2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2025)
October 19-25, 2025. Hangzhou, China

2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 979-8-3315-4393-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/IROS60139.2025.11247655

CA^{2}Point: Learning Keypoint Detection and Description with Context Aggregation and Cross Augmentation

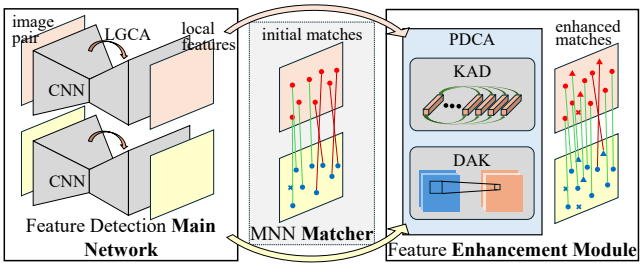
Xuebin Meng[1,2], Wei Li[1,3*], Yu Hu[1,3], and Yinhe Han[1,3]

***Abstract*— Keypoint detection and description are fundamental tasks for a variety of computer vision applications. Due to the limited receptive field of convolutional neural networks, most existing methods based on deep learning mainly focus on the local features, instead of taking into account the global context from entire image. The purpose of this work is to enhance the detection and description process of keypoints by leveraging global information obtained from Transformer, and to boost the consistence between keypoints and descriptors through their interaction. Specifically, the above two improvements are respectively implemented through the Local & Global Context Aggregation (LGCA) Module and Point & Descriptor Cross Augmentation (PDCA) Module proposed in this article. The LGCA module, which can model the long-range context, is inserted a Feature Pyramid Network (FPN) to extract features which contain diverse scales and different receptive fields. Moreover, the PDCA module enhances descriptors by the geometry information of keypoints detected, while enhancing the keypoint detection process by the position coordinates of correctly matched descriptors. Finally, we design a lightweight model to improve the running efficiency. Extensive experiments on various tasks demonstrate that our method achieves a substantial performance improvement over the current feature extraction methods. Code is available at: `https://github.com/meng152634/CA2Point`.**

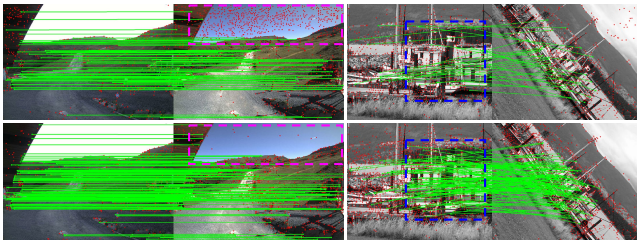
I. INTRODUCTION

Extracting and matching distinctive features from a given image pair is a key task for various computer visual applications, such as image matching [1], image stitching [2], Structure from Motion (SfM) [3] and Simultaneous Localization and Mapping (SLAM) [4]–[8]. Traditional handcrafted features [9]–[11] have been widely used in these tasks. However, handcrafted methods typically rely on expert prior knowledge, which made them fail to handle complex scenes such as illumination and viewpoint changes.

On the one hand, benefiting from the success of the Convolutional Neural Network (CNN), learning-based features [12]–[16] have been proposed and have achieved remarkable performance. However, due to the inherently limited receptive field of CNN, the pure CNN-based methods mainly focus on local regions which results in poor performance in dealing with complex scenes, as shown in Fig. 1(b).



(a) Feature detection and enhancement process



(b) Feature matching results

Fig. 1: **Feature detection and matching with CA^{2}Point.** (a) shows the feature detection process, where the proposed KAD increases the number of matches and DAK increases the number of keypoints. (b) shows the comparison between the proposed method and SuperPoint [12]. Our proposed method can avoid keypoints in the meaningless area and obtain more right matches under illumination and viewpoint changes. **First row:** Matching results of using SuperPoint. **Second row:** Matching results of using our CA^{2}Point.

On the other hand, with the successful transfer of Transformer [17] from Natural Language Processing to Computer Vision task [18], many advanced detector-based [19]–[22] and detector-free [23]–[26] matching methods have been proposed to create the correspondence across images. However, these methods need transform descriptors from the reference image and target image. In other words, the descriptors of the reference image may change with alterations of the target image, which impairs the invariance of the descriptor to some extent. Moreover, the detector-free methods do not extract explicit keypoints, which is inflexible for some applications such as back-end optimization in SLAM. In this work, we design a Local & Global Context Aggregation (LGCA) module that utilizes both local and global contexts from CNN and Transformer for feature detection, and explicitly extract keypoints and descriptors.

In addition, most feature detection and matching methods

This work was supported by Beijing Natural Science Foundation (L243008), and in part by National Natural Science Foundation of China under Grant No. 62003323 and No. 62176250.

[1]Research Center for Intelligent Computing Systems, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190, China.
[2]Hangzhou Institute for Advanced Study, UCAS, Hangzhou, 310024, China.
[3]University of Chinese Academy of Sciences, Beijing, 100049, China.
*Correspondence: Wei Li, liwei2019@ict.ac.cn

Authorized licensed use limited to: INSTITUTE OF COMPUTING TECHNOLOGY CAS. Downloaded on February 17,2026 at 23:06:07 UTC from IEEE Xplore. Restrictions apply.

focus on improving the discrimination of descriptors [19], [27], but the matchability of keypoints attracts few attention. The existing method usually could extract a lot of keypoints by adjusting the detection threshold, but only a small portion of the descriptors corresponding to these keypoints can be correctly matched, as shown in the first row of Fig. 1(b). In this work, we propose to simultaneously enhance keypoint detection and descriptor generation using the designed Point & Descriptor Cross Augmentation (PDCA) module, resulting in more matchable keypoints and descriptors.

Based on the aforementioned analysis, this work mainly focuses on detector-based feature matching methods, concentrating on improving the performance of both keypoints and descriptors simultaneously. Specifically, we propose a novel model for image feature detection that integrates local and global contexts by LGCA module, while enhancing keypoints and descriptors by PDCA module, as shown in Fig. 1(a). The key contributions of this work are:

- A novel image features detection method is proposed, which can extract robust keypoints and descriptors by context aggregation and cross augmentation strategies.
- The proposed LGCA module can adaptively aggregate local and global contexts. When embedding it into the Feature Pyramid Network (FPN) [28], the multi-scale features with global information can be extracted to disambiguate in unreliable area.
- The proposed PDCA module can obtain more keypoints and more accurate matches through the information interaction between keypoints and descriptors.
- Comprehensive experiments are conducted in image matching, pose estimation, visual localization and visual odometry task to verify the effectiveness of our method.

II. RELATED WORK

In this section, we introduce the related work about detector-based feature detection and matching methods, where the former can be divided into three categories: detect-then-describe, describe-then-detect, and detect-and-describe.

Detect-then-describe methods. Traditional handcrafted local features [9]–[11] and early learning-based one [29] belong to detect-then-describe pattern. These methods first detect a set of distinctive and repeatable points by a detector. Then, the descriptors are computed for every keypoint in a local region centred around keypoint. Finallly, the keypoints from different scenes are matched according to distance metric of descriptors. Due to each component is designed independently, improving individual component can not ensure to improve the performance of the whole pipeline.

Describe-then-detect methods. The existing methods first generate dense feature descriptors. Then, sparse keypoints are selected from these descriptors according to elaborate rules. D2D [30] designs a relative and absolute saliency strategy according to the expert prior, while PoSFeat [31] and SCFeat [32] select keypoints with the trainable network. But the handcrafted selection methods heavily rely on prior knowledge, and learning-based strategies usually require freezing the description network when training the detection network, which is difficult to guarantee optimal results.

Detect-and-describe methods. Recently, several detect-and-describe methods [12]–[15], [33]–[40] are proposed to use a single model to jointly learn keypoint detection and description. SuperPoint [12] uses a shared feature extraction backbone to obtain the feature map, and then obtain the keypoint heatmap and descriptors through the detection head and the descriptor head, respectively. Many subsequent methods [14]–[16], [37]–[39] also adopt the same design. However, these methods only output the last feature map of the backbone, lacking multi-scale features. Although ASLFeat [34], ALIKE [36] and AWDesc [38] introduce multi-scale features through simple concatenation or summation operations, the lack of learnable parameters may limit the performance of the backbone. Besides, AWDesc [38] employs Transformer [17], [18] to aggregate global information in the descriptor head, but not in the detection head. Differently, our proposed LGCA module can adaptively fuse local and global contexts, and inserting it into FPN can obtain shared feature map containing different scales and receptive fields.

Detector-based feature matching. The above methods use the nearest neighbor (NN) search to find correspondences across images. Recently, SuperGlue [19] and LightGlue [20] match local features by a learning-based approach. In addition, FeatureBooster [27] uses the geometric information of sparse keypoints to boost the off-the-shelf descriptors like SuperPoint [12]. However, using only sparse keypoints does not fully exploit the information from the entire keypoint heatmap. Inspired by [19] and [27], we propose a Keypoint Augment Descriptor (KAD) module that directly uses feature maps of keypoint to enhance dense descriptors. Moreover, a Descriptor Augment Keypoint (DAK) module is proposed to enhance keypoints through the matchability of descriptors, forcing positions corresponding to correctly matched descriptors to be detected as keypoints. Our PDCA module is composed of the KAD and DAK.

III. METHODOLOGY

As shown in Fig. 2, our proposed method consists of two components: keypoint detection and description main network and PDCA Module. Give an image I, the main network first extracts initial keypoint heatmap and descriptors, and then the PDCA module further enhances them to obtain more keypoints and more discriminative descriptors.

A. Keypoint detection and Description Main Network

The keypoint detection and description main network adopts the same architecture as SuperPoint [12] containing a feature extraction backbone and two heads (a keypoint head and a descriptor head). In particular, we extend its backbone into a Feature Pyramid Network (FPN) to fuse information from different scale space. Moreover, in order to aggregate local and global contexts into feature map directly in the backbone, the proposed LGCA module (III-B) is embedded before upsampling in the FPN. And we only execute the self-attention in the middle two blocks with the lower resolution

Authorized licensed use limited to: INSTITUTE OF COMPUTING TECHNOLOGY CAS. Downloaded on February 17,2026 at 23:06:07 UTC from IEEE Xplore. Restrictions apply.

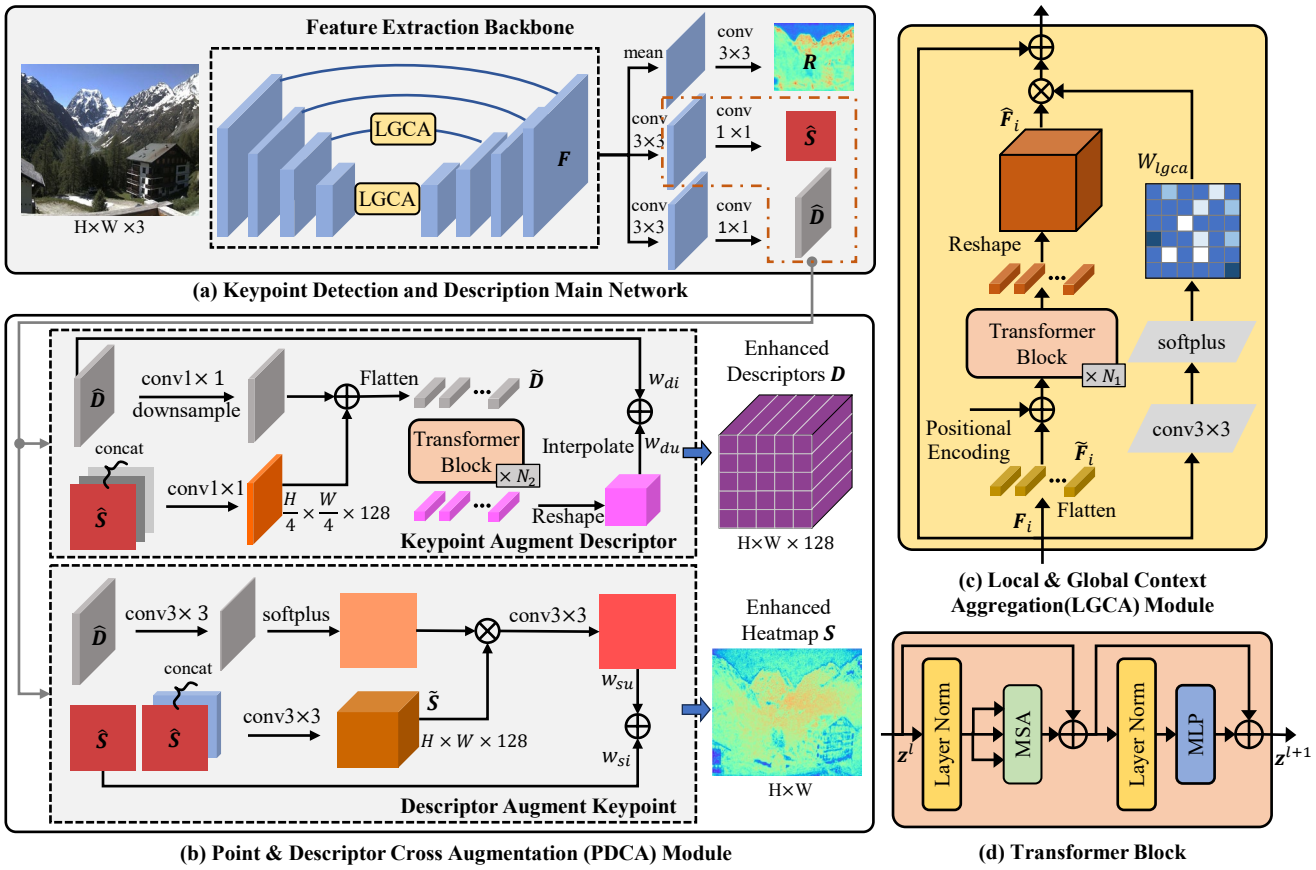


Fig. 2: **The architecture of our proposed method.** (a) Our proposed keypoint detection and description main network consists of a backbone with LGCA and two detection heads. (b) Our proposed PDCA simultaneously enhances initial keypoints and descriptors from main network. (c) Details of LGCA in main network. (d) Transformer block in LGCA.

to reduce computational and memory overhead. Thus, a shared feature map $\mathbf{F} \in \mathbb{R}^{dim \times H \times W}$ with multi-scale and different receptive field information is extracted through the FPN with LGCA module. The feature map is then fed into two simple heads to generate initial keypoint heatmap $\widehat{\mathbf{S}} \in \mathbb{R}^{1 \times H \times W}$ and descriptors $\widehat{\mathbf{D}} \in \mathbb{R}^{dim \times H \times W}$. Moreover, a descriptor reliability $\mathbf{R} \in \mathbb{R}^{1 \times H \times W}$ is predicted by an additional branch. The dimension of descriptors dim in this work is 128. It is worth noting that the initial keypoints and descriptors already have strong performance (see Table V). Fig. 2(a) shows the architecture of our main network.

B. Local & Global Context Aggregation Module

Intuitively, for those keypoints that are difficult to establish correspondence only using local features, fusing global information can gain more receptive field to disambiguate. But for those keypoints that are easy to match, introducing global information is unnecessary, even harmful.

We propose a Local & Global Context Aggregation (LGCA) Module to fuse local and global contexts adaptively according to the scene, as shown in Fig. 2(c). To achieve better speed-accuracy trade-off, the feature map $\mathbf{F}_i \in \mathbb{R}^{dim_i \times H_i \times W_i}$ from i-th CNN block is empirically transformed into a fixed-size feature map $\widetilde{\mathbf{F}}_i \in \mathbb{R}^{dim_i \times 32 \times 32}$ to avoid being affected by changes in the size of the input image. Each vector $\boldsymbol{f}_i^j \in \mathbb{R}^{dim_i}$ in $\widetilde{\mathbf{F}}_i$ can be regarded as token embeddings to be input into the vanilla Transformer [17]. As with most visual Transformer methods, we add learnable position embeddings $\mathbf{E}_i^{pos} \in \mathbb{R}^{C \times dim_i}$ to tokens to retain spatial position information as follow:

$$\mathbf{z}_i^0 = [\boldsymbol{f}_i^1; \boldsymbol{f}_i^2; \cdots; \boldsymbol{f}_i^C] + \mathbf{E}_i^{pos} \tag{1}$$

where $C = 32 \times 32$ is a constant.

Then, these token embeddings with position information are sent into a stack of $N_1 = 6$ identical Transformer blocks to obtain global information. The vanilla Transformer block is shown in Fig. 2(d), and the output of the l-th block is:

$$\hat{\mathbf{z}}^l = MSA(LN(\mathbf{z}^{l-1})) + \mathbf{z}^{l-1} \tag{2}$$

$$\mathbf{z}^l = MLP(LN(\hat{\mathbf{z}}^l)) + \hat{\mathbf{z}}^l \tag{3}$$

where $MSA(\cdot)$ donates Multi-Head Self-Attention, $MLP(\cdot)$ donates Multi-Layer Perceptron, $LN(\cdot)$ donates Layer Normalization. After reshaping and upsampling, the features with global context $\widehat{\mathbf{F}}_i \in \mathbb{R}^{dim_i \times H_i \times W_i}$ are obtained.

As aforementioned, not all keypoints need global receptive field, the LGCA module predicts a weight $W_{lgca} \in \mathbb{R}^{H_i \times W_i}$

Authorized licensed use limited to: INSTITUTE OF COMPUTING TECHNOLOGY CAS. Downloaded on February 17,2026 at 23:06:07 UTC from IEEE Xplore. Restrictions apply.

to select positions that require global context. Finally, we add the weighted global features with the input local features to obtain the features with local and global contexts.

C. Point & Descriptor Cross Augmentation Module

In this section, we propose a Point & Descriptor Cross Augmentation (PDCA) Module that aims to simultaneously enhance the initial keypoint heatmap $\widehat{\mathbf{S}}$ and descriptors $\widehat{\mathbf{D}}$. The enhanced descriptors can be obtain by aggregating the initial descriptors with the geometric information contained by the initial keypoint heatmap, such as coordinates and scores. And the initial keypoint heatmap is enhanced by the matchability of descriptors to generate enhanced keypoints. The PDCA module conforms to the following two properties:

Property 1: Those descriptors corresponding to the keypoints should be matched correctly.

Property 2: Those positions where descriptors can be matched correctly should be detected as keypoints.

Based on the above properties, we design a Keypoint Augment Descriptor (KAD) module to boost discrimination of descriptors and a Descriptor Augment Keypoint (DAK) module to increase the number of matchable keypoints.

1) ***Keypoint Augment Descriptor****:* As shown in the top of Fig. 2(b), the initial descriptors and the initial heatmap concatenated with coordinates are respectively downsampled to a size of $dim \times \frac{H}{4} \times \frac{W}{4}$. The downsampled heatmap and descriptors are transformed into $\widetilde{\mathbf{D}}$ through a summation and flattening operations. Similar to LGCA, $\widetilde{\mathbf{D}}$ are mapped into fixed size and fed into $N_2 = 6$ Transformer blocks. After reshaping and interpolating, updated descriptors and initial descriptors are summed by the learnable weight to obtain enhanced descriptors $\mathbf{D} \in \mathbb{R}^{dim \times H \times W}$. Thus, by explicitly incorporating the geometric information of keypoints which are valuable for matching into the descriptors, more robust descriptors can be obtained. We exploit the linear transformer [41] to reduce computation complexity in KAD.

2) ***Descriptor Augment Keypoint****:* The architecture of our proposed DAK module is shown in the bottom of Fig. 2(b). We predict a matching score from the initial descriptors $\widehat{\mathbf{D}}$. Then, initial keypoint heatmap and its previous feature map are concatenated to generate new feature map $\tilde{\mathbf{S}} \in \mathbb{R}^{dim \times H \times W}$. The updated heatmap, obtained by multiplying the new feature with the matching score, is added to the initial heatmap to generate enhanced keypoint heatmap $\mathbf{S} \in \mathbb{R}^{1 \times H \times W}$. This process embeds the matchability of the descriptors into the enhanced heatmap. In the enhanced heatmap, larger values indicate a higher likelihood of being a keypoint, and the corresponding descriptors have a greater probability to match correctly.

D. CA^{2}Point-Tiny

To meet the requirements of real-time tasks, we design a lightweight version of CA^{2}Point, termed CA^{2}Point-Tiny. And we employ knowledge distillation techniques to train CA^{2}Point-Tiny to preserve the performance of the full version CA^{2}Point (referred to as CA^{2}Point-Full for distinction) as much as possible. As shown in Fig. 3, we

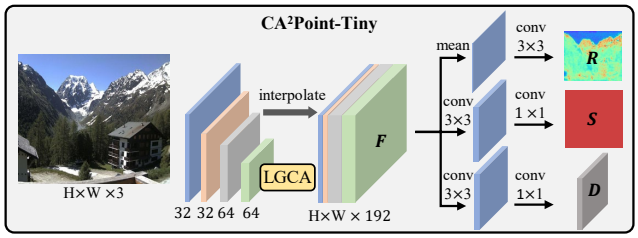


Fig. 3: **The architecture of our CA^{2}Point-Tiny.**

directly upsample and concatenate the feature maps output by the convolutional blocks without using FPN. The number of channels in convolutional blocks is set to 32-32-64-64. Futhermore, LGCA is only applied to the smallest feature map, with $\widetilde{\mathbf{F}}_i \in \mathbb{R}^{dim_i \times 16 \times 16}$ and $N_1 = 3$. The keypoint heatmaps, descriptors, and descriptor reliability are then produced through three branches identical to those in CA^{2}Point.

E. Supervision

To simplify the training process and improve the training stability, the main network and the PDCA module are separately trained in a supervised manner.

Main Network Supervision: Similar to [38], the keypoint loss $\mathcal{L}_{keyp}$ is the binary cross-entropy loss over the keypoint heatmap $\widehat{\mathbf{S}}$ as follow:

$$\mathcal{L}_{keyp} = \frac{1}{HW} \sum_{i,j}^{H,W} \mathcal{L}_{bce}(\widehat{\mathbf{S}}_{i,j}, \boldsymbol{Y}_{i,j}) \tag{4}$$

$$\mathcal{L}_{bce}(\hat{s}, y) = -\alpha \cdot y \log(\hat{s}) - (1-y)\log(1-\hat{s}) \tag{5}$$

where $\boldsymbol{Y} \in \mathbb{R}^{H \times W}$ is pseudo-ground truth label from pre-trained SuperPoint [12]. And we use iterative homographic adaptation to obtain keypoints heatmap with more pseudo-keypoints. The $\alpha = 200$ is a weight term to balance the number of ground truth keypoints and non-keypoints, $\hat{s} \in \widehat{\mathbf{S}}$ and $y \in \boldsymbol{Y}$.

The descriptor loss $\mathcal{L}_{desc}$ is the weighted triplet loss. Given a pair of overlapping images (I^1, I^2), the corresponding point sets (P^1, P^2) of size N can be obtained using ground-truth camera parameters and depth maps. These point sets, derived through sampling, are subsets of all pixels in the overlap area. The corresponding descriptor sets $(\widehat{\boldsymbol{D}}^1, \widehat{\boldsymbol{D}}^2)$ can be obtained by sampling at the (P^1, P^2). The descriptor loss is as follows:

$$\mathcal{L}_{desc}^{main} = \sum_{i=1}^{N} \mathcal{L}_{trip}(\hat{\boldsymbol{x}}_i) \tag{6}$$

$$\mathcal{L}_{trip}(\hat{\boldsymbol{x}}_i) = \frac{e^{r/\tau}}{\sum_{i=1}^{N} e^{r_i/\tau}} \max(0, \|\hat{\boldsymbol{x}}_i\|^+ - \|\hat{\boldsymbol{x}}_i\|^- + 1) \tag{7}$$

where $\hat{\boldsymbol{x}}_i = r_i \hat{\boldsymbol{d}}_i$ is the reliability weighted descriptor, $\|\hat{\boldsymbol{x}}_i\|^+ = \|r_i^1 \hat{\boldsymbol{d}}_i^1 - r_i^2 \hat{\boldsymbol{d}}_i^2\|_2$ is the positive distance of descriptor $\hat{\boldsymbol{d}}_i^1 \in \widehat{\boldsymbol{D}}^1$, $\|\hat{\boldsymbol{x}}_i\|^- = \min_{j \in 1,\ldots,N, j \neq i}(\|r_i^1 \hat{\boldsymbol{d}}_i^1 - r_j^2 \hat{\boldsymbol{d}}_j^2\|_2)$ is its hardest negative distance. The τ is the temperature and

Authorized licensed use limited to: INSTITUTE OF COMPUTING TECHNOLOGY CAS. Downloaded on February 17,2026 at 23:06:07 UTC from IEEE Xplore. Restrictions apply.

the $r \in \mathbf{R}$ is the descriptor reliability. The loss function of the main network is defined as:

$$\mathcal{L}^{main} = \mathcal{L}_{keyp} + \mathcal{L}^{main}_{desc} \tag{8}$$

KAD Supervision: To increase the number of matches, we additionally select M_1 detected keypoints during the training KAD, and append them to the corresponding point set (P^1, P^2). Specifically, the top-M_1 keypoints are selected according to the score in the initial keypoint heatmap.

In addition, to ensure that the initial descriptors will be enhanced, we design a new loss to force the discriminability of enhanced descriptors $(\boldsymbol{d}_i^1, \boldsymbol{d}_i^2) \in (\boldsymbol{D}^1, \boldsymbol{D}^2)$ to be better than the initial ones $(\hat{\boldsymbol{d}}_i^1, \hat{\boldsymbol{d}}_i^2)$:

$$\mathcal{L}^{boost}_{desc} = \frac{1}{N+M_1} \sum_{i=1}^{N+M_1} \max(0, \frac{\mathcal{L}_{trip}(\boldsymbol{x}_i)}{\mathcal{L}_{trip}(\hat{\boldsymbol{x}}_i)} - m_k) \tag{9}$$

where $m_k = 0.8$ is the margin used to control the degree of descriptor enhancement. The loss for KAD is defined as:

$$\mathcal{L}^{KAD} = \sum_{i=1}^{N+M_1} \mathcal{L}_{trip}(\boldsymbol{x}_i) + \beta \cdot \mathcal{L}^{boost}_{desc} \tag{10}$$

where $\beta = 0.1$ is a weight to regulate the second term.

DAK Supervision: The purpose of DAK is to increase the number of keypoints. We randomly select M_2 positions from all descriptors matched correctly, and append them to label $\boldsymbol{Y}$ to obtain new label $\boldsymbol{Y}^{'}$. For the selected points, if they are not already included in the pseudo-keypoints ground truth $\boldsymbol{Y}$, this operation will increase the number of detected keypoints. If they are already present in $\boldsymbol{Y}$, it indicates that these keypoints are highly reliable and stable. In this case, the operation effectively assigns greater weight to these reliable keypoints. Equation (4) is then used for training the enhanced keypoint heatmap $\mathbf{S}$. The difference is that α is set to 150 due to increase in the number of ground truth keypoints.

CA^{2}Point Tiny Supervision: To employ the knowledge distillation to improve the performance of CA^{2}Point-Tiny, we use CA^{2}Point-Full as the teacher model to transfer knowledge. Our knowledge distillation framework consists of three distillation losses formulated using the Mean Squared Error (MSE) loss: keypoint heatmap distillation loss $\mathcal{L}_{hd} = \frac{1}{HW}\sum_{i,j}^{H,W}(\mathbf{S}^t_{i,j} - \mathbf{S}^s_{i,j})^2$, reliability distillation loss $\mathcal{L}_{rd} = \frac{1}{HW}\sum_{i,j}^{H,W}(\mathbf{R}^t_{i,j} - \mathbf{R}^s_{i,j})^2$, and descriptor distillation loss $\mathcal{L}_{dd} = \frac{1}{HW\,dim}\sum_{i,j,c}^{H,W,dim}(\mathbf{D}^t_{i,j,c} - \mathbf{D}^s_{i,j,c})^2$. The total loss of CA^{2}Point-Tiny is defined as:

$$\mathcal{L}^{tiny} = \mathcal{L}_{keyp} + \mathcal{L}_{desc} + \mathcal{L}_{hd} + \mathcal{L}_{rd} + \mathcal{L}_{dd} \tag{11}$$

IV. EXPERIMENTS

In this section, we first introduce the implementation details in training and experiments. Then, we evaluate the performance of our method for the tasks of homography estimation, relative pose estimation, visual localization, and visual odometry. Finally, a complete ablation study is conducted to verify the effectiveness of each component.

A. Implementation Details

The proposed method implemented by Pytorch [42]. In the LGCA and PDCA module, the number of attention head is set to 8. For training, we adopt the same dataset as [38] which contains 11,800 image pairs with 400×400 image size. The size of point sets N is set to 400. The M_1 in KAD is set to 400, and the M_2 in DAK varies with the actual number of matches during the training. The Adam optimizer is used to optimize the network with the initial learning rate 0.001. The network converges after 50 epochs of training on 2 Tesla V100-SXM2 GPUs with a batch size of 2. For inference, the keypoint detection threshold is set to 0.85 and the non-maximum suppression radius is 4. All experiments are run on a single NVIDIA RTX 3090 GPU.

B. Homography Estimation

Setup. HPatches [1] is an image matching and homography estimation benchmark with illumination and viewpoint changes. Following [15], we resize the shorter image edge to 480 for features extraction, and report the keypoints *Repeatability*, *Mean Matching Accuracy (MMA)*, *Homography Estimation Accuracy (Hom. Est. Acc.)*, *Homography Estimation AUC (Hom. Est. AUC)*, *Match Score (M.S.)*, the number of keypoints extracted and matched within the covisible area, and the number of correctly matched keypoints for threshold $\epsilon = 3$. We compare CA^{2}Point with sparse feature extractors including SuperPoint (SP) [12], D2-Net [13], R2D2 [14], DISK [35], SFD2 [37], AWDesc [38] and SiLK [15], sparse feature booster and matchers including FeatureBooster (FB) [27], SuperGlue (SG) [19] and LightGlue (LG) [20].

Results. As shown in Table I, when using detection threshold to extract keypoints, our method outperforms all methods in terms of *Hom. Est. Acc.*, *Hom. Est. AUC*, and *M.S.*. Only the *MMA* and *Repeatability* are slightly lower than the DISK [35]. However, when the number of keypoints increases to 10k, all metrics show a significant increase, except for the *M.S.*, because more keypoints introduce more mismatches. It is worth noting that our method outperforms sparse boosting and matching methods in almost all metrics and the lightweight CA^{2}Point-Tiny shows impressive performance, even exceeding CA^{2}Point-Full.

C. Relative Pose Estimation

Setup. We use the image pairs from MegaDepth-1500 [23], [43] to show the performance of CA^{2}Point for relative pose estimation. Following [20], we resize the larger dimension to 1600 and extract 2048 features per image. We solve an essential matrix using RANSAC and LO-RANSAC with LM-refinement [44], respectively, and report the AUC of the pose error at thresholds 5°, 10°, and 20°. We compare CA^{2}Point with semi-dense matchers including LoFTR [23], MatchFormer [25], ASpanFormer [26] and Efficient LoFTR [24]. As for sparse keypoint detection and matching methods, we compare with SuperPoint [12] with different matchers including Nearest-Neighbor(NN), SuperGlue [19], LightGlue [20], SGMNet [21], and MambaGlue [22]. For fairness, we train the LightGlue matcher based on the CA^{2}Point.

Authorized licensed use limited to: INSTITUTE OF COMPUTING TECHNOLOGY CAS. Downloaded on February 17,2026 at 23:06:07 UTC from IEEE Xplore. Restrictions apply.

TABLE I: **Evaluation results of homography estimation on HPatches [1].** The top two results are marked with **bold** and underline. MNN is the Mutual Nearest Neighbor matcher. Ours-F and Ours-T denote CA^{2}Point-Full and CA^{2}Point-Tiny, respectively.

Method	Repeatability		Hom. Est. Acc.		Hom. Est. AUC		MMA		M.S.		# of keypoints($\epsilon = 3$)		
	$\epsilon = 1$	$\epsilon = 3$	$\epsilon = 1$	$\epsilon = 3$	$\epsilon = 1$	$\epsilon = 3$	$\epsilon = 1$	$\epsilon = 3$	$\epsilon = 1$	$\epsilon = 3$	pre-	post-	right
SuperPoint [12] + MNN	0.329	0.606	0.441	0.829	0.207	0.527	0.414	0.730	0.268	0.485	810	507	393
D2-Net [13] + MNN	0.088	0.374	0.066	0.424	0.023	0.175	0.134	0.476	0.071	0.240	2283	1062	547
R2D2 [14] + MNN	0.269	0.609	0.433	0.750	0.190	0.488	0.410	0.754	0.186	0.320	1805	666	577
DISK [35] + MNN	0.373	**0.690**	0.447	0.810	0.220	0.522	**0.521**	**0.845**	0.320	0.515	3149	1797	1621
SFD2 [37] + MNN	0.298	0.605	0.357	0.812	0.152	0.469	0.399	0.770	0.249	0.492	978	600	481
AWDesc [38] + MNN	0.335	0.599	0.500	0.852	0.222	0.559	0.455	0.784	0.298	0.520	1048	675	545
Ours-F ($\alpha = 0.85$)+MNN	**0.376**	0.636	**0.571**	**0.896**	**0.293**	0.614	0.503	0.809	**0.333**	**0.546**	1028	670	561
Ours-T ($\alpha = 0.85$)+MNN	0.359	0.617	0.566	0.888	0.291	**0.621**	0.486	0.778	0.306	0.498	1048	641	522
SiLK (top-10k) [15]+MNN	0.564	0.791	0.621	0.857	**0.398**	0.649	0.555	0.690	0.238	0.300	10108	3976	3028
Ours-F (top-10k)+MNN	0.574	**0.816**	0.588	0.872	0.363	0.630	**0.622**	0.829	**0.255**	**0.336**	10001	3835	3363
Ours-T (top-10k)+MNN	**0.600**	0.813	**0.633**	**0.891**	0.385	**0.665**	0.618	**0.835**	0.249	0.330	10001	3724	3305
SP [12]+FB [27]+MNN	0.329	0.606	0.409	0.819	0.186	0.501	0.416	0.738	0.276	0.504	810	527	408
SP [12]+SG [19] (Indoor)	0.329	0.606	0.393	0.790	0.184	0.488	0.416	0.754	0.290	0.538	810	572	436
SP [12]+SG [19] (Outdoor)	0.329	0.606	0.483	0.848	0.216	0.549	0.493	**0.909**	0.303	**0.578**	810	519	468
SP [12]+LG [20]	0.329	0.606	0.445	0.840	0.197	0.525	0.478	0.882	0.301	0.569	810	524	461
Ours-F ($\alpha = 0.85$)+MNN	**0.376**	**0.636**	**0.571**	**0.896**	**0.293**	0.614	**0.503**	0.809	**0.333**	0.546	1028	670	561
Ours-T ($\alpha = 0.85$)+MNN	0.359	0.617	0.566	0.888	0.291	**0.621**	0.486	0.778	0.306	0.498	1048	641	522

TABLE II: **Evaluation results of relative pose estimation on MegaDepth [43].** The group-specific best results are marked with **bold**. † indicates model trained on top of our method.

	Method	RANSAC AUC	LO-RANSAC AUC
		5° / 10° / 20°	
Dense	LoFTR [23]	52.8 / 69.2 / 81.2	66.4 / 78.6 / 86.5
	MatchFormer [25]	53.3 / 69.7 / 81.8	66.5 / 78.9 / 87.5
	ASpanFormer [26]	58.3 / 73.3 / **84.2**	69.4 / **81.1** / **88.9**
	Efficient LoFTR [24]	**58.4** / **73.4** / **84.2**	**69.5** / 80.9 / 88.8
SuperPoint	NN+mutual	31.7 / 46.8 / 60.1	51.0 / 54.1 / 73.6
	SuperGlue [19]	49.7 / 67.1 / **80.6**	65.8 / 78.7 / 87.5
	LightGlue [20]	49.9 / 67.0 / 80.1	66.7 / 79.3 / 87.9
	SGMNet [21]	43.2 / 61.6 / 75.6	59.8 / 74.1 / 83.9
	MambaGlue [22]	50.1 / 67.5 / 80.3	65.8 / 78.7 / 87.6
Ours	F+NN+mutual	35.3 / 50.8 / 63.8	55.9 / 68.2 / 77.0
	F+LightGlue†	**50.9** / **68.1** / **80.6**	**66.8** / **80.5** / **88.2**
	T+NN+mutual	33.0 / 48.3 / 60.9	53.4 / 65.2 / 73.9

TABLE III: **Visual localization evaluation on the Aachen Day-Night benchmark v1.1 [45].** The best results for each group are marked with **bold**.

	Method	Day	Night
		(0.25m, 2°) / (0.5m, 5°) / (1.0m, 10°)	
Dense	LoFTR [23]	88.7 / 95.6 / **99.0**	**78.5** / 90.6 / 99.0
	ASpanFormer [26]	89.4 / 95.6 / **99.0**	77.5 / **91.6** / **99.5**
	Efficient LoFTR [24]	**89.6** / **96.2** / **99.0**	77.0 / 91.1 / **99.5**
SuperPoint	NN+mutual	84.8 / 90.3 / 93.8	65.3 / 72.4 / 85.7
	SuperGlue [19]	88.2 / 95.5 / 98.7	86.7 / 92.9 / **100.0**
	LightGlue [20]	**89.2** / 95.4 / 98.5	**87.8** / 93.9 / **100.0**
	SGMNet [21]	86.8 / 94.2 / 97.7	83.7 / 91.8 / 99.0
	MambaGlue [22]	89.0 / 95.3 / 98.7	86.7 / 93.9 / **100.0**
Ours	F+NN+mutual	84.7 / 91.5 / 94.4	68.6 / 80.1 / 88.0
	F+LightGlue†	**89.2** / **96.2** / **98.9**	87.5 / **94.1** / **100.0**
	T+NN+mutual	84.6 / 90.3 / 94.1	66.0 / 78.5 / 85.9

Results. Table II shows that CA^{2}Point outperforms all feature extraction and matching methods in the sparse group. Notably, our CA^{2}Point-Full with LightGlue achieves competitive performance compared to the semi-dense method LoFTR. And CA^{2}Point-Tiny also shows promising results with a slight performance degradation, while boosting the running efficiency.

D. Outdoor Visual Localization

Setup. We evaluate long-term visual localization in the Aachen Day-Night benchmark [45]–[47] using the Hierarchical Localization pipeline [48]. We estimate a camera pose with RANSAC and a Perspective-n-Point(PnP) solver, and report the pose recall at multiple thresholds. We compare our method with semi-dense matchers and sparse feature extractor SuperPoint [12]. Following [20], we extract up to 4096 features for sparse extractor and match them with different matchers.

Result. As shown in TableIII, CA^{2}Point outperforms SuperPoint when using the same matchers and is even competitive with semi-dense matchers when paired with LightGlue. The lightweight version shows promising performance with a trade-off between speed and accuracy.

E. Monocular Visual Odometry

Setup. The KITTI [49] dataset is one of the most mainstream odometry and SLAM benchmarks. We replace ORB feature extractor in the original ORB-SLAM2 [5] with the proposed CA^{2}Point-Tiny, which is called CA-SLAM. Following ORB-SLAM2 [5], the Root Mean Square Error (RMSE) of the Absolute Trajectory Error (ATE) is used to evaluate the accuracy of visual odometry. We use the EVO tool [50] to

Authorized licensed use limited to: INSTITUTE OF COMPUTING TECHNOLOGY CAS. Downloaded on February 17,2026 at 23:06:07 UTC from IEEE Xplore. Restrictions apply.

TABLE IV: **Evaluation results of visual odometry on the KITTI [49] dataset.** RMSE (m) of ATE is reported.

Methods	00	02	03	04	05	06	07	08	09	10
ORB-SLAM2 [5]	**71.52**	**35.69**	1.16	1.14	**36.14**	51.47	16.92	48.63	58.00	7.95
SP-SLAM	77.68	41.88	1.63	**0.29**	36.54	51.21	17.93	48.34	55.07	**6.05**
CA-SLAM	77.96	43.32	**1.06**	**0.29**	36.99	**48.70**	**15.99**	**44.79**	**48.12**	6.14

compute ATE between ground truth trajectories and predicted trajectories and report the median RMSE error of the ATE over five executions for each sequence. We compare CA-SLAM with ORB-SLAM2 [5], and SP-SLAM which is implemented by replacing ORB features with SuperPoint [12] in [5]. We disable the loop closing for all methods to only evaluate the performance of visual odometry.

Results. As shown in Table IV, our CA-SLAM outperforms ORB-SLAM2 [5] in most sequences, with an average superiority of over 0.5m. For sequence 09, CA-SLAM shows a decrease of up to 9.88m compared to [5]. In addition, CA-SLAM also outperforms SP-SLAM in almost all sequences. This is attributed to the fact that our CA^{2}Point can extract more discriminative keypoints and descriptors.

F. Ablation Study

To verify the effectiveness of each proposed module in CA^{2}Point, we conduct detailed ablation study on the HPatches dataset with results shown in Table V. The model [A] is an extension of the SuperPoint [12] backbone that incorporates FPN for upsampling and feature aggregation. For model [B], the LGCA module is inserted into FPN to capture global context during the feature extraction process. Then, the model [C] enhances the discrimination of descriptors output by model [A] through the KAD module. For model [D], more keypoints can be detected by the DAK module. The model [E] is the full model of our CA^{2}Point. Finally, the model [F] is the lightweight version of our CA^{2}Point.

Compared to the [A], the performance on the HPatches is improved using the LGCA and KAD module. Although the DAK module results in a slight decrease in *MMA* and *M.S.*, the model [D] actually detects more matchable keypoints than [A] (929 vs. 851). And the model [E] also performs better than the model [B] and [C], which indicates that the DAK module is beneficial for overall performance improvement. The lightweight CA^{2}Point-Tiny achieves a trade-off between speed and accuracy, with significantly high computational efficiency and competitive matching accuracy.

V. CONCLUSIONS

In this work, we propose a novel keypoint detection and description method named CA^{2}Point, with a LGCA module to obtain local and global contexts and a PDCA module to simultaneously enhance keypoints and descriptors. With the help of the context aggregation and the cross augmentation, our proposed method can extract more keypoints and more discriminative descriptors for matching under some challenging scenarios. And we design a lightweight version of CA^{2}Point to achieve a balance between speed and accuracy.

TABLE V: **Ablation study.** We report Homography Estimation Accuracy (HA), Mean Matching Accuracy (MMA), matching score (M.S.) and average running time per image on HPatches datasets.

Models	LGCA	PDCA		HA	MMA	M.S.	Time (ms)
		KAD	DAK	$\epsilon = 3$	$\epsilon = 3$	$\epsilon = 3$	
[A]	✘	✘	✘	0.872	0.785	0.484	27.3
[B]	✓	✘	✘	0.879	0.803	0.534	41.9
[C]	✘	✓	✘	0.895	0.797	0.504	60.7
[D]	✘	✘	✓	0.886	0.784	0.483	39.4
[E]	✓	✓	✓	**0.896**	**0.809**	**0.546**	87.3
[F]	CA^{2}Point-Tiny			0.888	0.778	0.498	**22.1**

Experiments on various downstream tasks demonstrate the superiority of our method. Future works include jointly training the keypoint detection and description main network and the PDCA module to further improve performance and introducing the loop closure in the Visual SLAM task to improve the localization accuracy.

REFERENCES

[1] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, "Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5173–5182.

[2] E. Adel, M. Elmogy, and H. Elbakry, "Image stitching based on feature extraction techniques: a survey," *International Journal of Computer Applications*, vol. 99, no. 6, pp. 1–8, 2014.

[3] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4104–4113.

[4] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[5] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[6] X. Peng, Z. Liu, W. Li, P. Tan, S. Y. Cho, and Q. Wang, "Dvi-slam: A dual visual inertial slam network," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 12 020–12 026.

[7] X. Su, S. Eger, A. Misik, D. Yang, R. Pries, and E. Steinbach, "Hpf-slam: An efficient visual slam system leveraging hybrid point features," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 15 929–15 935.

[8] K. Xu, Y. Hao, S. Yuan, C. Wang, and L. Xie, "Airslam: An efficient and illumination-robust point-line visual slam system," *IEEE Transactions on Robotics*, 2025.

[9] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[10] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Surf: Speeded-up robust features," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

Authorized licensed use limited to: INSTITUTE OF COMPUTING TECHNOLOGY CAS. Downloaded on February 17,2026 at 23:06:07 UTC from IEEE Xplore. Restrictions apply.

[11] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 2564–2571.

[12] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 224–236.

[13] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler, "D2-net: A trainable cnn for joint description and detection of local features," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8092–8101.

[14] J. Revaud, C. De Souza, M. Humenberger, and P. Weinzaepfel, "R2d2: Reliable and repeatable detector and descriptor," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[15] P. Gleize, W. Wang, and M. Feiszli, "Silk: Simple learned keypoints," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 22 499–22 508.

[16] H. Yao, N. Hao, C. Xie, and F. He, "Edgepoint: Efficient point detection and compact description via distillation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 766–772.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[18] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proceedings of the International Conference on Learning Representations*, 2021.

[19] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4938–4947.

[20] P. Lindenberger, P.-E. Sarlin, and M. Pollefeys, "Lightglue: Local feature matching at light speed," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 17 627–17 638.

[21] H. Chen, Z. Luo, J. Zhang, L. Zhou, X. Bai, Z. Hu, C.-L. Tai, and L. Quan, "Learning to match features with seeded graph matching network," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 6301–6310.

[22] K. Ryoo, H. Lim, and H. Myung, "MambaGlue: Fast and Robust Local Feature Matching With Mamba," *arXiv preprint arXiv:2502.00462*, 2025.

[23] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, "Loftr: Detector-free local feature matching with transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8922–8931.

[24] Y. Wang, X. He, S. Peng, D. Tan, and X. Zhou, "Efficient loftr: Semi-dense local feature matching with sparse-like speed," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 21 666–21 675.

[25] Q. Wang, J. Zhang, K. Yang, K. Peng, and R. Stiefelhagen, "Matchformer: Interleaving attention in transformers for feature matching," in *Proceedings of the Asian Conference on Computer Vision*, 2022, pp. 2746–2762.

[26] H. Chen, Z. Luo, L. Zhou, Y. Tian, M. Zhen, T. Fang, D. Mckinnon, Y. Tsin, and L. Quan, "Aspanformer: Detector-free image matching with adaptive span transformer," in *Proceedings of European Conference on Computer Vision*. Springer, 2022, pp. 20–36.

[27] X. Wang, Z. Liu, Y. Hu, W. Xi, W. Yu, and D. Zou, "Featurebooster: Boosting feature descriptors with a lightweight neural network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7630–7639.

[28] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2117–2125.

[29] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "Lift: Learned invariant feature transform," in *Proceedings of European Conference on Computer Vision*. Springer, 2016, pp. 467–483.

[30] Y. Tian, V. Balntas, T. Ng, A. Barroso-Laguna, Y. Demiris, and K. Mikolajczyk, "D2d: Keypoint extraction with describe to detect approach," in *Proceedings of the Asian Conference on Computer Vision*, 2020.

[31] K. Li, L. Wang, L. Liu, Q. Ran, K. Xu, and Y. Guo, "Decoupling makes weakly supervised local feature better," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 15 838–15 848.

[32] J. Sun, J. Zhu, and L. Ji, "Shared coupling-bridge for weakly supervised local feature learning," *arXiv preprint arXiv:2212.07047*, 2022.

[33] W. Song, R. Yan, B. Lei, and T. Okatani, "Globalizing local features: Image retrieval using shared local features with pose estimation for faster visual localization," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 6290–6297.

[34] Z. Luo, L. Zhou, X. Bai, H. Chen, J. Zhang, Y. Yao, S. Li, T. Fang, and L. Quan, "Aslfeat: Learning local features of accurate shape and localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6589–6598.

[35] M. Tyszkiewicz, P. Fua, and E. Trulls, "Disk: Learning local features with policy gradient," *Advances in Neural Information Processing Systems*, vol. 33, pp. 14 254–14 265, 2020.

[36] X. Zhao, X. Wu, J. Miao, W. Chen, P. C. Chen, and Z. Li, "Alike: Accurate and lightweight keypoint detection and descriptor extraction," *IEEE Transactions on Multimedia*, vol. 25, pp. 3101–3112, 2022.

[37] F. Xue, I. Budvytis, and R. Cipolla, "Sfd2: Semantic-guided feature detection and description," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 5206–5216.

[38] C. Wang, R. Xu, K. Lu, S. Xu, W. Meng, Y. Zhang, B. Fan, and X. Zhang, "Attention weighted local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 9, pp. 10 632–10 649, 2023.

[39] M. A. Karaoglu, V. Markova, N. Navab, B. Busam, and A. Ladikos, "Ride: Self-supervised learning of rotation-equivariant keypoint detection and invariant description for endoscopy," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 10 764–10 771.

[40] J. He, Y. Gao, T. Zhang, Z. Zhang, and F. Wu, "D2former: Jointly learning hierarchical detectors and contextual descriptors via agent-based transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 2904–2914.

[41] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret, "Transformers are rnns: Fast autoregressive transformers with linear attention," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5156–5165.

[42] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[43] Z. Li and N. Snavely, "Megadepth: Learning single-view depth prediction from internet photos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2041–2050.

[44] V. Larsson and contributors, "PoseLib - Minimal Solvers for Camera Pose Estimation," 2020. [Online]. Available: https://github.com/vlarsson/PoseLib

[45] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, *et al.*, "Benchmarking 6dof outdoor visual localization in changing conditions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8601–8610.

[46] Z. Zhang, T. Sattler, and D. Scaramuzza, "Reference pose generation for long-term visual localization via learned features and view synthesis," *International Journal of Computer Vision*, vol. 129, no. 4, pp. 821–844, 2021.

[47] T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt, "Image retrieval for image-based localization revisited." in *BMVC*, vol. 1, no. 2, 2012, p. 4.

[48] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 12 716–12 725.

[49] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The international journal of robotics research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[50] M. Grupp. (2017) evo: Python package for the evaluation of odometry and slam. https://github.com/MichaelGrupp/evo.

Authorized licensed use limited to: INSTITUTE OF COMPUTING TECHNOLOGY CAS. Downloaded on February 17,2026 at 23:06:07 UTC from IEEE Xplore. Restrictions apply.