Efficient LoFTR: Semi-Dense Local Feature Matching with Sparse-Like Speed

Yifan Wang*   Xingyi He*   Sida Peng   Dongli Tan   Xiaowei Zhou†

Zhejiang University

Abstract

We present a novel method for efficiently producing semi-dense matches across images. Previous detector-free matcher LoFTR has shown remarkable matching capability in handling large-viewpoint change and texture-poor scenarios but suffers from low efficiency. We revisit its design choices and derive multiple improvements for both efficiency and accuracy. One key observation is that performing the transformer over the entire feature map is redundant due to shared local information, therefore we propose an aggregated attention mechanism with adaptive token selection for efficiency. Furthermore, we find spatial variance exists in LoFTR's fine correlation module, which is adverse to matching accuracy. A novel two-stage correlation layer is proposed to achieve accurate subpixel correspondences for accuracy improvement. Our efficiency optimized model is ∼ 2.5× faster than LoFTR which can even surpass state-of-the-art efficient sparse matching pipeline Super-Point + LightGlue. Moreover, extensive experiments show that our method can achieve higher accuracy compared with competitive semi-dense matchers, with considerable efficiency benefits. This opens up exciting prospects for large-scale or latency-sensitive applications such as image retrieval and 3D reconstruction. Project page: https: //zju3dv.github.io/efficientloftr/.

1. Introduction

Image matching is the cornerstone of many 3D computer vision tasks, which aim to find a set of highly accurate correspondences given an image pair. The established matches between images have broad usages such as reconstructing the 3D world by structure from motion (SfM) [1, 21, 29, 47] or SLAM system [34, 35], and visual localization [42, 44], etc. Previous methods typically follow a two-stage pipeline: they first detect [41] and describe [53] a set of keypoints on each image, and then establish keypoint correspondences by
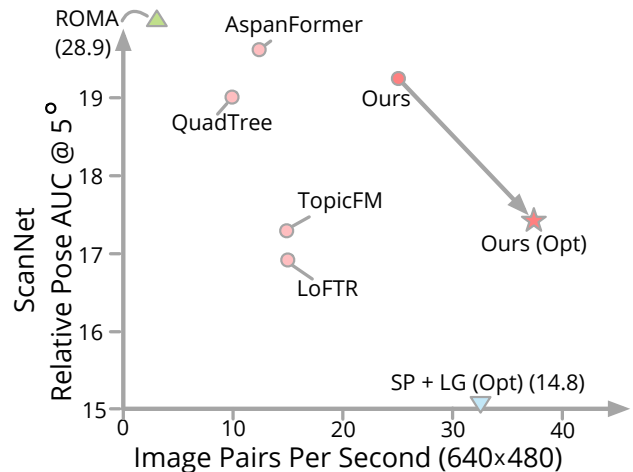

Figure 1. **Matching Accuracy and Efficiency Comparisons.** Our method achieves competitive accuracy compared with semi-dense matchers (●) at a significantly higher speed. Compared with dense matcher ROMA (▲), our method is ∼ 7.5× faster. Moreover, our efficiency optimized model (★) can surpass the robust sparse matching pipeline (▽) SuperPoint (SP) + LightGlue (LG) on efficiency with considerably better accuracy.

handcrafted [31] or learning-based matchers [30, 43]. These detector-based methods are efficient but suffer from robustly detecting repeatable keypoints across challenging pairs, such as extreme viewpoint changes and texture-poor regions.

Recently, LoFTR [50] introduces a detector-free matching paradigm with transformer to directly establish semi-dense correspondences between two images without detecting keypoints. With the help of the transformer mechanism to capture the global image context and the detector-free design, LoFTR shows a strong capability of matching challenging pairs, especially in texture-poor scenarios. To reduce the computation burden, LoFTR adopts a coarse-to-fine pipeline by first performing dense matching on downsampled coarse features maps, where transformer is applied. Then, the feature locations of coarse matches on one image are fixed, while their subpixel correspondences are searched on the other image by cropping feature patches based on coarse match, performing the feature correlation, and calculating expectation over the correlation patch.

Despite its impressive matching performance, LoFTR suffers from limited efficiency due to the large token size of performing transformer on the entire coarse feature map, which significantly barricades practical large-scale usages such as image retrieval [19] and SfM [47]. A large bunch of LoFTR's follow-up works [7, 18, 36, 52, 59] have attempted to improve its matching accuracy. However, there are rare methods that focus on matching efficiency of detector-free matching. QuadTree Attention [52] incorporates multi-scale transformation with a gradually narrowed attention span to avoid performing attention on large feature maps. This strategy can reduce the computation cost, but it also divides a single coarse attention process into multiple steps, leading to increased latency.

In this paper, we revisit the design decisions of the detector-free matcher LoFTR, and propose a new matching algorithm that squeezes out redundant computations for significantly better efficiency while further improving the accuracy. As shown in Fig. 1, our approach achieves the best inference speed compared with recent image matching methods while being competitive in terms of accuracy. Our key innovations lie in introducing a token aggregation mechanism for efficient feature transformation and a two-stage correlation layer for correspondence refinement. Specifically, we find that densely performing global attention over the entire coarse feature map as in LoFTR is unnecessary, as the attention information is similar and shared in the local region. Therefore, we devise an aggregated attention mechanism to perform feature transformation on adaptively selected tokens, which is significantly compact and effectively reduces the cost of local feature transformation.

In addition, we observe that there can be spatial variance in the matching refinement phase of LoFTR, which is caused by the expectation over the entire correlation patch when noisy feature correlation exists. To solve this issue, our approach designs a two-stage correlation layer that first locates pixel-level matches with the accurate mutual-nearest-neighbor matching on fine feature patches, and then further refines matches for subpixel-level by performing the correlation and expectation locally within tiny patches.

Extensive experiments are conducted on multiple tasks, including homography estimation, relative pose recovery, as well as visual localization, to show the efficacy of our method. Our pipeline pushes detector-free matching to unprecedented efficiency, which is ~ 2.5 times faster than LoFTR and can even surpass the current state-of-the-art efficient sparse matcher LightGlue [30]. Moreover, our framework can achieve comparable or even better matching accuracy compared with competitive detector-free baselines [7, 14, 15] with considerably higher efficiency.

In summary, this paper has the following contributions:

• A new detector-free matching pipeline with multiple improvements based on the comprehensive revisiting of LoFTR, which is significantly more efficient and with better accuracy.
• A novel aggregated attention network for efficient local feature transformation.
• A novel two-stage correlation refinement layer for accurate and subpixel-level refined correspondences.

2. Related Work

Detector-Based Image Matching. Classical image matching methods [4, 31, 41] adopt handcrafted critics for detecting keypoints, describing and then matching them. Recent methods draw benefits from deep neural networks for both detection [24, 41, 46] and description [13, 33, 53, 54], where the robustness and discriminativeness of local descriptors are significantly improved. Besides, some methods [10, 12, 32, 38, 55] managed to learn the detector and descriptor together. SuperGlue [43] is a pioneering method that first introduces the transformer mechanism into matching, which has shown notable improvements over classical handcrafted matchers. As a side effect, it also costs more time, especially with many keypoints to match. To improve the efficiency, some subsequent works, such as [6, 48], endeavor to reduce the size of the attention mechanism, albeit at the cost of sacrificing performance. LightGlue [30] introduces a new scheme for efficient sparse matching that is adaptive to the matching difficulty, where the attention process can be stopped earlier for easy pairs. It is faster than SuperGlue and can achieve competitive performance. However, robustly detecting keypoints across images is still challenging, especially for texture-poor regions. Unlike them, our method focuses on the efficiency of the detector-free method, which eliminates the restriction of keypoint detection and shows superior performance for challenging pairs.

Detector-Free Image Matching. Detector-free methods directly match images instead of relying on a set of detected keypoints, producing semi-dense or dense matches. NC-Net [39] represents all features and possible matches as a 4D correlation volume. Sparse NC-Net [40] utilizes sparse correlation layers to ease resolution limitations. Subsequently, DRC-Net [27] improves efficiency and further improves performance in a coarse-to-fine manner.

LoFTR [50] first employs the Transformer in detector-free matching to model the long-range dependencies. It shows remarkable matching capabilities, however, suffers from low efficiency due to the huge computation of densely transforming entire coarse feature maps. Many follow-up works further improve the matching accuracy. Matchformer [59] and AspanFormer [7] perform attention on multi-scale features, where local attention regions of [7] are found with the help of estimated flow. QuadTree [52] gradually restricts the attention span during hierarchical attention to relevant areas, which can reduce overall computation. How-
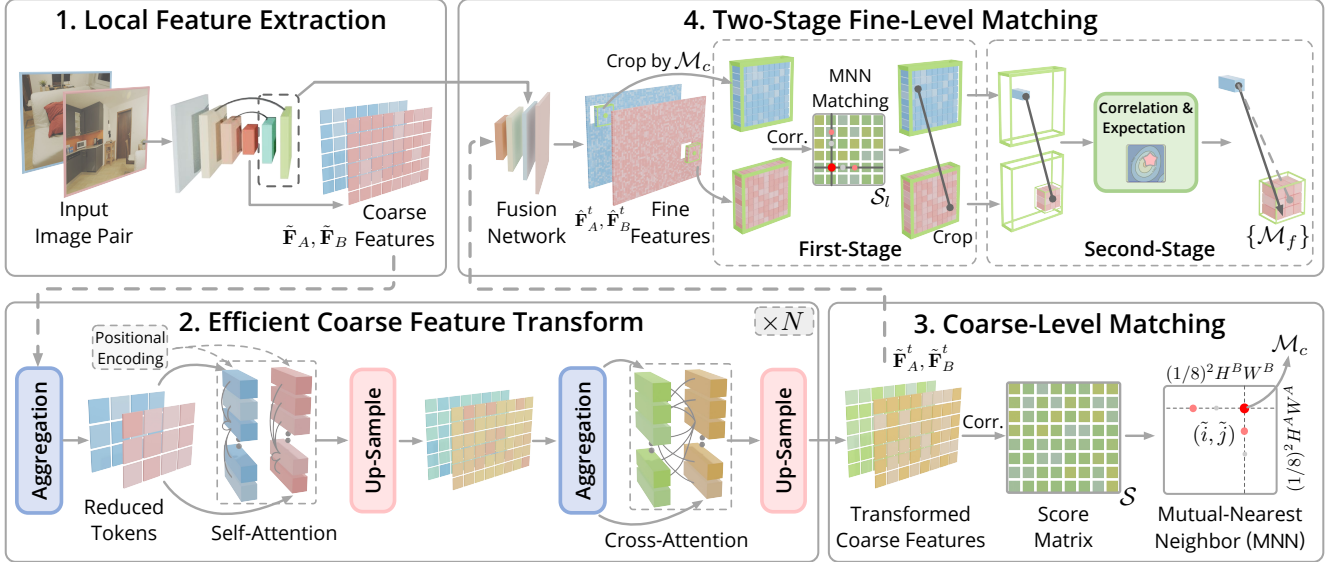
Figure 2. **Pipeline Overview. (1)** Given an image pair, a CNN network extracts coarse feature maps $\tilde{\mathbf{F}}_A$ and $\tilde{\mathbf{F}}_B$, as well as fine features. **(2)** Then, we transform coarse features for more discriminative feature maps by interleaving our aggregated self- and cross-attention N times, where adaptively feature aggregation is performed to reduce token size before each attention for efficiency. **(3)** Transformed coarse features are correlated for the score matrix $\mathcal{S}$. Mutual-nearest-neighbor (MNN) searching is followed to establish coarse matches $\{\mathcal{M}_c\}$. **(4)** To refine coarse matches, discriminative fine features $\hat{\mathbf{F}}_A^t$, $\hat{\mathbf{F}}_B^t$ in full resolution are obtained by fusing transformed coarse features $\tilde{\mathbf{F}}_A^t$, $\tilde{\mathbf{F}}_B^t$ with backbone features. Feature patches are then cropped centered at each coarse match $\mathcal{M}_c$. A two-stage refinement is followed to obtain sub-pixel correspondence $\mathcal{M}_f$.

ever, these designs contribute marginally or even decrease efficiency, since the hierarchical nature of multi-scale attention will further introduce latencies. TopicFM [18] first assigns features with similar semantic meanings to the same topic, where attention is conducted within each topic for efficiency. Since it needs to sequentially process each token's features for transformation, the efficiency improvement is limited. Moreover, performing local attention within topics can potentially restrict the capability of modeling long-range dependencies. Compared with them, the proposed aggregated attention module in our method significantly improves efficiency while achieving better accuracy.

Dense matching methods [14, 15, 56] are designed to estimate all possible correspondences between two images, which show strong robustness. However, they are generally much slower compared with sparse and semi-dense methods. Unlike them, our method produces semi-dense matches with competitive performance and considerably better efficiency.

Transformer has been broadly used in multiple vision tasks, including feature matching. The efficiency and memory footprint of handling large token sizes are the main limitations of transformer [58], where some methods [22, 23, 60] attempt to reduce the complexity to a linear scale to alleviate these problems. Some methods [9, 26] propose optimizing transformer models for specific hardware architectures for memory and running-time efficiency. They are orthogonal

to our method and can be naturally adapted into the pipeline for further efficiency improvement.

3. Method

Given a pair of images $\mathbf{I}_A, \mathbf{I}_B$, our objective is to establish a set of reliable correspondences between them. We achieve this by a coarse-to-fine matching pipeline, which first establishes coarse matches on downsampled feature maps and then refines them for high accuracy. An overview of our pipeline is shown in Fig. 2.

3.1. Local Feature Extraction

Image feature maps are first extracted by a lightweight backbone for later transformation and matching. Unlike LoFTR and many other detector-free matchers that use a heavy multi-branch ResNet [20] network for feature extraction, we alternate to a lightweight single-branch network with reparameterization [11] to achieve better inference efficiency while preserving the model performance.

In particular, a multi-branch CNN network with residual connections is applied during training for maximum representational power. At inference time, we losslessly convert the feature backbone into an efficient single-branch network by adopting the reparameterization technique [11], which is achieved by fusing parallel convolution kernels into a single one. Then, the intermediate $1/8$ down-sampled coarse fea-

3

tures $\tilde{\mathbf{F}}_A$, $\tilde{\mathbf{F}}_B$ and fine features in $1/4$ and $1/2$ resolutions are extracted efficiently for later coarse-to-fine matching.

3.2. Efficient Local Feature Transformation

After the feature extraction, the coarse-level feature maps $\tilde{\mathbf{F}}_A$ and $\tilde{\mathbf{F}}_B$ are transformed by interleaving self- and cross-attention [1] n times to improve discriminativeness. The transformed features are denoted as $\tilde{\mathbf{F}}_A^t$, $\tilde{\mathbf{F}}_B^t$.

Previous methods often perform attention on the entire coarse-level feature maps, where linear attention instead of vanilla attention is applied to ensure a manageable computation cost. However, the efficiency is still limited due to the large token size of coarse features. Moreover, the usage of linear attention leads to sub-optimal model capability. Unlike them, we propose efficient aggregated attention for both efficiency and performance.

Preliminaries. First, we provide a brief overview of the commonly used vanilla attention and linear attention. Vanilla attention is a core mechanism in transformer encoder layer, relying on three inputs: query Q, key K, and value V. The resultant output is a weighted sum of the value, where the weighted matrix is determined by the query and its corresponding key. Formally, the attention function is defined as follows:

$$\text{VanillaAttention}(Q, K, V) = \text{softmax}(QK^T)V \ . \quad (1)$$

However, applying the vanilla attention directly to dense local features is impractical due to the significant token size. To address this issue, previous methods use linear attention to reduce the computational complexity from quadratic to linear:

$$\text{LinearAttention}(Q, K, V) = \phi(Q)(\phi(K)^T\phi(V)) \ . \quad (2)$$

where $\phi(\cdot) = elu(\cdot) + 1$. However, it comes at the cost of reduced representational power, which is also observed by [5].

Aggragated Attention Module. After comprehensively investigating the mechanism of the transformer on coarse feature maps, we have two observations that motivate us to devise a new efficient aggregated attention. First, the attention regions of neighboring query tokens are similar, thus we can aggregate the neighboring tokens of f_i to prevent the redundant computation. Second, most of the attention weights of each query token are concentrated on a small number of key tokens, hence we can select the salient tokens of f_j before attention to reduce the computation.

---

[1] We feed feature of one image as query and feature of the other image as key and value into cross-attention, similar to SG [43] and LoFTR [50].

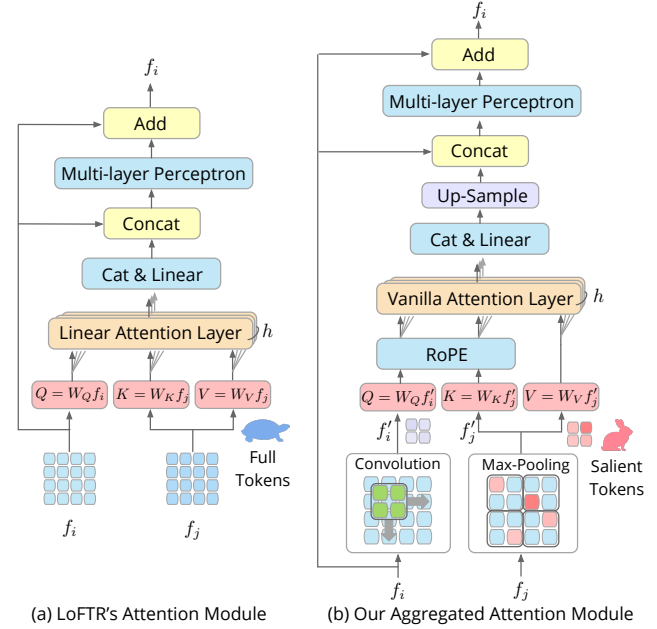(a) LoFTR's Attention Module (b) Our Aggregated Attention Module

Figure 3. **Detailed Transformer Module Comparison.** Unlike LoFTR which uses all tokens of feature maps to compute attention and resort to linear attention to reduce the computational cost, the proposed attention module first aggregates features for salient tokens, which is significantly more efficient for attention. Then the vanilla attention is utilized to transform aggregated features, where relative positional encoding is inserted to capture the spatial information. Transformed features are upsampled and fused with the original features to form the final features.

Therefore, we propose to first aggregate the f_i token utilizing a depth-wise convolution network, and f_j is aggregated by a max pooling layer to get reduced salient tokens:

$$f_i' = \text{Conv2D}(f_i), \ \ f_j' = \text{MaxPool}(f_j) \ , \quad (3)$$

where Conv2D is implemented by a strided depthwise convolution with a kernel size of $s \times s$, identical to that of the max-pooling layer. Then positional encoding and vanilla attention are followed to process reduced tokens. Positional encoding (PE) can help to model the spatial location contexts, where RoPE [49] is adopted in practice to account for more robust relative positions, inspired by [30]. Note that the PE layer is enabled exclusively for self-attention and skipped during cross-attention. The transformed feature map is then upsampled and fused with f_i for the final feature map. Due to the aggregation and selection, the number of tokens in f_i' and f_j' is reduced by s^2, which contributes to the efficiency of the attention phase.

3.3. Coarse-level Matching Module

We establish coarse-level matches based on the previously transformed coarse feature maps $\tilde{\mathbf{F}}_A^t$, $\tilde{\mathbf{F}}_B^t$. Coarse correspondences indicate rough match regions for later subpixel-level matching in the refinement phase. To achieve this, $\tilde{\mathbf{F}}_A^t$ and $\tilde{\mathbf{F}}_B^t$ are densely correlated to build a score matrix $\mathcal{S}$. The

softmax operator on both $\mathcal{S}$ dimensions (referred to as dual-softmax) is then applied to obtain the probability of mutual nearest matching, which is commonly used in [39, 50, 57]. The coarse correspondences $\{\mathcal{M}_c\}$ are established by selecting matches above the score threshold τ while satisfying the mutual-nearest-neighbor (MNN) constraint.

Efficient Inference Strategy. We observe that the dual-softmax operator in the coarse matching can significantly restrict the efficiency in inference due to the large token size, especially for high-resolution images. Moreover, we find that the dual-softmax operator is crucial for training, dropping it at inference time while directly using the score matrix $\mathcal{S}$ for MNN matching can also work well with better efficiency.

The reason for using the dual-softmax operator in training is that it can help to train discriminative features. Intuitively, with the softmax operation, the matching score between two pixels can also conditioned on other pixels. This mechanism forces the network to improve feature similarity of true correspondences while suppressing similarity with irrelevant points. With trained discriminative features, the softmax operation can be potentially eliminated during inference.

We denote the model skipping dual-softmax layer in inference as *efficiency optimized model*. Results in Tab. 1 demonstrate the effectiveness of this design.

3.4. Subpixel-Level Refinement Module

As overviewed in Fig. 2 (4), with established coarse matches $\{\mathcal{M}_c\}$, we refine them for sub-pixel accuracy with our refinement module. It is composed of an efficient feature patch extractor for discriminative fine features, followed by a two-stage feature correlation layer for final matches $\{\mathcal{M}_f\}$.

Efficient Fine Feature Extraction. We first extract discriminative fine feature patches centered at each coarse match $\mathcal{M}_c$ by an efficient fusion network for later match refinement. For efficiency, our key idea here is to re-leverage the previously transformed coarse features $\tilde{\mathbf{F}}_A^t$, $\tilde{\mathbf{F}}_B^t$ to obtain cross-view attended discriminative fine features, instead of introducing additional feature transform networks as in LoFTR [50].

To be specific, $\tilde{\mathbf{F}}_A^t$ and $\tilde{\mathbf{F}}_B^t$ are adaptively fused with $1/4$ and $1/2$ resolution backbone features by convolution and up-sampling to obtain fine feature maps $\hat{\mathbf{F}}_A^t$, $\hat{\mathbf{F}}_B^t$ in the original image resolution. Then local feature patches are cropped on fine feature maps centered at each coarse match. Since only shallow feed-forward networks are included, our fine feature fusion network is remarkably efficient.

Two-Stage Correlation for Refinement. Based on the extracted fine local feature patches of coarse matches, we

search for high-accurate sub-pixel matches. To refine a coarse match, a commonly used strategy [7, 18, 50] is to select the center-patch feature of $\mathbf{I}_A$ as a fixed reference point, and perform feature correlation and expectation on the entire corresponding feature patch for its fine match. However, this refinement-by-expectations will introduce location variance to the final match, because irrelevant regions also have weights and can affect results. Therefore, we propose a novel two-stage correlation module to solve this problem.

Our idea is to utilize a mutual-nearest-neighbor (MNN) matching to get intermediate pixel-level refined matches in the first stage, and then refine them for subpixel accuracy by correlation and expectation. Motivations are that MNN matching don't have spatial variance since matches are selected by directly indexing pixels with maximum scores, but cannot achieve sub-pixel accuracy. Conversely, refinement-by-expectation can achieve sub-pixel accuracy but variance exists. The proposed two-stage refinement can draw benefits by combining the best of both worlds.

Specifically, to refine a coarse-level correspondence $\mathcal{M}_c$, the first-stage refinement phase densely correlates their fine feature patches to obtain the local patch score matrix $\mathcal{S}_l$. MNN searching is then applied on $\mathcal{S}_l$ to get intermediate pixel-level fine matches. To limit the overall match number, we select the top-1 fine match for one coarse match by sorting the correlation scores.

Then, we further refine these pixel-level matches for sub-pixel accuracy by our second-stage refinement. Since the matching accuracy has already significantly improved in first-stage refinement, now we can use a tiny local window for correlation and expectation with a maximum suppression of location variance. In practice, we correlate the feature of each point in $\mathbf{I}_A$ with a 3×3 feature patch centered at its fine match in $\mathbf{I}_B$. The softmax operator is then applied to get a match distribution matrix and the final refined match is obtained by calculating expectations.

3.5. Supervision

The entire pipeline is trained end-to-end by supervising the coarse and refinement matching modules separately.

Coarse-Level Matching Supervision. The coarse ground truth matches $\{\mathcal{M}_c\}_{gt}$ with a total number of N are built by warping grid-level points from $\mathbf{I}_A$ to $\mathbf{I}_B$ via depth maps and image poses following previous methods [43, 50]. The produced correlation score matrix $\mathcal{S}$ in coarse matching is supervised by minimizing the log-likelihood loss over locations of $\{\mathcal{M}_c\}_{gt}$:

$$\mathcal{L}_c = -\frac{1}{N} \sum_{(\tilde{i},\tilde{j}) \in \{\mathcal{M}_c\}_{gt}} \log \mathcal{S}\left(\tilde{i}, \tilde{j}\right) \quad . \tag{4}$$

Fine-Level Matching Supervision. We train the proposed two-stage fine-level matching module by separately super-

Category	Method	MegaDepth Dataset			ScanNet Dataset			Time (ms)
		AUC@5°	AUC@10°	AUC@20°	AUC@5°	AUC@10°	AUC@20°	
Sparse	SP + NN	31.7	46.8	60.1	7.5	18.6	32.1	**10.8**
	SP + SG	49.7	**67.1**	**80.6**	**16.2**	**32.8**	**49.7**	48.3
	SP + LG	**49.9**	67.0	80.1	14.8	30.8	47.5	31.9/30.7
Semi-Dense	DRC-Net	27.0	42.9	58.3	7.7	17.9	30.5	328.0
	LoFTR	52.8	69.2	81.2	16.9	33.6	50.6	66.2
	QuadTree	54.6	70.5	82.2	19.0	37.3	53.5	100.7
	MatchFormer	53.3	69.7	81.8	15.8	32.0	48.0	128.9
	TopicFM	54.1	70.1	81.6	17.3	35.5	50.9	66.4
	AspanFormer	55.3	71.5	83.1	**19.6**	**37.7**	**54.4**	81.6
	Ours	**56.4**	**72.2**	**83.5**	19.2	37.0	53.6	40.1/34.4
	Ours (Optimized)	55.4	71.4	82.9	17.4	34.4	51.2	**35.6/27.0**
Dense	DKM	60.4	74.9	85.1	26.64	47.07	64.17	210.8
	ROMA	62.6	76.7	86.3	28.9	50.4	68.3	302.7

Table 1. **Results of Relative Pose Estimation on MegaDepth Dataset and ScanNet Dataset.** We use the models trained on the MegaDepth dataset to evaluate all methods on both datasets, which can show the intra- and inter-dataset generalization abilities. The AUC of pose error at different thresholds, along with the processing time for matching image pair at a resolution of 640×480, is presented. For SP + LG, Ours, and Ours (Optimized), the running times of the model using FP32/Mixed-Precision numerical precisions are shown.

vising the two phases. The first stage fine loss $\mathcal{L}_{f1}$ is to minimize the log-likelihood loss of each fine local score matrix $\mathcal{S}_l$ based on the pixel-level ground truth fine matches, similar to coarse loss. The second stage is trained by $\mathcal{L}_{f2}$ that calculates the ℓ_2 loss between the final subpixel matches $\{\mathcal{M}_f\}$ and ground truth fine matches $\{\mathcal{M}_f\}_{gt}$.

The total loss is the weighted sum of all supervisions:
$\mathcal{L} = \mathcal{L}_c + \alpha \mathcal{L}_{f1} + \beta \mathcal{L}_{f2}$.

4. Experiments

In this section, we evaluate the performance of our method on several downstream tasks, including homography estimation, pairwise pose estimation and visual localization. Furthermore, we evaluate the effectiveness of our design by conducting detailed ablation studies.

4.1. Implementation Details

We adopt RepVGG [11] as our feature backbone, and self- and cross-attention are interleaved for $N = 4$ times to transform coarse features. For each attention, we aggregate features by a depth-wise convolution layer and a max-pooling layer, both with a kernel size of 4×4. Our model is trained on the MegaDepth dataset [28], which is a large-scale outdoor dataset. The test scenes are separated from training data following [50]. The loss function's weights α and β are set to 1.0 and 0.25, respectively. We use the AdamW optimizer with an initial learning rate of 4×10^{-3}. The network training takes about 15 hours with a batch size of 16 on 8 NVIDIA V100 GPUs. And the coarse and fine stages are trained together from scratch. The trained model on MegaDepth is used to evaluate all datasets and tasks in our experiments to demonstrate the generalization ability.

4.2. Relative Pose Estimation

Datasets. We use the outdoor MegaDepth [28] dataset and indoor ScanNet [8] dataset for the evaluation of relative pose estimation to demonstrate the efficacy of our method.

MegaDepth dataset is a large-scale dataset containing sparse 3D reconstructions from 196 scenes. The key challenges on this dataset are large viewpoints and illumination changes, as well as repetitive patterns. We follow the test split of the previous method [50] that uses 1500 sampled pairs from scenes "Sacre Coeur" and "St. Peter's Square" for evaluation. Images are resized so that the longest edge equals 1200 for all semi-dense and dense methods. Following [30], sparse methods are provided resized images with longest edge equals 1600.

ScanNet dataset contains 1613 sequences with ground-truth depth maps and camera poses. They depict indoor scenes with viewpoint changes and texture-less regions. We use the sampled test pairs from [43] for the evaluation, where images are resized to 640×480 for all methods.

Baselines. We compare the proposed method with three categories of methods: 1) sparse keypoint detection and matching methods, including SuperPoint [10] with Nearest-Neighbor (NN), SuperGlue (SG) [43], LightGlue (LG) [30] matchers, 2) semi-dense matchers, including DRC-Net [27], LoFTR [50], QuadTree Attention [52], MatchFormer [59], AspanFormer [7], TopicFM [18], and 3) state-of-the-art dense matcher ROMA [15] that predict matches for each pixel.

Evaluation protocol. Following previous methods, the recovered relative poses by matches are evaluated for reflecting matching accuracy. The pose error is defined as the maximum of angular errors in rotation and translation. We report the AUC of the pose error at thresholds ($5°$, $10°$, and $20°$). Moreover, the running time of matching each image pair in the ScanNet dataset is reported for comprehensively understanding the matching accuracy and efficiency balance. We use a single NVIDIA 3090 to evaluate the running time of all methods.

Category	Method	Homography est. AUC		
		@3px	@5px	@10px
Sparse	D2Net + NN	23.2	35.9	53.6
	R2D2 + NN	50.6	63.9	76.8
	DISK + NN	52.3	64.9	78.9
	SP + SG	**53.9**	**68.3**	**81.7**
Semi-Dense	Sparse-NCNet	48.9	54.2	67.1
	DRC-Net	50.6	56.2	68.3
	LoFTR	65.9	75.6	84.6
	Ours	**66.5**	**76.4**	**85.5**

Table 2. **Results of Homography Estimation on HPatches Dataset.** Our method is compared with sparse and semi-dense methods. The AUC of reprojection error of corner points at different thresholds is reported.

Results. As shown in Tab. 1, the proposed method achieves competitive performances compared with sparse and semi-dense methods on both datasets. Qualitative comparisons are shown in Fig. 4. Specifically, our method outperforms the best semi-dense baseline AspanFormer on all metrics of the MegaDepth dataset and has lower but comparable performance on the ScanNet dataset, with ~ 2 times faster. Our optimized model that eliminates the dual-softmax operator in coarse-level matching further brings efficiency improvements, with slight performance decreases. Using this strategy, our method can outperform the efficient and robust sparse method SP + LG in efficiency with significantly higher accuracy. Dense matcher ROMA shows remarkable matching capability but is slow for applications in practice. Moreover, since ROMA utilizes the pre-trained DINOv2 [37] backbone, its strong generalizability on ScanNet may be attributed to the similar indoor training data in DINOv2, where other methods are trained on outdoor MegaDepth only. Compared with it, our method is ~ 7.5 times faster, which has a good balance between accuracy and efficiency.

4.3. Homography Estimation

Dataset. We evaluate our method on HPatches dataset [3]. HPatches dataset depicts planar scenes divided into sequences. Images are taken under different viewpoints or illumination changes.

Baselines. We compare our method with sparse methods including D2Net [12], R2D2 [38], DISK [57] detectors with NN matcher, and SuperPoint [10] + SuperGlue [43]. As for semi-dense methods, we compare with Sparse-NCNet [40], DRC-Net [27], and LoFTR [50]. For SuperGlue and all semi-dense methods, we use their models trained on MegaDepth dataset for evaluation.

Evaluation Protocol. Following SuperGlue [43] and LoFTR [50], we resize all images for matching so that their smallest edge equals 480 pixels. We collect the mean reprojection error of corner points, and report the area under the cumulative curve (AUC) under 3 different thresholds, including 3 px, 5 px, and 10 px. For all baselines, we employ the

Method	DUC1	DUC2
	(0.25m,2°)/(0.5m,5°)/(1.0m,10°)	
SP+SG	49.0 / 68.7 / 80.8	53.4 / 77.1 / 82.4
LoFTR	47.5 / 72.2 / 84.8	54.2 / 74.8 / 85.5
TopicFM	52.0 / **74.7** / **87.4**	53.4 / 74.8 / 83.2
PATS	**55.6** / 71.2 / 81.0	**58.8** / **80.9** / 85.5
AspanFormer	51.5 / 73.7 / 86.0	55.0 / 74.0 / 81.7
Ours	52.0 / **74.7** / 86.9	58.0 / **80.9** / **89.3**

Table 3. **Results of Visual Localization on InLoc Dataset.**

Method	Day	Night
	(0.25m,2°)/(0.5m,5°)/(1.0m,10°)	
SP+SG	89.8 / 96.1 / **99.4**	77.0 / 90.6 / **100.0**
LoFTR	88.7 / 95.6 / 99.0	**78.5** / 90.6 / 99.0
TopicFM	**90.2** / 95.9 / 98.9	77.5 / 91.1 / 99.5
PATS	89.6 / 95.8 / 99.3	73.8 / **92.1** / 99.5
AspanFormer	89.4 / 95.6 / 99.0	77.5 / 91.6 / 99.5
Ours	89.6 / **96.2** / 99.0	77.0 / 91.1 / 99.5

Table 4. **Results of Visual Localization on Aachen v1.1 Dataset.**

same RANSAC method as a robust homography estimator for a fair comparison. Following LoFTR, we select only the top 1000 predicted matches of semi-dense methods for the sake of fairness.

Results. As shown in Tab. 2, even though the number of matches is restricted, our method can also work remarkably well and outperform sparse methods significantly. Compared with semi-dense, our method can surpass them with significantly higher efficiency. We attribute this to the effectiveness of two-stage refinement for accuracy improvement and proposed aggregation module for efficiency.

4.4. Visual Localization

Datasets and Evaluation Protocols. Visual localization is an important downstream task of image matching, which aims to estimate the 6-DoF poses of query images based on the 3D scene model. We conduct experiments on two commonly used benchmarks, including InLoc [51] dataset and Aachen v1.1 [45] dataset, for evaluation to demonstrate the superiority of our method. InLoc dataset is captured on indoor scenes with plenty of repetitive structures and textureless regions, where each database image has a corresponding depth map. Aachen v1.1 is a challenging large-scale outdoor dataset for localization with large-viewpoint and day-and-night illumination changes, which particularly relies on the robustness of matching methods. We adopt its full localization track for benchmarking.

Following [7, 50], the open-sourced localization framework HLoc [42] is utilized. For both datasets, the percentage of pose errors satisfying both angular and distance thresholds is reported following the benchmarks, where different thresholds are used. For the InLoc dataset, the metrics of two test scenes including DUC1 and DUC2 are separately reported. As for the Aachen v1.1 dataset, the metrics corresponding to the daytime and nighttime divisions are reported.
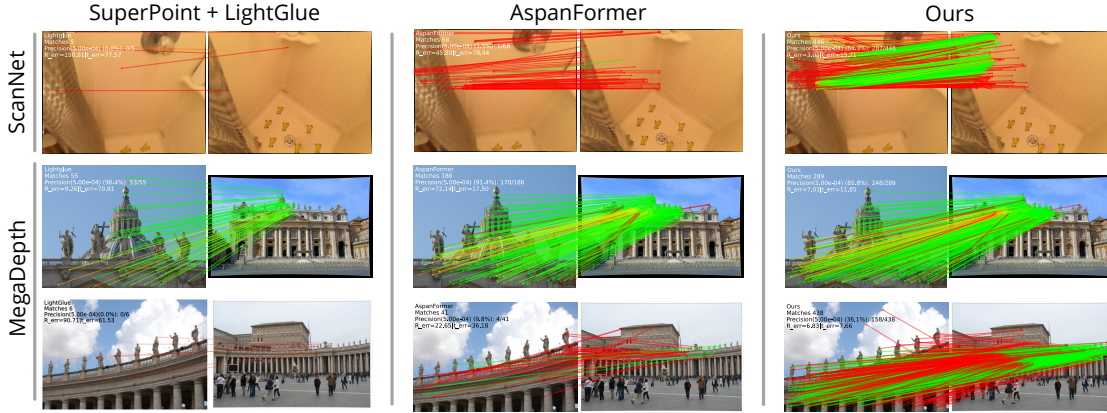
Figure 4. **Qualitative Results.** Our method is compared with the sparse matching pipeline SuperPoint [10]+LightGlue [30], semi-dense matcher AspanFormer [7]. Image pairs with texture-poor regions and large-viewpoint changes can be robustly matched by our method. The red color indicates epipolar error beyond 5×10^{-4} (in the normalized image coordinates).

Method	Pose Estimation AUC			Time (ms)
	@5°	@10°	@20°	
Ours Full	**56.4**	**72.2**	**83.5**	139.2
1) Ours Optimal (w/o dual-softmax)	55.4	71.4	82.9	**102.0**
2) Replace Agg. Attention to LoFTR's Trans.	54.7	70.5	82.2	171.4
3) Replace two-stage refine. to LoFTR's refine.	54.7	70.9	82.7	135.3
4) No second-stage refinement	55.8	71.8	83.3	138.1
5) Replace RepVGG with ResNet	55.4	71.4	82.9	156.2

Table 5. **Ablation Studies.** The components of our method are ablated on the MegaDepth dataset for a comprehensive understanding of our method, where averaged running times for an image pair with high-resolution 1200×1200 are reported.

Baselines. We compare the proposed method with both detector-based method SuperPoint [10]+SuperGlue [43] and detector-free methods including LoFTR [50], TopicFM [18], PATS [36] and Aspanformer [7].

Results. We adhere to the pipeline and evaluation settings of the online visual localization benchmark[2] to ensure fairness. As presented in Tab. 3, the proposed method achieves competitive results, taking both detector-based and detector-free methods into account. Being a method primarily geared towards efficiency, our approach can deliver results comparable to those of many accuracy-oriented methods. As depicted in Tab. 4, our method also demonstrates performance on par with the best-performing approaches.

4.5. Ablation Studies

In this part, we conduct detailed ablation studies to analyze the effectiveness of our proposed modules with results shown in Tab. 5. 1) Without dual-softmax, our optimal model can bring huge efficiency improvement in high-resolution images. 2) In coarse feature transformation, replacing the proposed aggregated attention module with LoFTR's transformer can bring significant efficiency dropping, as well as accuracy decrease. Note that the replaced transformer

is also equipped with RoPE same as ours for fair comparison. This demonstrates the efficacy of the proposed module that performing vanilla attention on aggregated features can achieve higher efficiency with even better matching accuracy. 3) Compared with using LoFTR's refinement that performs expectation on the entire correlation patch, the proposed two-stage refinement layer can bring accuracy improvement with neglectable latency. We attribute this to the two-stage refinement's property that can maximize the suppression of location variance in correlation refinement. 4) Dropping the second refinement stage will lead to degraded pose accuracy with minor efficiency changes, especially on the strict AUC@5° metric. 5) Changing the backbone from reparameterized VGG [11] back to multi-branch ResNet [20] leads to decreased efficiency with similar accuracy, which demonstrates the effectiveness of our design choice for efficiency.

5. Conclusions

This paper introduces a new semi-dense local feature matcher based on the success of LoFTR. We revisit its designs and propose several improvements for both efficiency and matching accuracy. A key observation is that performing the Transformer on the entire coarse feature map is redundant due to the similar local information, where an aggregated attention module is proposed to perform transformer on reduced tokens with significantly better efficiency and competitive performance. Moreover, a two-stage correlation layer is devised to solve the location variance problem in LoFTR's refinement design, which further brings accuracy improvements. As a result, our method can achieve ~ 2.5 times faster compared with LoFTR with better matching accuracy. Moreover, as a semi-dense matching method, the proposed method can achieve comparable efficiency with the recent robust sparse feature matcher LightGlue [30]. We believe this opens up the applications of our method in large-scale or

latency-sensitive downstream tasks, such as image retrieval and 3D reconstruction. Please refer to the supplementary material for discussions about limitations and future works.

References

[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M. Seitz, and Richard Szeliski. Building rome in a day. *ICCV*, 2009. 1

[2] Relja Arandjelović, Petr Gronát, Akihiko Torii, Tomás Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. *CVPR*, pages 5297–5307, 2016. 14

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

[4] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *CVIU*, 110 (3):346–359, 2008. 2

[5] Han Cai, Chuang Gan, and Song Han. Efficientvit: Enhanced linear attention for high-resolution low-computation visual recognition. *arXiv preprint arXiv:2205.14756*, 2022. 4

[6] Hongkai Chen, Zixin Luo, Jiahui Zhang, Lei Zhou, Xuyang Bai, Zeyu Hu, Chiew-Lan Tai, and Long Quan. Learning to match features with seeded graph matching network. In *ICCV*, pages 6301–6310, 2021. 2

[7] Hongkai Chen, Zixin Luo, Lei Zhou, Yurun Tian, Mingmin Zhen, Tian Fang, David Mckinnon, Yanghai Tsin, and Long Quan. Aspanformer: Detector-free image matching with adaptive span transformer. In *ECCV*, pages 20–36. Springer, 2022. 2, 5, 6, 7, 8, 12, 13, 14

[8] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, pages 5828–5839, 2017. 6

[9] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *NeurIPS*, 2022. 3

[10] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. *CVPRW*, 2018. 2, 6, 7, 8, 13

[11] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *CVPR*, pages 13733–13742, 2021. 3, 6, 8, 12

[12] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint description and detection of local features. In *CVPR*, pages 8092–8101, 2019. 2, 7

[13] Patrick Ebel, Anastasiia Mishchuk, Kwang Moo Yi, Pascal Fua, and Eduard Trulls. Beyond cartesian representations for local descriptors. In *ICCV*, pages 253–262, 2019. 2

[14] Johan Edstedt, Ioannis Athanasiadis, Mårten Wadenbäck, and Michael Felsberg. Dkm: Dense kernelized feature matching for geometry estimation. In *CVPR*, pages 17765–17775, 2023. 2, 3, 14

[15] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. Roma: Revisiting robust losses for dense feature matching. *arXiv preprint arXiv:2305.15404*, 2023. 2, 3, 6, 14

[16] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. *ICCV*, pages 6804–6815, 2021. 12

[17] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24:381–395, 1981. 14

[18] Khang Truong Giang, Soohwan Song, and Sung-Guk Jo. Topicfm: Robust and interpretable topic-assisted feature matching. In *AAAI*, 2023. 2, 3, 5, 6, 8, 12, 14

[19] Stephen Hausler, Sourav Garg, Ming Xu, Michael Milford, and Tobias Fischer. Patch-netvlad: Multi-scale fusion of locally-global descriptors for place recognition. *CVPR*, pages 14136–14147, 2021. 2

[20] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CVPR*, pages 770–778, 2015. 3, 8

[21] Xingyi He, Jiaming Sun, Yifan Wang, Sida Peng, Qixing Huang, Hujun Bao, and Xiaowei Zhou. Detector-free structure from motion. In *CVPR*, 2024. 1

[22] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and Franccois Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *ICML*, 2020. 3

[23] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *ArXiv*, abs/2001.04451, 2020. 3

[24] Axel Barroso Laguna, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Key.net: Keypoint detection by handcrafted and learned cnn filters. *ICCV*, pages 5835–5843, 2019. 2

[25] Viktor Larsson and contributors. PoseLib - Minimal Solvers for Camera Pose Estimation, 2020. 14

[26] Benjamin Lefaudeux, Francisco Massa, Diana Liskovich, Wenhan Xiong, Vittorio Caggiano, Sean Naren, Min Xu, Jieru Hu, Marta Tintore, Susan Zhang, Patrick Labatut, and Daniel Haziza. xformers: A modular and hackable transformer modelling library. https://github.com/facebookresearch/xformers, 2022. 3

[27] Xinghui Li, Kai Han, Shuda Li, and Victor Prisacariu. Dual-resolution correspondence networks. In *NeurIPS*, 2020. 2, 6, 7

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

[29] Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-perfect structure-from-motion with featuremetric refinement. *ICCV*, pages 5967–5977, 2021. 1

[30] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local Feature Matching at Light Speed. In *ICCV*, 2023. 1, 2, 4, 6, 8, 13, 15

[31] G LoweDavid. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 1, 2

[32] Zixin Luo, Lei Zhou, Xuyang Bai, Hongkai Chen, Jiahui Zhang, Yao Yao, Shiwei Li, Tian Fang, and Long Quan. Aslfeat: Learning local features of accurate shape and localization. In *CVPR*, pages 6589–6598, 2020. 2

[33] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. *NeurIPS*, 30, 2017. 2

[34] Raul Mur-Artal and Juan D. Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *TR*, 33:1255–1262, 2016. 1

[35] Raul Mur-Artal, José M. M. Montiel, and Juan D. Tardós. Orb-slam: A versatile and accurate monocular slam system. *TR*, 31:1147–1163, 2015. 1

[36] Junjie Ni, Yijin Li, Zhaoyang Huang, Hongsheng Li, Hujun Bao, Zhaopeng Cui, and Guofeng Zhang. Pats: Patch area transportation with subdivision for local feature matching. *CVPR*, pages 17776–17786, 2023. 2, 8, 12

[37] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 7

[38] Jérôme Revaud, César Roberto de Souza, M. Humenberger, and Philippe Weinzaepfel. R2d2: Reliable and repeatable detector and descriptor. In *NeurIPS*, 2019. 2, 7

[39] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Neighbourhood consensus networks. *NeurIPS*, 31, 2018. 2, 5

[40] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. Efficient neighbourhood consensus networks via submanifold sparse convolutions. In *ECCV*, pages 605–621. Springer, 2020. 2, 7

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

[42] Paul-Edouard Sarlin, César Cadena, Roland Y. Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. *CVPR*, pages 12708–12717, 2019. 1, 7, 14

[43] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 1, 2, 4, 5, 6, 7, 8, 14

[44] Paul-Edouard Sarlin, Ajaykumar Unagar, Måns Larsson, Hugo Germain, Carl Toft, Victor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, and Torsten Sattler. Back to the Feature: Learning Robust Camera Localization from Pixels to Pose. In *CVPR*, 2021. 1

[45] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. Benchmarking 6dof outdoor visual localization in changing conditions. In *CVPR*, pages 8601–8610, 2018. 7

[46] Nikolay Savinov, Akihito Seki, Lubor Ladicky, Torsten Sattler, and Marc Pollefeys. Quad-networks: unsupervised learning to rank for interest point detection. In *CVPR*, pages 1822–1830, 2017. 2

[47] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1, 2

[48] Yan Shi, Jun-Xiong Cai, Yoli Shavit, Tai-Jiang Mu, Wensen Feng, and Kai Zhang. Clustergnn: Cluster-based coarse-to-fine graph neural network for efficient feature matching. In *CVPR*, pages 12517–12526, 2022. 2

[49] Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *ArXiv*, abs/2104.09864, 2021. 4, 12

[50] 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, 2, 4, 5, 6, 7, 8, 12, 14

[51] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. Inloc: Indoor visual localization with dense matching and view synthesis. In *CVPR*, pages 7199–7209, 2018. 7

[52] Shitao Tang, Jiahui Zhang, Siyu Zhu, and Ping Tan. Quadtree attention for vision transformers. *ICLR*, 2022. 2, 6, 12, 14

[53] Yurun Tian, Bin Fan, and Fuchao Wu. L2-net: Deep learning of discriminative patch descriptor in euclidean space. In *CVPR*, pages 661–669, 2017. 1, 2

[54] Yurun Tian, Xin Yu, Bin Fan, Fuchao Wu, Huub Heijnen, and Vassileios Balntas. Sosnet: Second order similarity regularization for local descriptor learning. In *CVPR*, pages 11016–11025, 2019. 2

[55] Yurun Tian, Vassileios Balntas, Tony Ng, Axel Barroso-Laguna, Yiannis Demiris, and Krystian Mikolajczyk. D2d: Keypoint extraction with describe to detect approach. In *ACCV*, 2020. 2

[56] Prune Truong, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning accurate dense correspondences and when to trust them. In *CVPR*, pages 5714–5724, 2021. 3

[57] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. Disk: Learning local features with policy gradient. *NeurIPS*, 2020. 5, 7

[58] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 3, 12

[59] Qing Wang, Jiaming Zhang, Kailun Yang, Kunyu Peng, and Rainer Stiefelhagen. Matchformer: Interleaving attention in transformers for feature matching. In *ACCV*, 2022. 2, 6, 12, 14

[60] Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *ArXiv*, abs/2006.04768, 2020. 3

[61] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. *CVPR*, pages 10809–10819, 2022. 12

Supplementary Material

A. Insight and Discussion about Aggregated Attention Module

Some previous works explored using pooling in ViT but are with different design choices from our method due to different tasks. PoolFormer [61] replaces the multi-head attention with pooling, which cannot be used for cross-attention in matching that two images are not pixel-aligned. MVit [16] uses pooling to reduce tokens like ours, but they cannot get high-res features that are required for matching.

Differently, we propose to first conduct attention on aggregated features and then *upsample* before feed-forward network (FFN) for later fusion with input feature, as shown in Fig. 3. This aggregate-and-upsample block can minimize information loss in aggregation and efficiently get high-res informative features, where conducting upsampling before fusion is crucial to fuse smoothly interpolated messages with a detailed feature map. Ablation is in Tab. 10 (8).

Moreover, We perform Conv on Q value instead of pooling because salient tokens should *not* represent neighbors to query attention. The transformer is crucial for enhancing non-salient features for matching. Pooling on Q causes the attention of texture-less areas dominated by neighboring salient tokens, reducing the performance as ablated in Tab. 10 (6,7).

B. Implementation Details

B.1. Local Feature Extraction

RepVGG [11] blocks are used to build a four-stage feature backbone. We use a width of 64 and a stride of 1 for the first stage and widths of [64, 128, 256] and strides of 2 for the subsequent three stages. Each stage is composed of [1, 2, 4, 14] RepVGG blocks and ReLU activations, respectively. The output of the last stage in ⅛ image resolution is used for efficient local feature transformer modules to get attended coarse feature maps. The second and third stages' feature maps are in ½ and ¼ image resolutions, respectively, which are used for fusing with transformed coarse features for fine features.

B.2. Position Encoding

We use the 2D extension of Rotary position encoding [49] to encode the relative position between coarse features in self-attention modules. Given the projected features q and k, the attention score between two features q_i and k_j is computed as:

$$a_{ij} = q_i^T R(x_j - x_i, y_j - y_i)k_j \ , \tag{5}$$

Position Encoding	Pose Estimation AUC			Time (ms)
	@5°	@10°	@20°	
RoPE	**56.4**	**72.2**	**83.5**	139.2
sinusoidal	55.5	71.5	83.1	**137.5**

Table 6. Impact of position encoding on the MegaDepth dataset, where averaged running times for an image pair with high-resolution 1200×1200 are reported.

where x_i, y_i, x_j, y_j are the coordinates of q_i and k_j, R is a block diagonal matrix:

$$R(\Delta x, \Delta y) = \begin{pmatrix} R_1(\Delta x, \Delta y) & & & \\ & R_2(\Delta x, \Delta y) & & \\ & & \ddots & \\ & & & R_{d/4}(\Delta x, \Delta y) \end{pmatrix} \ , \tag{6}$$

$$R_k(\Delta x, \Delta y) = \begin{pmatrix} \cos(\theta_k \Delta x) & -\sin(\theta_k \Delta x) & 0 & 0 \\ \sin(\theta_k \Delta x) & \cos(\theta_k \Delta x) & 0 & 0 \\ 0 & 0 & \cos(\theta_k \Delta y) & -\sin(\theta_k \Delta y) \\ 0 & 0 & \sin(\theta_k \Delta y) & \cos(\theta_k \Delta y) \end{pmatrix} \ , \tag{7}$$

where $\theta_k = \frac{1}{10000^{4k/d}}, \ k \in [1, 2, ..., d/4]$ encode the index of feature channels.

Compared to the absolute position encoding used in previous methods [7, 18, 36, 50, 52, 59], we utilize 2D RoPE to allow the model to focus more on the interaction between features rather than their specific locations, which benefits capturing the context of local features. Moreover, relative position encoding is more robust to transformations like rotation, translation, and scaling, which is important for matching local features in different views.

C. More Experiments Results

C.1. More Ablation Studies

In this part, we conduct more ablation studies on the MegaDepth and ScanNet dataset to validate the design choices of our proposed modules.

Position Encoding We compare the performance of our 2D RoPE with the sinusoidal position encoding [58] in Tab. 6. The results show that using 2D RoPE can achieve better performance than sinusoidal position encoding.

Aggregation Range We show the performance of our method with different aggregation range s in Tab. 7. In our aggregated attention module, we use a 4×4 aggregation range to reduce token size before performing attention. Using a smaller aggregation range leads to more tokens, with slight performance changes but significantly slower matching speed. This validates the effectiveness of our parameter choice in the aggregation attention module.

Image Resolution We test the performance of our method with different image resolutions to show the performance

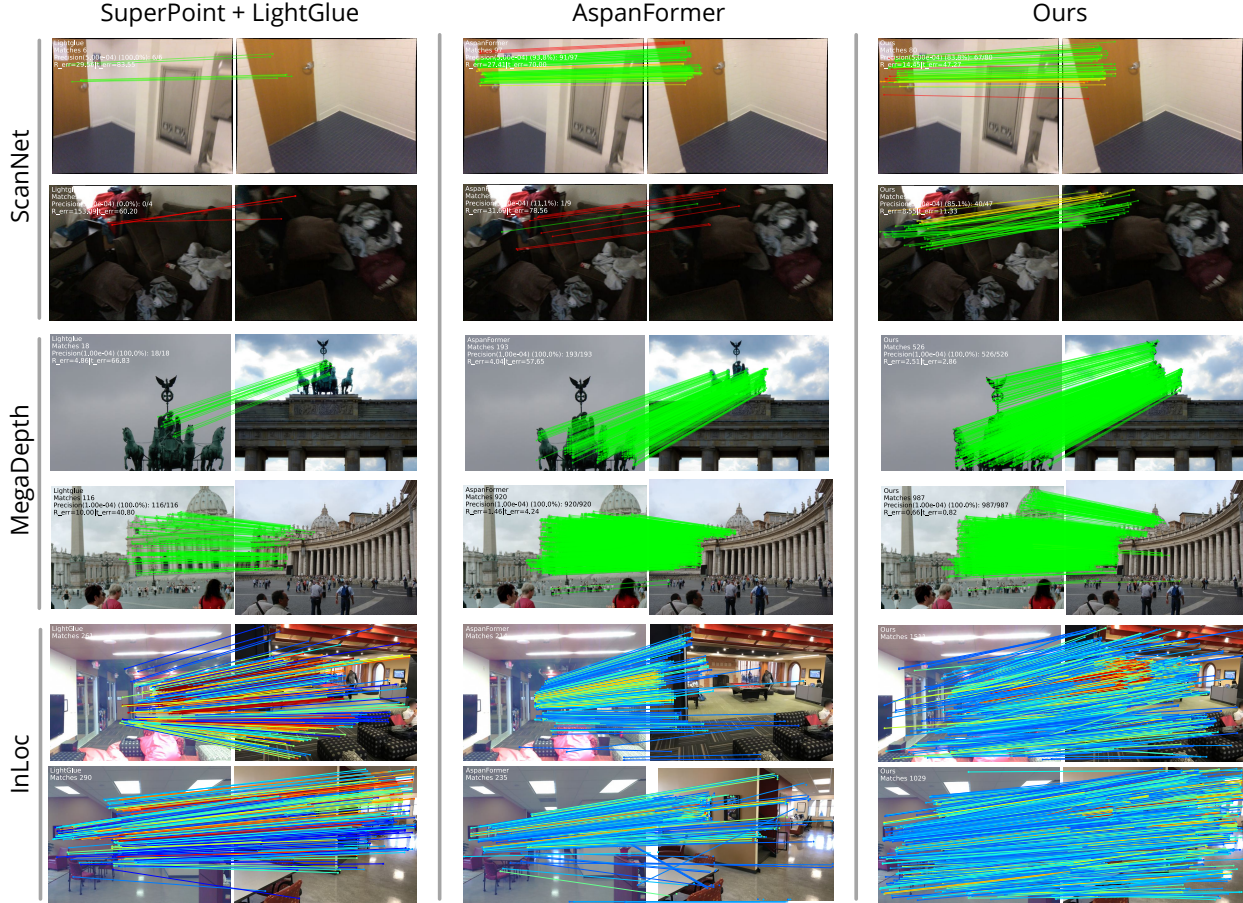| | SuperPoint + LightGlue | AspanFormer | Ours |

Figure 5. **Qualitative Results.** Our method is compared with the sparse matching pipeline SuperPoint [10]+LightGlue [30], semi-dense matcher AspanFormer [7]. The red color indicates epipolar error beyond 5×10^{-4} on ScanNet and 1×10^{-4} on MegaDepth (in the normalized image coordinates). Since no ground-truth pose is available on InLoc dataset, we color the match with predicted confidence. Red indicates higher confidence and blue for the opposite.

aggregation range	Pose Estimation AUC			Time (ms)
	@5°	@10°	@20°	
$s = 4$	**56.4**	**72.2**	83.5	**139.2**
$s = 2$	56.2	**72.2**	**83.6**	271.1

Table 7. Impact of aggregation range on the MegaDepth dataset, where averaged running times for an image pair with high-resolution 1200×1200 are reported.

and efficiency changes. Results are shown in Tab. 8. Compared with the default resolution 1184×1184 used in the MegaDepth evaluation, using a larger image size leads to noticeable accuracy improvement with a slower matching speed. Our method can still achieve competitive performance using low-resolution 640×640 images with the fastest speed. Therefore, our method is pretty robust in image resolution choices for flexible real-world applications.

Linear Attention After Aggregation Using linear attention in our aggregated attention module introduces minor

Resolution	Pose Estimation AUC			Time (ms)
	@5°	@10°	@20°	
640×640	51.0	67.4	79.8	**41.7**
800×800	53.4	70.0	81.9	58.2
960×960	54.7	70.7	82.4	81.8
1184×1184	**56.4**	72.2	**83.5**	139.2
1408×1408	56.2	**73.1**	83.4	223.9

Table 8. Impact of test image resolution on the MegaDepth dataset.

Method	MegaDepth Dataset			Time (ms)	ScanNet Dataset			Time (ms)
	AUC@5°	AUC@10°	AUC@20°		AUC@5°	AUC@10°	AUC@20°	
Full	**56.4**	**72.2**	**83.5**	139.1	**19.2**	**37.0**	**53.6**	**34.4**
Linear	54.1	70.3	82.1	**132.7**	16.8	33.2	49.0	36.9

Table 9. Impact of linear attention after aggregation on the MegaDepth and ScanNet dataset, where the resolution are 1200×1200 and 640×480, respectively.

efficiency gain on high-resolution but with accuracy dropping as shown in Tab. 9.

Method	Pose Estimation AUC			Time (ms)
	@5°	@10°	@20°	
Ours Full	**19.2**	**37.0**	**53.6**	34.4
1) Ours Optimal (w/o dual-softmax)	17.4	34.4	51.2	**27.0**
2) Replace Agg. Attention to LoFTR's Trans.	17.1	33.2	49.4	41.3
3) Replace two-stage refine. to LoFTR's refine.	18.1	35.8	52.4	31.8
4) No second-stage refinement	18.8	36.7	53.4	32.2
5) Replace RepVGG with ResNet	18.6	36.3	52.8	38.1
6) Both Conv in Agg. Attention	18.6	35.8	52.5	34.4
7) Both Pool in Agg. Attention	18.3	35.2	51.7	34.1
8) Upsample after FFN	17.3	34.6	51.4	32.6

Table 10. The components of our method are ablated on the Scan-Net dataset again for a comprehensive understanding of our method, where averaged running times for an image pair with resolution 640×480 are reported.

Method	RANSAC			LO-RANSAC		
	AUC@5°	AUC@10°	AUC@20°	AUC@5°	AUC@10°	AUC@20°
LightGlue	49.9	67.0	80.1	66.8	79.3	87.9
AspanFormer	58.3	73.3	**84.2**	69.4	**81.1**	**88.9**
Ours	**58.4**	**73.4**	**84.2**	**69.5**	80.9	88.8

Table 11. Results of Relative Pose Estimation on MegaDepth Dataset following LightGlue's setting. The AUC of pose error at different thresholds is presented.

Additional ablation studies on the ScanNet dataset We further repeat the ablation studies in the main paper and conduct additional ablation studies on the ScanNet dataset to validate the design choices of our proposed modules. Results are shown in Tab. 10.

C.2. More Qualitative Results

More qualitative results on the ScanNet dataset, InLoc dataset, and MegaDepth dataset are shown in Fig. 5.

C.3. Additional Results on other RANSAC setting

LightGlue uses a RANSAC setting different from other baseline papers [7, 14, 15, 18, 43, 50, 52, 59] in relative pose estimation evaluations on MegaDepth. We further conduct experiments following LightGlue's setting (OpenCV RANSAC [17] and LO-RANSAC [25] with carefully tuned RANSAC inlier thresholds), as shown in Tab. 11. Using the naive RANSAC method, the performance gap between ours and LightGlue becomes larger after RANSAC threshold tuning (compared with our untuned results in Tab. 1). This demonstrates that our method can achieve significantly better accuracy without depending on the sophisticated modern RANSAC method, thereby revealing its superior match quality. Using the stronger outlier filter LO-RANSAC, the accuracy of all methods is boosted and our method consistently achieves better performance than LightGlue, especially on the AUC5° metric.

D. Details About Timing

The running times evaluated in the paper are averaged over all pairs in the test dataset with a warm-up of 50 pairs for

Process	Time(ms)	
	Full	Optimized
Total	40.1	27.0
Feature Backbone	9.1	5.8
Coarse Feature Transformation	11.7	12.9
Coarse Matching	8.3	1.7
Fine Feature Fusion	8.0	4.8
Two-Stage Refinement	3.0	2.0

Table 12. Time cost for an image pair of 640×480 on the ScanNet dataset. The optimized model uses Mixed-Precision numerical accuracy and drops the dual-softmax operator in the coarse matching phase.

Method	Time(ms)		#Matches
	Matching	RANSAC	
SP + SG	43.6	0.53	487
DRC-Net	143.9	2.78	1019
LoFTR	76.2	1.59	995
Ours	45.9/38.6	1.39	997

Table 13. Running times of different methods on HPatches dataset. All images are resized so that their short edge equals 480 pixels following SuperGlue [43] and LoFTR [50]. For Ours, the running times of the model using FP32/Mixed-Precision numerical precisions are shown.

Method	Time(ms)	
	Aachen	InLoc
SP+SG	55.9	83.3
LoFTR	83.2	147.6
TopicFM	66.0	119.6
PATS	315.8	1148.0
AspanFormer	95.4	164.5
Ours	**40.6/25.5**	**82.0/46.3**

Table 14. Running times of different methods on Aachen and InLoc dataset. To measure the running time, we sample 818 and 356 pairs of images from the NetVLAD [2]'s retrieval results for Aachen and InLoc, respectively. All images are resized so that their long edge equals 1024 pixels following HLoc [42]. For Ours, the running times of the model using FP32/Mixed-Precision numerical precisions are shown.

accurate measurement. All the methods are tested on a single NVIDIA RTX 3090 GPU with 14 cores of Intel Xeon Gold 6330 CPU.

We further report each part running time of our method in Tab. 12, where both full and optimized models are shown. We noticed that skipping the dual-softmax of Coarse Matching in the optimized model can significantly reduce the running time. What's more, with the benefit of Mixed-Precision, the running time of feature extraction can be further reduced.

The latency on 3 datasets included in Tabs. 2 to 4 are shown in Tab. 13 and Tab. 14, where conclusion and speed rankings are the same as indicated in Tab. 1. We further

show RANSAC time on HPatches dataset. Both matching and RANSAC latency of our method are smaller than LoFTR with a similar number of matches.

E. Limitations and Future Works

We find that our method may fail when strong repetitive structures exist, such as matching an image pair that depicts different scenes containing the same chair. We think this may be due to the current model focusing more on local features for accurate matching, where global semantic context is lacking. Therefore, the mechanism of high-level contexts can be added to the model for performance improvement on ambiguous scenes. Moreover, we believe the efficiency of our method can be further improved by adopting the early stop strategy of LightGlue [30] since the contribution of the proposed efficient aggregation module is orthogonal to it.