ParaFormer: Parallel Attention Transformer for Efficient Feature Matching

Xiaoyong Lu[1], Yaping Yan[1], Bin Kang[2], Songlin Du[1]*

[1]Southeast University, Nanjing, China
[2]Nanjing University of Posts and Telecommunication, Nanjing, China
{220211846, yan, sdu}@seu.edu.cn, kangb@njupt.edu.cn

Abstract

Heavy computation is a bottleneck limiting deep-learning-based feature matching algorithms to be applied in many real-time applications. However, existing lightweight networks optimized for Euclidean data cannot address classical feature matching tasks, since sparse keypoint based descriptors are expected to be matched. This paper tackles this problem and proposes two concepts: 1) a novel parallel attention model entitled ParaFormer and 2) a graph based U-Net architecture with attentional pooling. First, ParaFormer fuses features and keypoint positions through the concept of amplitude and phase, and integrates self- and cross-attention in a parallel manner which achieves a win-win performance in terms of accuracy and efficiency. Second, with U-Net architecture and proposed attentional pooling, the ParaFormer-U variant significantly reduces computational complexity, and minimize performance loss caused by downsampling. Sufficient experiments on various applications, including homography estimation, pose estimation, and image matching, demonstrate that ParaFormer achieves state-of-the-art performance while maintaining high efficiency. The efficient ParaFormer-U variant achieves comparable performance with less than 50% FLOPs of the existing attention-based models.

Introduction

Feature matching is a fundamental problem for many computer vision tasks, such as object recognition (Liu et al. 2008), structure from motion (SFM) (Schonberger and Frahm 2016), and simultaneous localization and mapping (SLAM) (Engel, Koltun, and Cremers 2017). But with illumination changes, viewpoint changes, motion blur and occlusion, it is challenging to find the invariance and get robust matches from two images.

Feature matching pipelines can be categorized into detector-based methods, which first detect keypoints and descriptors from the images and then match two sets of sparse features, and detector-free methods, which directly match dense features. Benefiting from the global modeling capability of Transformer (Vaswani et al. 2017), attention-based networks become dominant methods in both detector-based and detector-free pipelines, where self- and cross-attention are applied to match learning-based descriptors or dense features. However, despite the high performance, attention-based networks tend to bring high training costs, large memory requirements, and high inference latency, especially for detector-free pipelines, where processing dense features exacerbates the problem of quadratic complexity of attention mechanism. So we focus on *detector-based* pipeline, seeking the best trade-off between efficiency and performance.

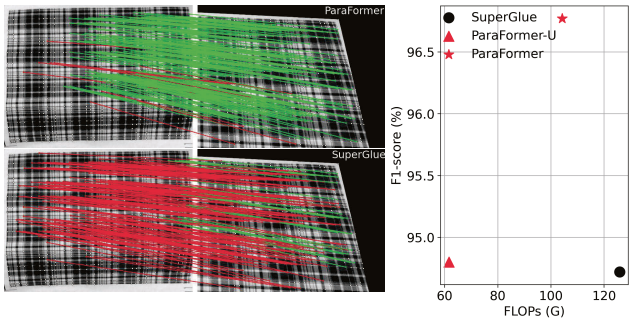

Figure 1: Comparison between the ParaFormer and SuperGlue. With the same input features, ParaFormer can deliver more robust matches with higher matching precision. ParaFormer-U can achieve comparable performance to SuperGLue with significantly fewer FLOPs.

As most lightweight operations (Chollet 2017; Howard et al. 2017) are designed for Euclidean data, sparse descriptors cannot be handled by mainstream lightweight networks. Note that Transformer and Graph Neural Networks are suitable for processing non-Euclidean data, so we design efficient models from both perspectives, giving birth to the ParaFormer and its ParaFormer-U variant.

Rethinking the self- and cross-attention in feature matching, all existing attention-based methods arrange two kinds of attention in a serial manner, a strategy derived from the behavior of people looking back and forth when matching images. Specifically, SuperGlue (Sarlin et al. 2020) and LoFTR (Sun et al. 2021) alternately arrange self- and cross-attention, *i.e.*, the $self \rightarrow cross$ strategy as illustrated in Figure 2 (a). For MatchFormer (Wang et al. 2022), the $self \rightarrow self \rightarrow cross$ strategy is used in the early stages, and the $self \rightarrow cross \rightarrow cross$ strategy is used in later

*Corresponding author

stages as shown in Figure 2 (b). However, computer vision is not necessarily designed based on human behavior, the fixed serial attention structure limits the diversity of the integration of self- and cross-attention. We propose parallel attention to compute self- and cross-attention synchronously, and train the network to optimally fuse two kinds of attention instead of tuning the permutation of both as a hyperparameter.

For the efficiency of the attention-based feature matching, instead of simply applying attention variants (Shen et al. 2021; Wang et al. 2021), weight sharing and attention weight sharing strategies in the parallel attention layer are explored to reduce redundant parameters and computations. We further construct the U-Net architecture with parallel attention layers and propose attentional pooling, which identifies important context points by attention weights.

In summary, the contributions of this paper include:

- We rethink the attention-based feature matching networks, and propose the **parallel attention layer** to perform self- and cross-attention synchronously and adaptively integrate both with learnable networks.
- We further explore the **U-Net architecture** for efficient feature matching and propose attentional pooling, which keeps only the important context points to reduce the FLOPs with minimal performance loss.
- A novel **wave-based position encoder** is proposed for detector-based feature matching networks, which dynamically fuses descriptors and positions through the concepts of amplitude and phase of waves.

Related Works

Local Feature Matching. Classical feature matching tends to be a detector-based pipeline, *i.e.*, the detector is first applied to generate keypoints and descriptors from images, and then the descriptors are matched. For detectors, some outstanding handcrafted methods (Lowe 2004; Bay, Tuytelaars, and Van G. 2006; Calonder et al. 2010; Rublee et al. 2011) were first proposed and widely used for various 3D computer vision tasks. With the advent of the deep learning era, many learning-based detectors (Revaud et al. 2019; DeTone, Malisiewicz, and Rabinovich 2018; Dusmanu et al. 2019; Ono et al. 2018) have been proposed to further improve the robustness of descriptors under illumination changes and viewpoint changes. In addition to detectors, other work has focused on better matchers. SuperGlue (Sarlin et al. 2020) was the first to propose an attention-based feature matching network that uses self- and cross-attention to find matches with global context information. OETR (Chen et al. 2022) further constrains attention-based feature matching in the commonly visible region by overlap estimation.

Besides matching the sparse descriptors generated by the detector, LoFTR (Sun et al. 2021) applies self- and cross-attention directly on the feature maps extracted by convolutional neural network (CNN) and generates matches in a coarse-to-fine manner. MatchFormer (Wang et al. 2022) further abandons the CNN backbone and adopts a completely attention-based hierarchical framework that can extract features while finding similarities utilizing the attention mechanism. Noting that the permutation of self- and cross-attention in SuperGlue and LoFTR is a simple alternating strategy, MatchFormer further proposes an interleaving strategy, which focuses on self-attention at the shallow stages of the network and cross-attention at the deep stages. This improvement gives us inspiration about the permutation of self- and cross-attention.

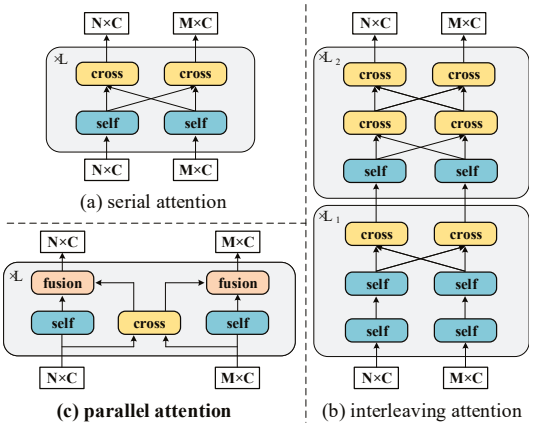

Figure 2: Conceptual difference among three attention architectures. (a) Serial attention in SuperGlue. (b) Interleaving attention in MatchFormer. (c) Proposed parallel attention.

All existing attention-based approaches artificially arrange self- and cross-attention in a serial manner to mimic human behavior, which does not take advantage of the benefits of deep learning network and parallel computing. We propose to compute two kinds of attention efficiently in a parallel manner, and let the network learn the optimal way to integrate the two kinds of attention.

Position Encoder. The position encoder is a critical part for all transformer-based networks, which allows the network to sense the relative or absolute position of each vector in a sentence or image. The first proposed position encoding method (Vaswani et al. 2017) uses fixed sine and cosine functions to calculate the position encoding or uses the position encoding vector as a learnable parameter, and finally adds the position encoding to the original vector. Although position information can be provided, this approach severely limits the flexibility of the model because the position encodings are fixed-length at training time, which limits the model to only process fixed-length inputs at inference time.

Another way of position encoding is relative position encoding (Liu et al. 2021), *i.e.*, adjusting attention weights with relative position. However, it is not only computationally intensive but also needs to handle inputs of different lengths by interpolation, which severely damages the performance. Convolution-based position encoders (Chu et al. 2021; Wang et al. 2022) are proposed to augment local features with convolution and enable the model to be aware of position information with zero padding. But this method can only be applied to Euclidean data such as feature maps, thus it cannot be applied to methods based on sparse descriptors.

To handle arbitrary length and non-Euclidean inputs, SuperGlue proposes a position encoder based on the multilayer perceptron (MLP), which uses MLP to extend the co-

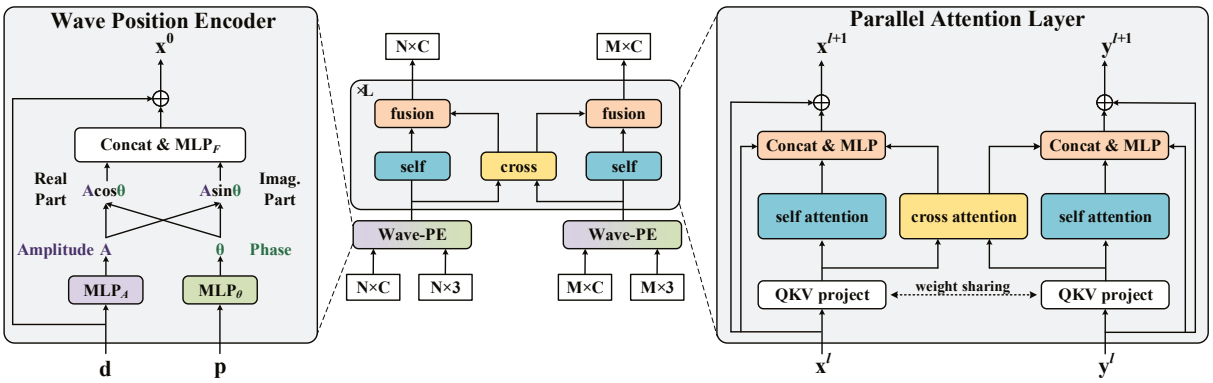

Figure 3: ParaFormer architecture. Wave-PE fuses the amplitude $\mathbf{A}$ estimated with the descriptor $\mathbf{d}$ and the phase $\boldsymbol{\theta}$ estimated with the position $\mathbf{p}$ to generate position encoding. Stacked parallel attention layers utilize self- and cross-attention to enhance the descriptors and find potential matches, where self- and cross-attention are adaptively integrated through a learnable network.

ordinate vector to align with the dimension of the descriptor to get the position encoding. However, the weak encoding ability becomes the bottleneck of the matching network. Inspired by Wave-MLP (Tang et al. 2022), phase information is equally important in vectors compared to amplitude information. Wave-MLP encodes the same input as both amplitude and phase information, while we encode the descriptor as amplitude information and the position as phase information, then fuse the two types of information with the Euler formula to generate position-aware descriptors.

U-Net Architecture. The U-Net (Ronneberger, Fischer, and Brox 2015) architecture consists of an encoder-decoder structure, where the encoder reduces the spatial resolution and the decoder recovers it. This architecture can efficiently handle dense prediction tasks such as semantic segmentation, so we seek to improve the efficiency of attention-based feature matching with U-Net. However conventional pooling operations cannot be directly applied to non-Euclidean data like sparse descriptors, so Graph U-Nets (Gao and Ji 2019) proposes the graph pooling (gPool) layer to enable downsampling on graph data in a differentiable way. The gPool layer measures the information that can be retained by each feature vector through scalar projection and applies topk sampling so that the new graph preserves as much information as possible from the original graph. Based on the gPool layer, we propose to utilize attention weights to measure how much information can be retained by each feature vector, which can better cooperate with the attention-based network without introducing additional parameters.

Methodology

Assuming that *M* and *N* keypoints are detected in image *X* and image *Y*, we let the positions be $\mathbf{p}^X \in \mathbb{R}^{M\times 3}$, $\mathbf{p}^Y \in \mathbb{R}^{N\times 3}$ and the descriptors be $\mathbf{d}^X \in \mathbb{R}^{M\times C}$, $\mathbf{d}^Y \in \mathbb{R}^{N\times C}$. As illustrated in Figure 3, our proposed method first dynamically fuses positions $\mathbf{p}$ and descriptors $\mathbf{d}$ in amplitude and phase manner with wave position encoder (Wave-PE). The parallel attention module is then applied to compute self- and cross-attention synchronously, utilizing global information to enhance the representation capability of features and find potential matches. $\mathbf{x}^l$ and $\mathbf{y}^l$ denote the intermediate features of image *X* and *Y* in layer *l*. Finally, the enhanced descriptors are matched by the optimal matching layer (Sarlin et al. 2020) appliying the Sinkhorn algorithm.

Wave Position Encoder

For the MLP-based position encoder (MLP-PE), the main drawback is the limited encoding capacity because the parameters of MLP-PE are less than 1% of the whole network, yet the position information is important for feature matching. Therefore, Wave-PE is designed to dynamically adjust the relationship between descriptor and position by amplitude and phase to obtain better position encoding.

In Wave-PE, position encoding is represented as a wave $\tilde{\boldsymbol{w}}$ with both amplitude $\mathbf{A}$ and phase $\boldsymbol{\theta}$ information, and the Euler formula is employed to unfold waves into real parts and imaginary parts to process waves efficiently,

$$\begin{aligned}\tilde{\boldsymbol{w}}_j &= \mathbf{A}_j \odot e^{i\boldsymbol{\theta}_j} \\ &= \mathbf{A}_j \odot \cos\boldsymbol{\theta}_j + i\mathbf{A}_j \odot \sin\boldsymbol{\theta}_j, j = 1, 2, ..., n.\end{aligned} \tag{1}$$

As shown in Figure 3, the amplitude and phase are estimated by two learnable networks via descriptors and position, respectively. Then a learnable network is applied to fuse the real and imaginary parts into position encoding,

$$\begin{aligned}&\mathbf{A}_j = MLP_A(\mathbf{d}_j), \\ &\boldsymbol{\theta}_j = MLP_\theta(\mathbf{p}_j), \\ &\mathbf{x}_j^0 = \mathbf{d}_j + MLP_F([\mathbf{A}_j \odot \cos\boldsymbol{\theta}_j, \mathbf{A}_j \odot \sin\boldsymbol{\theta}_j]).\end{aligned} \tag{2}$$

$[\cdot,\cdot]$ denotes concatenation. For three learnable networks in equation (2), two-layer MLP is chosen for simplicity.

Parallel Attention

As illustrated in the right side of Figure 3, the two sets of descriptors are first linearly projected as $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$. Then self- and cross-attention are computed in a parallel manner. In the self-attention module, standard attention computation $softmax(\boldsymbol{Q}\boldsymbol{K}^T/\sqrt{d})\boldsymbol{V}$ is employed, where $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$ come from the same input, *i.e.*, $(\boldsymbol{Q}_x, \boldsymbol{K}_x, \boldsymbol{V}_x)$ or $(\boldsymbol{Q}_y, \boldsymbol{K}_y, \boldsymbol{V}_y)$. In the cross-attention module, the attention

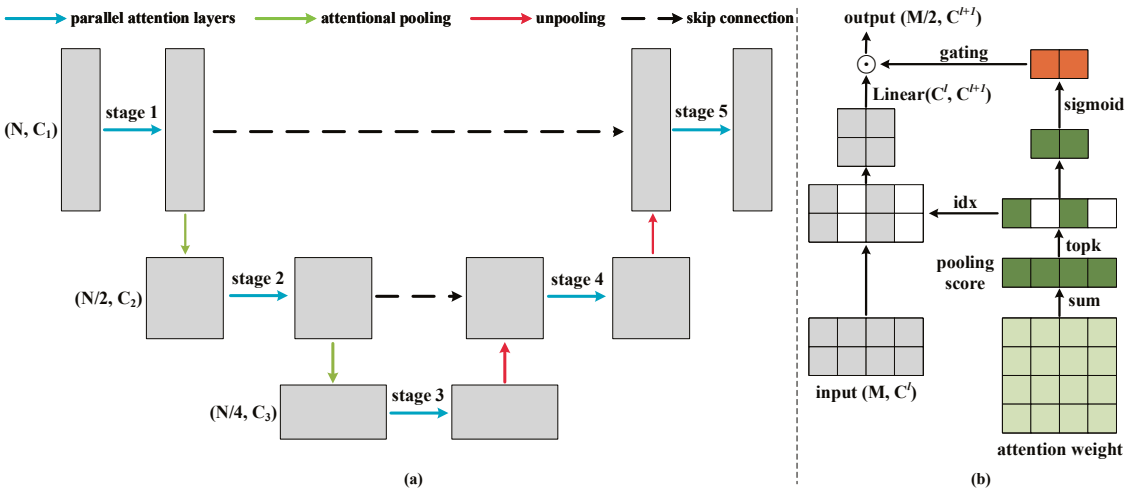

Figure 4: (a) U-Net architecture. The descriptors are processed in an encoder-decoder fashion. After the stages 1 and 2, the descriptors are downsampled with attention pooling to filter out the insignificant descriptors. After the stages 4 and 5, the descriptors are upsampled and fused with descriptors in previous stage by skip connections. (b) Attentional pooling. Pooling scores are computed from attention weights to identify context points and provide the gating signal through *sigmoid* function.

weight sharing strategy is proposed to improve model efficiency, which is replacing $\boldsymbol{Q}_y\boldsymbol{K}_x^T$ with $(\boldsymbol{Q}_x\boldsymbol{K}_y^T)^T$, so the input of the cross-attention module is $(\boldsymbol{Q}_x, \boldsymbol{V}_x, \boldsymbol{K}_y, \boldsymbol{V}_y)$. The impact of weight sharing and attention weight sharing is investigated in the ablation studies. Finally, self- and cross-attention outputs are fused by a two-layer MLP. Parallel attention saves redundant parameters and computations while boosting performance through learnable fusion.

Since the parallel attention layer updates two sets of descriptors simultaneously, it is formally similar to the self-attention layer in mainstream Transformers, except with two inputs. We can simply stack L parallel attention layers to form ParaFormer and conveniently explore various architectures like U-Net architecture to design model variants.

U-Net Architecture

As shown in Figure 4 (a), ParaFormer-U is designed for efficiency. Spatial downsampling is performed first to extract the high-level semantic information, then upsampling is performed to recover the spatial information, and the low-level and high-level information are fused by skip connections.

As illustrated in Figure 4 (b), attentional pooling is proposed for downsampling. Observing the attention map by column shows that certain points have strong weight with all other points, indicating that they are important context points in the image. Suppose the feature in layer l is $\mathbf{x}^l \in \mathbb{R}^{N\times D}$ and the attention weight is $\mathbf{A}^l \in \mathbb{R}^{N\times N}$. Our proposed attentional pooling is defined as

$$\begin{aligned}
\mathbf{s} &= sum(\mathbf{A}, dim = 1),\\
\mathbf{idx} &= rank(\mathbf{s}, \mathbf{k}),\\
\tilde{\mathbf{x}}^l &= Linear(\mathbf{x}^l(\mathbf{idx}, :)),\\
\mathbf{g} &= sigmoid(\mathbf{s}(\mathbf{idx}))\\
\mathbf{x}^{l+1} &= \tilde{\mathbf{x}}^l \odot \mathbf{g}.
\end{aligned} \tag{3}$$

$\mathbf{k}$ is the number of points selected for next layer l+1, which is set to half the number of points in previous layer l. The sum of each column of the self-attention map is computed as the pooling score $\mathbf{s} \in \mathbb{R}^N$, which measures the importance of each point. Then topk points are selected based on the attentional pooling score to filter out insignificant points, and the *Linear* layer is used to adjust the dimension size of the descriptors. $\mathbf{s}(\mathbf{idx})$ extracts values in $\mathbf{s}$ with indices $\mathbf{idx}$ followed by a *sigmoid* operation to generate gating signal, and $\odot$ represents the element-wise matrix multiplication.

Following (Gao and Ji 2019), the unpooling operation is defined as

$$\begin{aligned}
\tilde{\mathbf{x}}^l &= Linear(\mathbf{x}^l),\\
\mathbf{x}^{l+1} &= distribute(\mathbf{0}_{N\times C^{l+1}}, \tilde{\mathbf{x}}^l, \mathbf{idx}),
\end{aligned} \tag{4}$$

where $\mathbf{x}^l \in \mathbb{R}^{k\times C^l}$ is the current feature matrix and $\mathbf{0}_{N\times C^{l+1}}$ initially empty feature matrix for the next layer. The *Linear* layer is employed first to adjust the feature matrix dimension. $\mathbf{idx} \in \mathbb{R}^k$ is the indices of points selected in the corresponding pooling layer. Then the current feature matrix is inserted into the corresponding row of the empty feature matrix according to $\mathbf{idx}$, while the other rows remain zero. In other words, the unselected features in the pooling layer are represented by zero vectors to perform upsampling.

Implementation Details

The homography model is pretrained on the $\mathcal{R}$1M dataset (Radenović et al. 2018), and then the model is finetuned on the MegaDepth dataset (Li and Snavely 2018) for outdoor pose estimation and image matching tasks. On the $\mathcal{R}$1M dataset, we employ the AdamW (Kingma and Ba 2014) optimizer for 10 epochs using the cosine decay learning rate scheduler and 1 epoch of linear warm-up. A batch size of 8 and an initial learning rate of 0.0001 are used. On the MegaDepth dataset, we use the same AdamW optimizer for

Matcher	AUC	Precision	Recall	F1-score
NN	39.47	21.7	65.4	32.59
NN + mutual	42.45	43.8	56.5	49.35
NN + PointCN	43.02	76.2	64.2	69.69
NN + OANet	44.55	82.8	64.7	72.64
SuperGlue	52.65	90.9	98.88	94.72
ParaFormer-U	53.16	90.93	99.01	94.80
ParaFormer	**54.91**	**94.55**	**99.10**	**96.77**

Table 1: Homography estimation on $\mathcal{R}$1M. AUC @10 pixels is reported. The best method is highlighted in bold.

50 epochs using the same learning rate scheduler and linear warm-up. A batch size of 2 and a lower initial learning rate of 0.00001 are used. For training on $\mathcal{R}$1M/MegaDepth dataset, we resize the images to 640×480/960×720 pixels and detect 512/1024 keypoints, respectively. When the detected keypoints are not enough, random keypoints are added for efficient batching. All models are trained on a single NVIDIA 3070Ti GPU. For ParaFormer, we stack $L = 9$ parallel attention layers, and all intermediate features have the same dimension $C = 256$. For ParaFormer-U, the depth of each stage is $\{2, 1, 2, 1, 2\}$, resulting in a total of $L = 8$ parallel attention layers, and the intermediate feature dimension of each stage is $\{256, 384, 128, 384, 256\}$. More details are provided in the supplementary material.

Experiments

Homography Estimation

Dataset. We split $\mathcal{R}$1M dataset (Radenović et al. 2018), which contains over a million images of Oxford and Paris, into training, validation, and testing sets. To perform self-supervised training, random ground-truth homographies are generated to get image pairs.

Baselines. SuperPoint (DeTone, Malisiewicz, and Rabinovich 2018) is applied as the unified descriptor to generate the input for the matcher. ParaFormer and ParaFormer-U are compared with attention-based matcher SuperGlue (Sarlin et al. 2020) and NN matcher with learning-based outlier rejection methods (Yi et al. 2018; Zhang et al. 2019). The results of SuperGlue are from our own implementation.

Metrics. Precision and recall are computed based on ground truth matches. The area under the cumulative error curve (AUC) up to a value of 10 pixels is reported, where the reprojection error is computed with the estimated homography.

Results. As shown in Table 1, ParaFormer outperforms all outlier rejection methods and attention-based matcher on homography estimation. It can be seen that the attention-based approaches have a remarkable superiority due to the global receptive field of attention. Compared with the attention-based approach SuperGlue, ParaFormer further boosts the performance by integrating self- and cross-attention with parallel attention layers, bringing a $+2.05\%$ improvement on the F1-score over SuperGlue. The visualization of matches can be found in Figure 6. Moreover, compared to SuperGlue, our efficient U-Net variant has only 49% FLOPs, yet achieves better performance.

Matcher	Exact AUC			Approx. AUC		
	@5°	@10°	@20°	@5°	@10°	@20°
NN + mutual	16.94	30.39	45.72	35.00	43.12	54.25
NN + OANet	26.82	45.04	62.17	50.94	61.41	71.77
SuperGlue	28.45	48.6	67.19	55.67	66.83	74.58
ParaFormer-U	29.40	49.76	68.29	56.47	67.66	75.67
ParaFormer	**31.73**	**52.28**	**70.43**	**60.05**	**70.72**	**78.13**

Table 2: Pose estimation on YFCC100M. ParaFormer and ParaFormer-U lead other methods at all thresholds.

Outdoor Pose Estimation

Dataset. ParaFormer is trained on the MegaDepth dataset (Li and Snavely 2018) and evaluated on the YFCC100M dataset (Thomee et al. 2016). For training, 200 pairs of images in each scene are randomly sampled for each epoch. For evaluation, the YFCC100M image pairs and ground truth poses provided by SuperGlue are used.

Baselines. SuperPoint is applied as the descriptor and combined with baseline matchers, which contain attention-based matcher SuperGlue and NN matcher with outlier rejection methods (Lowe 2004; Zhang et al. 2019). The results of SuperGlue are from our own implementation.

Metrics. The AUC of the pose error at thresholds (5°, 10°, 20°) are reported. Evaluation is performed with both approximate AUC (Zhang et al. 2019) and exact AUC (Sarlin et al. 2020) for a fair comparison.

Results. As shown in Table 2, ParaFormer achieves the best performance at all thresholds, demonstrating the robustness of our models. With wave position encoder and parallel attention architecture, ParaFormer can bring $(+3.28\%, +4.2\%, +3.24\%)$ improvement on exact AUC and $(+4.38\%, +3.89\%, +3.55\%)$ improvement on approximate AUC at three thresholds of $(5^\circ, 10^\circ, 20^\circ)$, respectively. In outdoor scenes with a large number of keypoints, ParaFormer-U can effectively alleviate the computational complexity problem by downsampling, while still maintaining state-of-the-art performance by attentional pooling.

Image Matching

Dataset. We follow the evaluation protocol as in D2-Net (Dusmanu et al. 2019) and evaluate our methods on 108 HPatches (Balntas et al. 2017) sequences, which contain 52 sequences with illumination changes and 56 sequences with viewpoint changes.

Baselines. Baseline methods include learning-based descriptors R2D2, D2Net and SuperPoint (Revaud et al. 2019; Dusmanu et al. 2019; DeTone, Malisiewicz, and Rabinovich 2018) and advanced matchers LoFTR, Patch2Pix, SuperGlue and CAPS (Sun et al. 2021; Zhou, Sattler, and Leal-Taixe 2021; Sarlin et al. 2020; Wang et al. 2020). The results of SuperGlue are from our own implementation.

Metrics. A match is considered correct if the reprojection error is below the matching threshold, where the reprojection error is computed from the homographies provided by the dataset. The matching threshold is varied from 1 to 10

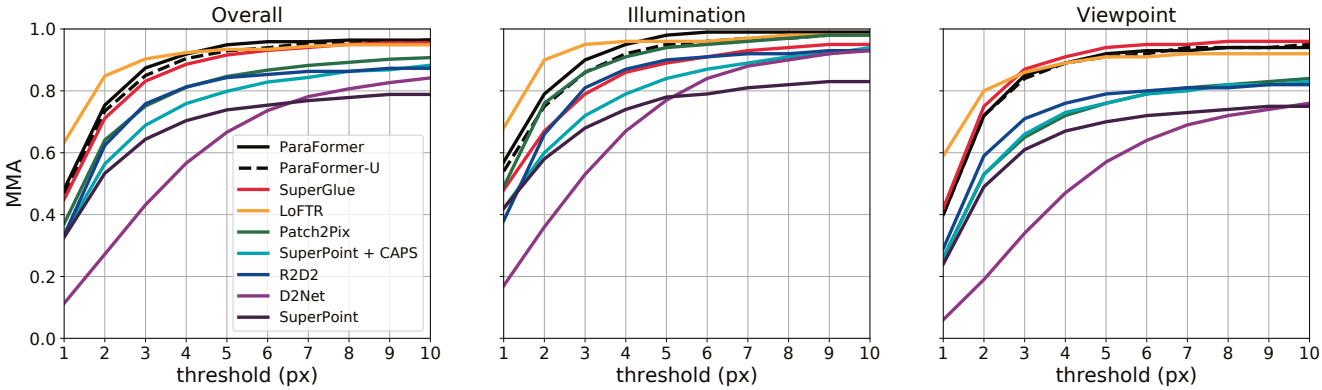

Figure 5: Image matching on HPatches. The mean matching accuracy (MMA) at thresholds from 1 to 10 pixels are reported.

SA	**PA**	MLP-PE	**Wave-PE**	Precision	Recall	F1-score
✓		✓		86.68	96.56	91.35
	✓	✓		86.79	98.23	92.16
	✓		✓	**87.69**	**98.73**	**92.88**

Table 3: Ablation study on main designs.

FFN	QKV proj	Head Merging	#Params (M)	F1-score
			1.70	88.79
✓			**1.18**	80.86
	✓		1.51	90.15
		✓	1.51	**90.16**
	✓	✓	1.31	90.02

Table 4: Ablation study on weight sharing.

attention weight sharing	FLOPs (G)	F1-score
	108.72	89.98
✓	**99.05**	**90.02**

Table 5: Ablation study on attention weight sharing.

random pooling	gPool	**attentional pooling**	F1-score
✓			90.87
	✓		90.95
		✓	**91.23**

Table 6: Ablation study on pooling.

to plot the mean matching accuracy (MMA), which is the average percentage of correct matches for each image.

Results. As shown in Figure 5, ParaFormer achieves the best overall performance at matching thresholds of 5 or more pixels. The results indicate that detector-based methods such as SuperGlue and our methods are better at handling scenarios with large viewpoint changes, while the detector-free methods such as LoFTR are better suited to address illumination changes. But ParaFormer still outperforms LoFTR in illumination change experiments, benefiting from the superior modeling capability of parallel attention. With ParaFormer and ParaFormer-U, the overall performance of SuperPoint grows from the last to the first and the second place, demonstrating the effectiveness of our matchers.

ParaFormer Structural Study

A complete ablation study is conducted on the $\mathcal{R}$1M dataset for further understanding of our designs. The FLOPs and runtimes between our methods and SuperGlue are also compared to demonstrate the high efficiency of our methods.

Main Designs. We did ablation experiments on parallel attention architecture and Wave-PE. As can be seen from Table 3, when both use MLP-PE, parallel attention leads serial attention on both precision and recall, resulting in a $+0.81\%$ improvement on F1-score. When parallel attention is combined with Wave-PE, the performance can be further boosted by $+0.72\%$ over MLP-PE, indicating that Wave-PE provides stronger position information to guide matching.

Weight Sharing. As shown in Table 4, we conduct ablation experiments on weight sharing strategies and find that the performance of the network improved when self-attention and cross-attention share the $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$ projection weights and the multi-head merging weights, while it also helps to reduce the model parameters. This occurs because self- and cross-attention are essentially indistinguishable except for the inputs, and the shared weights align the $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$ projections of both inputs so that self- and cross-attention can be performed in the same vector space, which makes a uniform standard for the cosine similarity of both.

Attention Weight Sharing. We find that computing cross-attention twice is redundant for attention-based methods, because of the high correlation between $\boldsymbol{Q}_y\boldsymbol{K}_x^T$ and $\boldsymbol{Q}_x\boldsymbol{K}_y^T$. So attention weight sharing is proposed for efficiency, *i.e.*, replacing $\boldsymbol{Q}_y\boldsymbol{K}_x^T$ with $(\boldsymbol{Q}_x\boldsymbol{K}_y^T)^T$. As shown in Table 5, attention weight sharing can reduce FLOPs without performance loss, which makes a significant difference in scenarios with a large number of keypoints.

Attentional Pooling. As shown in Table 6, compared to gPool (Gao and Ji 2019) which gets pooling scores by linear projection, our proposed attentional pooling achieves better performance and saves the parameters of linear projection by identifying important context points by attention weights.

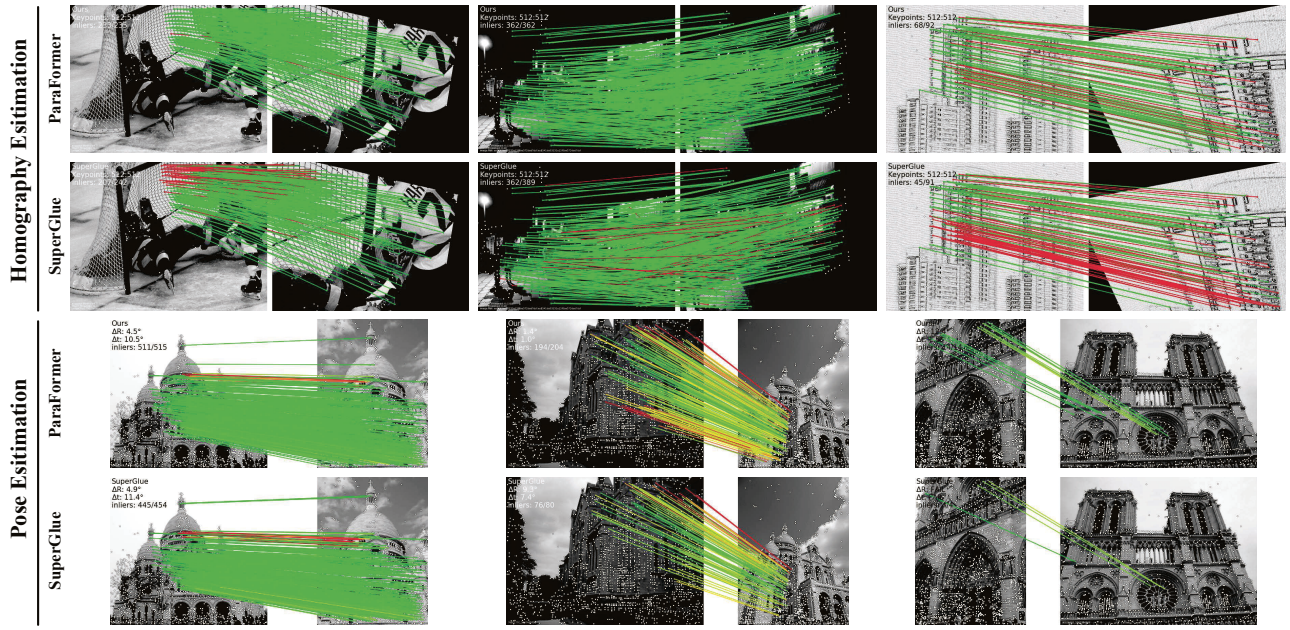

Figure 6: Qualitative results of homography estimation and pose estimation experiments.

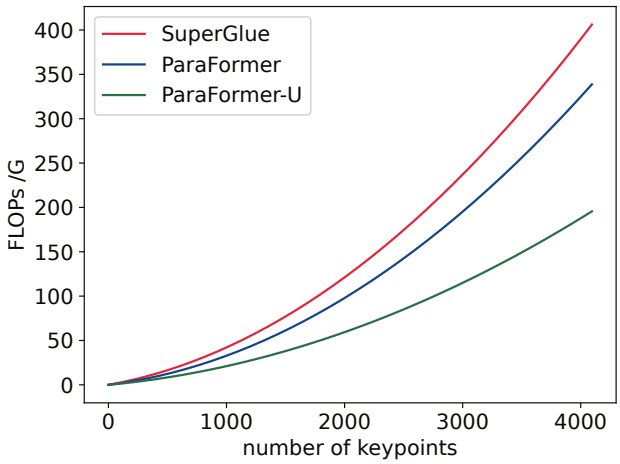

Figure 7: Comparison between FLOPs of models.

As expected, the strategy of computing pooling scores by features or attention weights is superior to random pooling.

Efficiency Analysis. Benefiting from the above designs, our model is remarkable in efficiency beyond just achieving state-of-the-art performance. As shown in Table 7, when matching 2048 descriptors, ParaFormer reduces FLOPs by 17.8% compared to SuperGlue with better performance. ParaFormer-U further improves efficiency with FLOPs of only 49% of SuperGlue, while it still outperforms SuperGlue due to the advantage of parallel attention and Wave-PE. As shown in Figure 7, the attention weight sharing strategy in ParaFormer alleviates the squared complexity of the attention mechanism, and the U-Net architecture further significantly reduces computational cost through downsampling.

Methods	F1-score	FLOPs (G)	Runtime (ms)
SuperGlue	90.68	125.85	26.99
ParaFormer-U	90.72	**61.67**	**20.23**
ParaFormer	**94.92**	104.22	24.99

Table 7: Efficiency analysis @2048 keypoints.

Conclusion

In this paper, we propose a novel attention-based network named ParaFormer to handle feature matching tasks efficiently. As a preprocessing module, the proposed Wave-PE dynamically fuses features and positions in amplitude and phase manner. In contrast to employing serial attention that intuitively mimics human behavior, we propose a parallel attention architecture that not only integrates self-attention and cross-attention in a learnable way but also saves redundant parameters and computations through weight sharing and attention weight sharing strategies. To further improve efficiency, the ParaFormer-U is designed with U-Net architecture, which reduces FLOPs by downsampling and minimizes the performance loss by the proposed attentional pooling. Experiments show that ParaFormer and ParaFormer-U deliver state-of-the-art performance with remarkable efficiency, enabling a broader application scenario for attention-based feature matching networks.

Acknowledgments

This work was jointly supported by the National Natural Science Foundation of China under grants 62001110, 62201142 and 62171232, the Natural Science Foundation of Jiangsu Province under grant BK20200353, and the China Postdoctoral Science Foundation under grant 2020M681684.

References

Balntas, V.; Lenc, K.; Vedaldi, A.; and Mikolajczyk, K. 2017. HPatches: A Benchmark and Evaluation of Handcrafted and Learned Local Descriptors. In *Proceedings of the CVPR*, 5173–5182.

Bay, H.; Tuytelaars, T.; and Van G., L. 2006. SURF: Speeded Up Robust Features. In *Proceedings of the ECCV*, 404–417.

Calonder, M.; Lepetit, V.; Strecha, C.; and Fua, P. 2010. BRIEF: Binary Robust Independent Elementary Features. In *Proceedings of the ECCV*, 778–792.

Chen, Y.; Huang, D.; Xu, S.; Liu, J.; and Liu, Y. 2022. Guide Local Feature Matching by Overlap Estimation. In *Proceedings of the AAAI*.

Chollet, F. 2017. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the CVPR*, 1251–1258.

Chu, X.; Tian, Z.; Zhang, B.; Wang, X.; Wei, X.; Xia, H.; and Shen, C. 2021. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*.

DeTone, D.; Malisiewicz, T.; and Rabinovich, A. 2018. SuperPoint: Self-Supervised Interest Point Detection and Description. In *Proceedings of the CVPRW*, 224–236.

Dusmanu, M.; Rocco, I.; Pajdla, T.; Pollefeys, M.; Sivic, J.; Torii, A.; and Sattler, T. 2019. D2-Net: A Trainable CNN for Joint Description and Detection of Local Features. In *Proceedings of the CVPR*, 8092–8101.

Engel, J.; Koltun, V.; and Cremers, D. 2017. Direct sparse odometry. *Journal of the PAMI*, 40(3): 611–625.

Gao, H.; and Ji, S. 2019. Graph U-Nets. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the ICML*, 2083–2092.

Howard, A. G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; and Adam, H. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Li, Z.; and Snavely, N. 2018. MegaDepth: Learning Single-View Depth Prediction from Internet Photos. In *Proceedings of the CVPR*, 2041–2050.

Liu, C.; Yuen, J.; Torralba, A.; Sivic, J.; and Freeman, W. T. 2008. Sift flow: Dense correspondence across different scenes. In *Proceedings of the ECCV*, 28–42.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. *Proceedings of the ICCV*, 10012–10022.

Lowe, D. G. 2004. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.*, 60(2): 91–110.

Ono, Y.; Trulls, E.; Fua, P.; and Yi, K. M. 2018. LF-Net: Learning Local Features from Images. In *Proceedings of the NeurIPS*, volume 31.

Radenović, F.; Iscen, A.; Tolias, G.; Avrithis, Y.; and Chum, O. 2018. Revisiting Oxford and Paris: Large-Scale Image Retrieval Benchmarking. In *Proceedings of the CVPR*, 5706–5715.

Revaud, J.; De Souza, C.; Humenberger, M.; and Weinzaepfel, P. 2019. R2D2: Reliable and Repeatable Detector and Descriptor. In *Proceedings of the NeurIPS*, 12414–12424.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Proceedings of the MICCAI*, 234–241.

Rublee, E.; Rabaud, V.; Konolige, K.; and Bradski, G. 2011. ORB: An efficient alternative to SIFT or SURF. In *Proceedings of the ICCV*, 2564–2571.

Sarlin, P.; DeTone, D.; Malisiewicz, T.; and Rabinovich, A. 2020. SuperGlue: Learning Feature Matching With Graph Neural Networks. In *Proceedings of the CVPR*, 4937–4946.

Schonberger, J. L.; and Frahm, J.-M. 2016. Structure-from-motion revisited. In *Proceedings of the CVPR*, 4104–4113.

Shen, Z.; Zhang, M.; Zhao, H.; Yi, S.; and Li, H. 2021. Efficient Attention: Attention With Linear Complexities. In *Proceedings of the WACV*, 3531–3539.

Sun, J.; Shen, Z.; Wang, Y.; Bao, H.; and Zhou, X. 2021. LoFTR: Detector-Free Local Feature Matching With Transformers. In *Proceedings of the CVPR*, 8922–8931.

Tang, Y.; Han, K.; Guo, J.; Xu, C.; Li, Y.; Xu, C.; and Wang, Y. 2022. An Image Patch Is a Wave: Phase-Aware Vision MLP. In *Proceedings of the CVPR*, 10935–10944.

Thomee, B.; Elizalde, B.; Shamma, D. A.; Ni, K.; Friedland, G.; Poland, D.; Borth, D.; and Li, L. J. 2016. YFCC100M: The New Data in Multimedia Research. *Commun. ACM*, 59(2): 64–73.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *Proceedings of the NeurIPS*, 5998–6008.

Wang, Q.; Zhang, J.; Yang, K.; Peng, K.; and Stiefelhagen, R. 2022. Matchformer: Interleaving attention in transformers for feature matching. In *Proceedings of the ACCV*, 2746–2762.

Wang, Q.; Zhou, X.; Hariharan, B.; and Snavely, N. 2020. Learning Feature Descriptors Using Camera Pose Supervision. In Vedaldi, A.; Bischof, H.; Brox, T.; and Frahm, J.-M., eds., *Proceedings of the ECCV*, 757–774.

Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; and Shao, L. 2021. Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction Without Convolutions. In *Proceedings of the ICCV*, 568–578.

Yi, K. M.; Trulls, E.; Ono, Y.; Lepetit, V.; Salzmann, M.; and Fua, P. 2018. Learning to Find Good Correspondences. In *Proceedings of the CVPR*, 2666–2674.

Zhang, J.; Sun, D.; Luo, Z.; Yao, A.; Zhou, L.; Shen, T.; Chen, Y.; Quan, L.; and Liao, H. 2019. Learning Two-View Correspondences and Geometry Using Order-Aware Network. In *Proceedings of the ICCV*, 5845–5854.

Zhou, Q.; Sattler, T.; and Leal-Taixe, L. 2021. Patch2Pix: Epipolar-Guided Pixel-Level Correspondences. In *Proceedings of the CVPR*, 4669–4678.