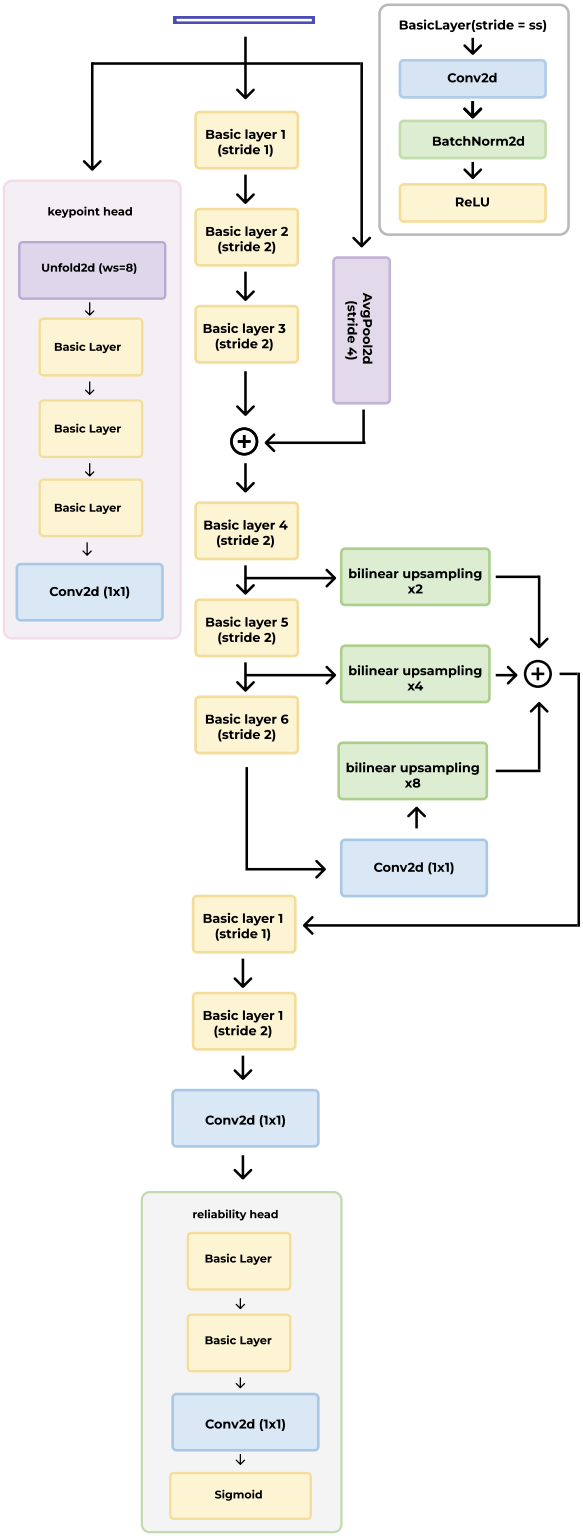


Figure 1: Reproduced XFeat architecture. A six-block convolutional backbone with a skip connection feeds a multi-scale descriptor head producing a feature and reliability map. A separate keypoint head predicts locations within $8 \times 8$ cells.

### 3.2 Training objective and computational requirements

The original paper defines four losses: a dual-softmax descriptor loss ($\mathcal{L}_{ds}$), a reliability loss ($\mathcal{L}_{rel}$), a fine matching loss ($\mathcal{L}_{fine}$), and a keypoint classification loss ($\mathcal{L}_{kp}$), combined linearly as

$$\mathcal{L} = \alpha\mathcal{L}_{ds} + \beta\mathcal{L}_{rel} + \gamma\mathcal{L}_{fine} + \delta\mathcal{L}_{kp}.$$

The descriptor loss is the standard symmetric dual-softmax formulation over matched descriptor pairs; the reliability map is supervised from the confidence derived from dual-softmax matching; the fine matcher is described in the paper as a cell classification problem over an $8 \times 8$ offset grid; and the keypoint head is trained by distilling keypoints from ALIKE (Zhao et al., 2022).

Notably, the released training code differs from the written formula. The repository retains the dual-softmax descriptor objective but trains fine matching using a confidence-weighted coordinate classification loss rather than the exact loss described in the paper. Additionally, the training loop adds an L1 loss that regresses the reliability heatmap toward the dual-softmax confidence and separately applies ALIKE-based distillation to the 65-way keypoint logits. In other words, the released code distributes the "reliability" and "keypoint" supervision described in the paper across several loss terms, with equal weighting through simple averaging in the training loop. Because these deviations affect reproduction, our implementation follows the paper more closely where possible, while using the released code for omitted details. We deliberately do not tune loss weights as we did not wish to perform hyperparameter optimization.

The supplement reports training with batches of 10 image pairs using Adam, an initial learning rate of $3 \times 10^{-4}$, and exponential decay by a factor of 0.5 every 30,000 iterations, with convergence after 160,000 iterations in about 36 hours on a single RTX 4090 using approximately 6.5 GB of VRAM. Our experiments were conducted on a Slurm cluster with NVIDIA H100 80GB HBM3 and NVIDIA L4 24GB PCIe GPUs. Training our reproduced models converged within 17 hours on a single H100. Evaluation on 1,500 pairs took at most a few minutes on an L4 GPU, and the CPU-only timing experiments were performed on a MacBook Pro with Apple M1 Pro and a Windows laptop with an AMD Ryzen 7 5800H. The training code required no modification to use the more powerful hardware, except for some initial tests with batch size and convergence speed.

## 4 Results

### 4.1 Main reproduction results

We begin with the main speed–accuracy experiment of the original paper. The purpose of this experiment is to rank methods by AUC or FPS and to recover the Pareto frontier that motivates XFeat: sparse XFeat is the fastest learned method among competitive baselines, while semi-dense XFeat* achieves a more accurate operating point without significantly reducing inference speed.

Table 1: Speed and relative pose estimation accuracy on Megadepth-1500. The best results are in bold, and the second best are underlined.

| Method | AUC@10° | FPS |
|---|---|---|
| XFeat | 76.9 ± 1.09 | <u>11.8</u> ± 1.07 |
| XFeat* | **86.9** ± 0.87 | 6.3 ± 0.24 |
| ALIKE | 78.1 ± 1.07 | 2.8 ± 0.15 |
| DISK | <u>80.9</u> ± 1.01 | 1.2 ± 0.04 |
| ORB | 51.4 ± 1.29 | **16.9** ± 5.64 |
| SuperPoint | 71.1 ± 1.17 | 1.1 ± 0.02 |

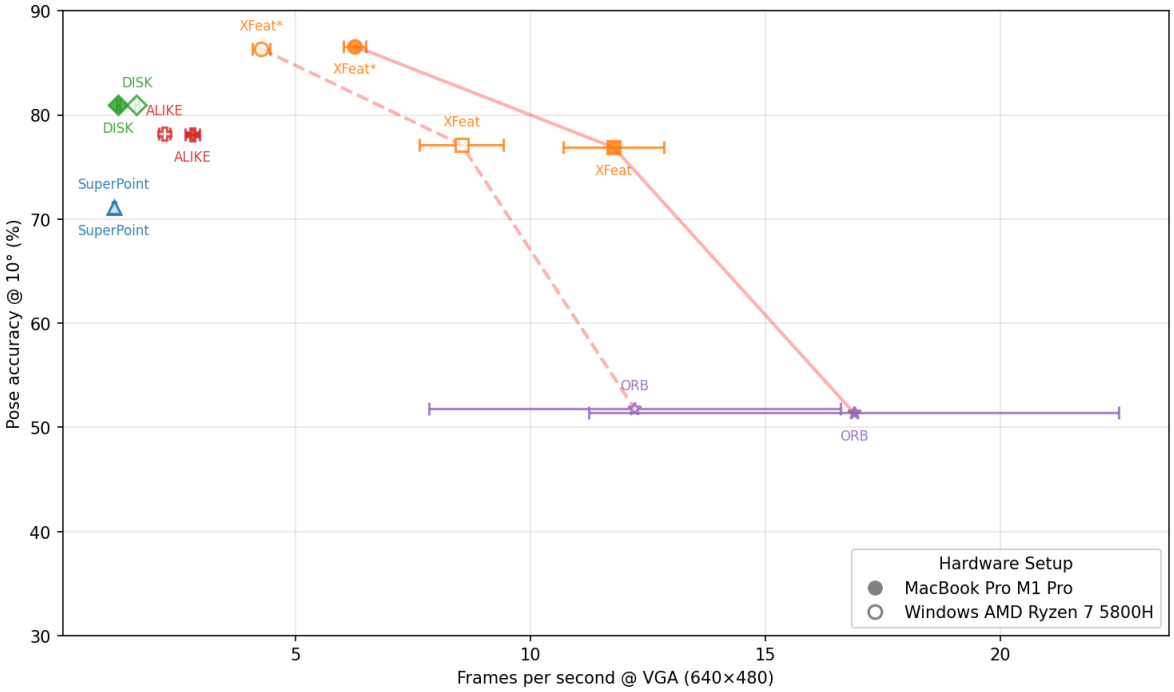


Figure 2: Speed–accuracy trade-off for relative pose estimation on Megadepth-1500 under CPU inference. Each point reports AUC@10° and frames per second on our two laptop platforms. XFeat remains the fastest learned method among the evaluated baselines, while XFeat* achieves higher accuracy at lower, but still practical, throughput.

Figure 2 (compare with Figure 1 in the original paper) and Table 1 reproduce the central trade-off claimed by the original paper. In our CPU benchmark, XFeat is the fastest learned method by a wide margin and remains competitive in pose accuracy, while XFeat* substantially improves AUC@10° at about half the inference speed. ORB remains the fastest overall, but with a significant accuracy cost and large deviations. The absolute FPS values differ from those in the original paper because measurements were taken on different CPUs; what matters is that the relative ordering and the Pareto shape are preserved.

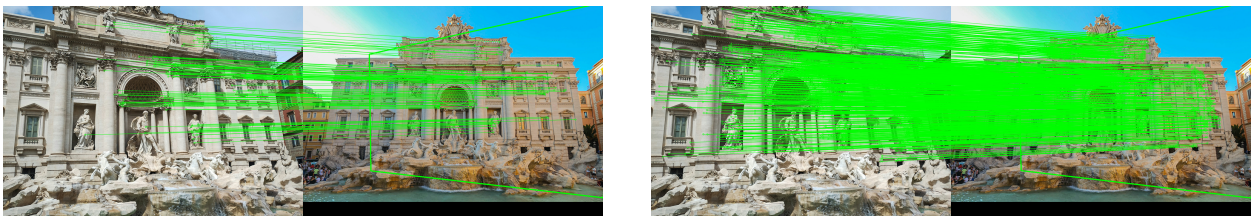

Figure 3: Qualitative comparison of correspondences produced by XFeat (left) and XFeat* (right) on the same image pair. The sparse model returns a set of high-confidence matches, while the semi-dense variant produces denser correspondences with broader support.

Figure 3 (compare with Figure 2 in the original paper) shows that the reproduced model performs well in both operating modes on a real-world image. Sparse XFeat is geometrically consistent, while XFeat* covers more of the scene and quickly recovers correspondences. This is important because the original paper's argument for the refinement module is primarily related to efficiency: the semi-dense mode should help recover more constraints for pose estimation while maintaining as much accuracy as possible. This is exactly what we observe in Figure 2. The qualitative behavior also aligns with the paper's analysis of sparse versus semi-dense modes.

Table 2: Megadepth-1500 relative camera pose estimation. We compare the numbers reported in the original paper, our re-evaluation of the authors' released checkpoint, and our independently trained reproduction.

| Method | AUC@5° | AUC@10° | AUC@20° | Acc@10° | MIR | #inliers | FPS |
|---|---|---|---|---|---|---|---|
| XFeat Original | 42.6 | 56.4 | 67.7 | 74.9 | 0.55 | 892 | **27.1** ± 0.33 |
| XFeat Re-evaluated | 43.8 ± 1.23 | 57.6 ± 1.10 | 68.6 ± 0.99 | 75.9 ± 1.10 | 0.56 | 916 ± 14 | 12.9 ± 1.61 |
| XFeat Reproduced | 43.4 ± 1.23 | 57.5 ± 1.10 | 68.9 ± 0.97 | 75.9 ± 1.10 | 0.65 | 1090 ± 16 | 11.8 ± 1.07 |
| XFeat* Original | 50.2 | 65.4 | 77.1 | 85.1 | 0.74 | 1885 | 19.2 ± 1.12 |
| XFeat* Re-evaluated | **50.8** ± 1.13 | 66.1 ± 0.90 | 78.0 ± 0.71 | 86.2 ± 0.90 | 0.75 | 2131 ± 36 | 7.2 ± 0.38 |
| XFeat* Reproduced | 50.2 ± 1.11 | **66.2** ± 0.88 | **78.4** ± 0.65 | **86.5** ± 0.88 | **0.76** | **2159** ± 36 | 6.3 ± 0.24 |

Table 3: ScanNet-1500 relative camera pose estimation. As in Table 2, we report the original paper, our re-evaluation of the released checkpoint, and our reproduction.

| Method | AUC@5° | AUC@10° | AUC@20° |
|---|---|---|---|
| XFeat Original | 16.7 | 32.6 | 47.8 |
| XFeat Re-evaluated | 15.5 ± 1.23 | 30.4 ± 1.29 | 45.6 ± 1.22 |
| XFeat Reproduced | 15.5 ± 1.23 | 30.7 ± 1.29 | 45.6 ± 1.22 |
| XFeat* Original | 18.4 | 34.7 | 50.3 |
| XFeat* Re-evaluated | 18.5 ± 1.26 | 34.6 ± 1.27 | 50.0 ± 1.18 |
| XFeat* Reproduced | **19.5** ± 1.28 | **36.7** ± 1.26 | **52.8** ± 1.13 |

Next, we compare the original paper's tables to our re-evaluation of the XFeat model and to our independently trained reproduction. While we expect close agreement with the reported results, minor deviations may arise due to training randomness and hardware differences.

Table 2 (compare with Table 1 in the original paper) shows strong reproduction results on Megadepth-1500. Our reproduced XFeat slightly outperforms the re-evaluated original checkpoint on some accuracy metrics, and our reproduced XFeat* is nearly identical to the re-evaluated checkpoint while remaining slightly stronger on AUC and Acc@10°. Model behavior is stable, and the improvements over the released checkpoint are modest, likely due to random chance. The largest disagreement with the paper is in inference speed rather than accuracy. Since the re-evaluated checkpoint is much slower than the FPS originally reported in the paper, this speed discrepancy can be mainly attributed to hardware differences.

The ScanNet-1500 results in Table 3 (compare with Table 2 in the original paper) show a similar pattern, though with a more pronounced difference between sparse and semi-dense matching. Sparse XFeat is nearly indistinguishable from the re-evaluated original checkpoint, indicating that minor implementation differences do not significantly affect performance. In contrast, semi-dense XFeat* demonstrates a more noticeable improvement over both the re-evaluated checkpoint and the results reported in the paper. This is notable because ScanNet differs from the landmark-focused images of Megadepth, so stronger performance on ScanNet may suggest less overall model overfitting. The conclusion from Tables 2 and 3 is that the main empirical claims in the original paper are reproducible, with only minor evaluation variance introduced by training initialization and hardware differences.

### 4.2 Architectural ablations

The original paper does not provide code for its ablation study, so we fully re-implemented the architectural ablations ourselves. Specifically, the paper's "joint keypoint extraction" experiment states that a convolutional block was used on top of the encoder features to extract keypoints, but provides no further details. We found no justification for making this head less expressive than the reliability head, so we replicated the reliability head's design and used it instead. This non-parallel keypoint variant predicts keypoints from the encoder feature map rather than from the dedicated unfolded-input branch. This approach makes the comparison fairer and more clearly reveals the effects of coupling detection and description.

Table 4: Megadepth-1500 ablation study for relative camera pose estimation. Higher values indicate better performance for AUC, Acc@10°, MIR, and the number of inliers. The variants compare the default architecture with a coupled keypoint head, no skip connection, input-routed skips, and ResNet-style residual skips.

| Method | AUC@5° | AUC@10° | AUC@20° | Acc@10° | MIR | #inliers |
|---|---|---|---|---|---|---|
| XFeat Default | 43.4 ± 1.23 | 57.5 ± 1.10 | 68.9 ± 0.97 | 75.9 ± 1.10 | 0.65 | 1090 ± 16 |
| XFeat Non-parallel Kpts | 43.6 ± 1.24 | 57.3 ± 1.10 | 68.8 ± 0.97 | 75.9 ± 1.10 | 0.64 | 1105 ± 17 |
| XFeat Skip Input | 43.8 ± 1.23 | 57.5 ± 1.10 | 69.1 ± 0.96 | 76.1 ± 1.10 | 0.65 | 1094 ± 17 |
| XFeat Skip None | 42.4 ± 1.24 | 56.3 ± 1.12 | 68.1 ± 0.96 | 74.9 ± 1.12 | 0.64 | 1057 ± 16 |
| XFeat Skip ResNet | 43.5 ± 1.23 | 57.6 ± 1.08 | 69.4 ± 0.95 | 77.3 ± 1.08 | 0.64 | 1073 ± 16 |
| XFeat* Default | 50.2 ± 1.11 | 66.2 ± 0.88 | 78.4 ± 0.65 | 86.5 ± 0.88 | <u>0.76</u> | **2159** ± 36 |
| XFeat* Non-parallel Kpts | 46.8 ± 1.18 | 62.6 ± 0.94 | 75.6 ± 0.73 | 84.2 ± 0.94 | 0.73 | 1974 ± 33 |
| XFeat* Skip Input | 51.0 ± 1.12 | 66.3 ± 0.88 | 78.6 ± 0.67 | 86.7 ± 0.88 | **0.79** | <u>2131</u> ± 36 |
| XFeat* Skip None | **53.0** ± 1.08 | **68.2** ± 0.84 | **79.7** ± 0.65 | <u>87.9</u> ± 0.84 | <u>0.76</u> | 2116 ± 35 |
| XFeat* Skip ResNet | <u>51.7</u> ± 1.10 | 67.5 ± 0.84 | <u>79.2</u> ± 0.67 | **88.0** ± 0.84 | **0.79** | 2089 ± 36 |

Table 5: ScanNet-1500 ablation study for relative camera pose estimation. The same architectural variants as in Table 4 are evaluated on indoor data.

| Method | AUC@5° | AUC@10° | AUC@20° |
|---|---|---|---|
| XFeat Default | 16.1 ± 1.23 | 31.5 ± 1.28 | 47.3 ± 1.20 |
| XFeat Non-parallel Kpts | 16.0 ± 1.24 | 31.6 ± 1.28 | 46.8 ± 1.21 |
| XFeat Skip Input | 15.4 ± 1.23 | 30.4 ± 1.29 | 45.9 ± 1.21 |
| XFeat Skip None | 15.1 ± 1.22 | 30.3 ± 1.29 | 45.7 ± 1.22 |
| XFeat Skip ResNet | 16.3 ± 1.24 | 31.0 ± 1.29 | 45.6 ± 1.23 |
| XFeat* Default | **20.4** ± 1.28 | **37.6** ± 1.25 | **53.8** ± 1.12 |
| XFeat* Non-parallel Kpts | 19.6 ± 1.27 | 36.3 ± 1.26 | 52.1 ± 1.15 |
| XFeat* Skip Input | 19.8 ± 1.27 | <u>36.7</u> ± 1.25 | 52.8 ± 1.14 |
| XFeat* Skip None | 19.5 ± 1.27 | 36.6 ± 1.26 | 52.7 ± 1.13 |
| XFeat* Skip ResNet | <u>19.9</u> ± 1.27 | <u>36.7</u> ± 1.25 | <u>53.0</u> ± 1.13 |

The ablations yield two clear conclusions and one remaining open question. The first clear conclusion concerns the parallel keypoint branch. On sparse XFeat, replacing the dedicated branch with a non-parallel detector has almost no effect on Megadepth and only a negligible effect on ScanNet. On XFeat*, however, the same change causes a large drop on Megadepth. For example, AUC@10° decreases from 66.2 to 62.6, and AUC@20° from 78.4 to 75.6. The drop on ScanNet is smaller but still consistent. This supports the paper's architectural intuition for the semi-dense matching: when descriptor features are also used in refinement, coupling them too closely to detector features appears harmful.

The second conclusion relates to the presence of at least one skip connection. On sparse XFeat, removing the original skip connection reduces performance on both Megadepth and ScanNet. The decrease is modest but consistent across metrics, somewhat supporting the supplement's claim that the original skip connection was beneficial. However, the magnitude of the improvement is small enough that it cannot serve as decisive evidence that skip connections significantly enhance performance. The differences among the stronger skip connection experiments are often comparable to the reported uncertainty. This suggests that skip connections do not play a major role in XFeat's performance, likely due to the network's shallow architecture.

The remaining open question is whether the original paper's specific single-skip-connection design is optimal. Our evidence does not appear to support this claim. On Megadepth, `skip_input` and ResNet-style skips perform at least as well as the default sparse model, and for XFeat*, both `skip_none` and `skip_resnet` outperform the default on several metrics. ScanNet challenges this conclusion, as the default XFeat* is the strongest indoor model overall, yet even there, the performance differences between skip-connection designs are small. In our view, skip-connections matter, but the ablation study does not justify the stronger claim that one carefully placed skip-connection is uniquely preferable to other alternatives.

Table 6: Homography estimation on HPatches. Illumination and viewpoint splits are reported separately. We use the same numbers as in our experiments and report reproduced results with the best-performing MAGSAC++ threshold of 1.5.

| | Illumination | | | Viewpoint | | |
|---|---|---|---|---|---|---|
| **Method** | **MHA@3** | **MHA@5** | **MHA@7** | **MHA@3** | **MHA@5** | **MHA@7** |
| SiLK | 78.5 | 82.3 | 83.8 | 48.6 | 59.6 | 62.5 |
| SuperPoint | 94.6 | 98.5 | 98.8 | **71.1** | 79.6 | 83.9 |
| DISK | 94.6 | **98.8** | **99.6** | 66.4 | 77.5 | 81.8 |
| ORB | 74.6 | 84.6 | 85.4 | 63.2 | 71.4 | 78.6 |
| ZippyPoint | 94.2 | 96.9 | 98.5 | 66.1 | 76.8 | 80.7 |
| ALIKE | 94.6 | 98.5 | **99.6** | 68.2 | 77.5 | 81.4 |
| XFeat Original | **95.0** | 98.1 | 98.8 | 68.6 | 81.1 | 86.1 |
| XFeat Re-evaluated | 93.7 ± 1.44 | 97.5 ± 0.92 | 98.3 ± 0.78 | 68.8 ± 2.70 | **82.4** ± 2.22 | 85.8 ± 2.04 |
| XFeat Reproduced | 93.3 ± 1.48 | 97.9 ± 0.85 | 98.6 ± 0.70 | 66.4 ± 2.75 | 79.0 ± 2.38 | 84.1 ± 2.13 |
| XFeat* Re-evaluated | 92.6 ± 1.55 | 96.8 ± 1.04 | 98.6 ± 0.70 | 51.5 ± 2.91 | 74.9 ± 2.53 | 84.4 ± 2.12 |
| XFeat* Reproduced | 93.7 ± 1.44 | 97.5 ± 0.92 | 97.9 ± 0.85 | 65.8 ± 2.77 | 80.3 ± 2.32 | **88.8** ± 1.84 |

Table 7: Visual localization evaluation on the original Aachen dataset (Day vs. Night). We report the exact numbers from the original paper, our re-evaluation of the released checkpoint, and our reproduction.

| | Day | | | Night | | |
|---|---|---|---|---|---|---|
| **Method** | **(0.25m, 2°)** | **(0.5m, 5°)** | **(5m, 10°)** | **(0.25m, 2°)** | **(0.5m, 5°)** | **(5m, 10°)** |
| SuperPoint | **87.4** | 93.2 | 97.0 | 77.6 | 85.7 | 95.9 |
| DISK | 86.9 | **95.1** | **97.8** | **83.7** | **89.8** | **99.0** |
| ORB | 66.9 | 76.1 | 81.7 | 10.2 | 12.2 | 19.4 |
| ZippyPoint | 80.7 | 88.6 | 93.7 | 61.2 | 70.4 | 79.6 |
| ALIKE | 85.7 | 92.4 | 96.7 | 81.6 | 88.8 | **99.0** |
| XFeat Original | 84.7 | 91.5 | 96.5 | 77.6 | **89.8** | 98.0 |
| XFeat Re-evaluated | 76.3 | 83.6 | 89.8 | 65.3 | 79.6 | 87.8 |
| XFeat Reproduced | 79.0 | 86.4 | 90.9 | 66.3 | 80.6 | 89.8 |

### 4.3 Downstream tasks

The original paper highlights the practical applicability of XFeat with two higher-level downstream tasks: homography estimation and visual localization. These tasks are important because they test whether the local features generated by XFeat remain useful for real-world problems, not just in visual matching scenarios.

For homography estimation on the HPatches dataset, the authors state that they use MAGSAC++ (Barath et al., 2020), but they do not explicitly mention which threshold they used. Through our basic experiments, we determined that a threshold of 1.5 performed best overall. We use this threshold for all subsequent experiments on HPatches.

The HPatches results in Table 6 (compare with Table 3 in the original paper) broadly confirm the original paper's claim that XFeat is competitive for homography estimation. Our reproduced sparse model is slightly weaker than the re-evaluated original checkpoint on the viewpoint split, but it remains within the same overall performance level and retains good illumination robustness. The most notable extension is XFeat*: although the original paper emphasizes XFeat rather than XFeat* for this task, our reproduced semi-dense model achieves the highest MHA@7 on the viewpoint split. This suggests that the denser correspondence set can be especially useful when the evaluation threshold is less strict, which is also consistent with the paper's observation that XFeat tends to perform better under looser geometric thresholds.

Aachen is the one downstream task where our reproduction clearly does not match the original table. Although we match the processing resolution of 1024 pixels and extract the top 4096 keypoints, Table 7 (compare with Table 4 in the original paper) shows that both our re-evaluated original checkpoint and our reproduced model underperform the paper's reported numbers by a noticeable margin. We note that this gap is already present when we evaluate the authors' own checkpoint. The underperformance suggests that localization evaluation with the HLoc (Sarlin et al., 2019) library is highly sensitive to implementation details not specified in the original paper. Since our reproduced model consistently improves over the re-evaluated checkpoint, but both remain below the paper's results, the most likely source of difference is the end-to-end localization setup rather than the model itself. This makes Claim 4 only partially reproducible: the qualitative conclusion that XFeat is useful for localization remains valid, but the exact published Aachen numbers are not trivially independently reproducible.

## 5 Discussion

### 5.1 Assessment of the original claims

**Claim 1: real-time performance on resource-constrained hardware.** We can confidently state that this claim is valid. In our CPU benchmark, sparse XFeat is much faster than the learned baselines while maintaining competitive relative-pose accuracy, and XFeat* remains practically usable while providing a significant accuracy gain. The absolute FPS differs from the original figure, but the original repository notes that RANSAC stochasticity and refactoring can cause slight differences in evaluation. Our re-evaluation confirms that runtime is much more platform-dependent than accuracy. In conclusion, XFeat genuinely occupies a favorable efficiency niche among learned local feature methods.

**Claim 2: the parallel keypoint branch is necessary.** This claim is only partially supported if interpreted broadly, but is strongly supported for the semi-dense matcher that motivated the original argument. In sparse XFeat, our non-parallel keypoint variant is nearly identical to the default model. In XFeat*, however, it causes a clear and repeatable drop in performance, especially on Megadepth. We conclude that the dedicated keypoint branch is important when the descriptor must also support refinement and dense matching, but it is not necessary for every architecture.

**Claim 3: one skip-connection is beneficial and additional skip-connections do not help.** This is the least supported claim. Our results confirm that removing all skip-connections is generally detrimental to sparse XFeat, indicating that some form of feature reuse is beneficial. However, we do not find evidence that the specific single skip-connection proposed in the paper is uniquely optimal. Alternative designs, including input-routed and ResNet-style skip-connections, match or slightly exceed the default on Megadepth, and in the semi-dense setting, even the skip-free variant can outperform the default configuration. A likely explanation is that XFeat's backbone is relatively shallow, limiting the role of long-range feature propagation. The observed variation across datasets further suggests that skip-connection effects are data-dependent. Overall, while skip-connections are useful, our results do not support the stronger claim that a single carefully placed skip-connection is sufficient or uniquely preferable.

**Claim 4: competitiveness on downstream tasks.** This claim holds for homography estimation and only partially for visual localization. HPatches is reproduced well and confirms that XFeat remains competitive beyond pose estimation. In contrast, Aachen reveals a substantial reproducibility gap. Because this gap appears even when re-evaluating the original checkpoint, our results suggest that the paper's localization numbers depend on implementation details that cannot be easily recovered from the paper alone. This does not invalidate the claim that XFeat can serve as a practical local feature method, but it does weaken the strength of the originally published experiments.

### 5.2 What was easy

The easiest part of the reproduction was rebuilding and training the model using the paper and code repository. The public code and checkpoint release supported our reproduction, and the supplementary material was especially useful because it clarified the intended backbone architecture and training procedure. In this respect, XFeat was easy to reproduce and apply to various real-world problems.

### 5.3 What was difficult

The hardest part was resolving ambiguities. The paper, supplement, and code disagree in several places, so a faithful re-implementation still requires explicit implementation decisions. The losses are the clearest example: the written objective and the released training loop are not the same. The downstream evaluations were also substantially more difficult than reproducing the core model. Aachen is computationally expensive, time-consuming to debug, and sensitive to hyperparameters in the localization library. Additionally, the original paper did not provide the full HPatches configuration, which forced us to search over RANSAC thresholds to find the one that worked best in our reproduction. These examples highlight the importance of reproducing papers in the scientific process.

### 5.4 Communication with original authors

We did not contact the original authors; all implementation decisions were based solely on the paper, the supplement, the repository, and empirical validation using the released checkpoint.

## 6 Conclusion

This reproduction confirms the key message of XFeat: a lightweight local feature architecture can achieve a strong accuracy–efficiency trade-off without relying on specialized hardware. We reimplemented the XFeat architecture and the training losses from the paper and the supplementary material, and implemented our own ablation variants. We reused the authors' released checkpoint, refactored the benchmark scripts for Megadepth-1500 and ScanNet-1500, and implemented our own pipelines for HPatches and Aachen following standard evaluation protocols. Our reproduced models closely match the released checkpoint on the two main pose benchmarks, and our experiments strongly support the paper's central efficiency claim. The parallel keypoint branch is important primarily for semi-dense matching, the skip connections are more nuanced than the original ablation suggests, and downstream localization is more sensitive to hyperparameters in the evaluation library than the published paper indicates. The method's main strength is efficiency, and our experiments show that this strength holds in independent re-implementation. We conclude that XFeat is well-suited for real-world applications where both efficiency and accuracy are crucial, such as robotics and VR environments.

## References

Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5173–5182, 2017.

Daniel Barath, Jana Noskova, Maksym Ivashechkin, and Jiri Matas. MAGSAC++, a fast, reliable and accurate robust estimator. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1304–1312, 2020.

Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, 2004.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2041–2050, 2018.

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 *European conference on computer vision*, pp. 740–755. Springer, 2014.

Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5):1255–1262, 2017.

Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

Guilherme Potje, Felipe Cadar, André Araujo, Renato Martins, and Erickson R. Nascimento. XFeat: Accelerated Features for Lightweight Image Matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2682–2691, 2024.

Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12716–12725, 2019.

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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8601–8610, 2018.

Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

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.