Segment Anything Model Is a Good Teacher for Local Feature Learning

Jingqian Wu, Rongtao Xu, *Member, IEEE*, Zach Wood-Doughty, Changwei Wang, Shibiao Xu, *Member, IEEE*, and Edmund Y. Lam, *Fellow, IEEE*

***Abstract*—Local feature detection and description play an important role in many computer vision tasks, which are designed to detect and describe keypoints in any scene and any downstream task. Data-driven local feature learning methods need to rely on pixel-level correspondence for training. However, a vast number of existing approaches ignored the semantic information on which humans rely to describe image pixels. In addition, it is not feasible to enhance generic scene keypoints detection and description simply by using traditional common semantic segmentation models because they can only recognize a limited number of coarse-grained object classes. In this paper, we propose SAMFeat to introduce SAM (segment anything model), a foundation model trained on 11 million images, as a teacher to guide local feature learning. SAMFeat learns additional semantic information brought by SAM and thus is inspired by higher performance even with limited training samples. To do so, first, we construct an auxiliary task of Attention-weighted Semantic Relation Distillation (ASRD), which adaptively distillates feature relations with category-agnostic semantic information learned by the SAM encoder into a local feature learning network, to improve local feature description using semantic discrimination. Second, we develop a technique called Weakly Supervised Contrastive Learning Based on Semantic Grouping (WSC), which utilizes semantic groupings derived from SAM as weakly supervised signals, to optimize the metric space of local descriptors. Third, we design an Edge Attention Guidance (EAG) to further improve the accuracy of local feature detection and description by prompting the network to pay more attention to the edge region guided by SAM. SAMFeat's performance on various tasks, such as image matching on HPatches, and long-term visual localization on Aachen Day-Night showcases its superiority over previous local features. The release code is available at https://github.com/vignywang/SAMFeat.**

***Index Terms*—Local feature and descriptor learning, segment anything model, computer vision.**

I. Introduction

LOCAL feature detection and description is a basic task of computer vision, which is widely used in image matching [1], structure from motion (SfM) [2], simultaneous localization and mapping (SLAM) [3], visual localization [4], and image retrieval [5] tasks. Traditional schemes such as SIFT [6], and ORB [7] based hand-crafted heuristics are not able to cope with drastic illumination and viewpoint changes [1]. Under the wave of deep learning [8], [9], [10], [11], [12], many data-driven local feature learning methods [13], [14] have recently achieved excellent performance. While many works have been done for training local descriptors based on completely accurate and dense ground truth correspondences [15] between image pairs, a vast number of these works ignored the semantic information on which humans rely to describe image pixels. Even though few previous works adapted the straightforward idea of using traditional common semantic segmentation models to facilitate the detection and description of keypoints, it is not feasible in practice because they can only recognize a limited number of coarse-grained object categories and are not competent for keypoint detection and description in generalized scenarios [16].

Recently, foundation models [17] have revolutionized the field of artificial intelligence [18]. These models, trained on billions of examples, presented strong zero-shot generalization capabilities across a variety of downstream tasks. In this study, we advocate the integration of SAM [16], a foundation model that is able to segment "anything" in "any scene", into the realm of local feature learning. This synergy enhances the robustness and enriches the supervised signals available for local feature learning, encompassing high-level category-agnostic semantics and detailed low-level edge structure information.

In recent years, works have attempted to introduce pixel-level semantics of images (*i.e.* semantic segmentation) into local feature learning-based visual localization. Some methods

Received 17 June 2024; revised 14 January 2025 and 11 February 2025; accepted 18 March 2025. Date of publication 28 March 2025; date of current version 3 April 2025. This work was supported in part by the National Science and Technology Major Project under Grant 2022ZD0116800; in part by Beijing Natural Science Foundation under Grant JQ23014 and Grant L231013; in part by Taishan Scholars Program under Grant TSQN202211214 and Grant TSQN202408245; in part by Shandong Excellent Young Scientists Fund Program (Overseas) under Grant 2023HWYQ-113; in part by Shandong Provincial Natural Science Foundation for Young Scholars under Grant ZR2024QF110 and Grant ZR2024QF052; in part by the National Natural Science Foundation of China under Grant 62271074, Grant U21A20515, Grant 62376271, and Grant U22B2034; and in part by the Key Laboratory of Computing Power Network and Information Security, Ministry of Education under Grant 2024PY021. The associate editor coordinating the review of this article and approving it for publication was Dr. Hasan AlMarzouqi. *(Corresponding author: Changwei Wang.)*

Jingqian Wu and Edmund Y. Lam are with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong.

Rongtao Xu is with MAIS, Institute of Automation, Chinese Academy of Sciences, Beijing 100045, China.

Zach Wood-Doughty is with the Department of Computer Science, Northwestern University, Evanston, IL 60201 USA.

Changwei Wang is with the Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250316, China, and also with Shandong Provincial Key Laboratory of Computing Power Internet and Service Computing, Shandong Fundamental Research Center for Computer Science, Jinan 250000, China (e-mail: changweiwang@sdas.org).

Shibiao Xu is with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China.

Digital Object Identifier 10.1109/TIP.2025.3554033

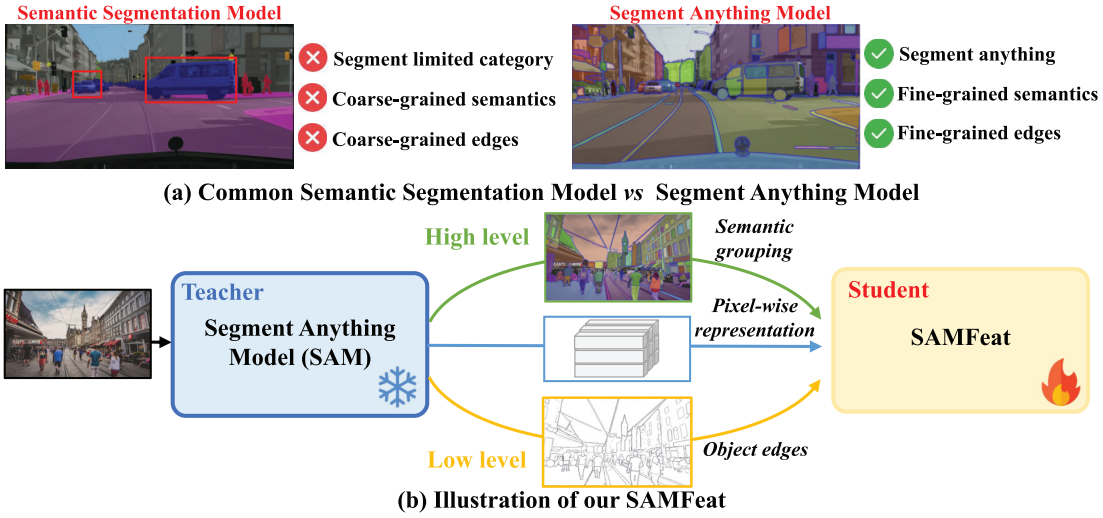


Fig. 1. (a): Difference between segment anything model and common semantic segmentation model. (b): Schematic diagram of proposed SAMFeat.

utilized semantic information to filter keypoints [19] and optimize matching [20], while other works utilized semantic information [21] to guide the learning of keypoints detection and improve the performance of the local descriptors in a specific visual localization setting by using feature-level distillation. However, these visual localization pipelines and methods are based on common semantic segmentation models and are difficult to generalize to feature matching tasks. As shown in Figure 1 (a), this is because, on one hand, common semantic segmentation can only assign semantics to limited categories (*e.g.* cars, streets, people) which is difficult to generalize to generic scenarios and open world situations [16]. On the other hand, the semantic information for semantic segmentation is coarse-grained, *e.g.*, pixels of wheels and windows are given the same labels for a car. This is detrimental to mining the unique discriminative properties of local features.

The recent SAM [16] is a visual foundation model trained on 11 million images that can segment any objects based on prompt input. Compared to common semantic segmentation models, SAM has three unique properties that can be used to fuel local feature learning. ***i)*** SAM is trained on a large amount of data, and therefore, can segment any object and can be adapted to any scene rather than being limited to certain categories and scenes like common semantic segmentation models. SAM's robust zero-shot performance could be a helpful enhancement in learning and describing features in complex scenes. ***ii)*** SAM can obtain fine-grained component-level semantic segmentation results, thus allowing for more accurate modeling of semantic relationships between pixels. In addition, SAM can derive fine-grained category-agnostic semantic masks that can be used as semantic groupings of pixels to guide local feature learning. ***iii)*** SAM can detect more detailed edges, whereas edge regions tend to be more prone to critical points and contain more distinguishing information, which helps feature learning by providing accurate guidance in keypoint localization.

In our SAMFeat, we propose three special strategies to boost the performance of local feature learning based on these three properties of SAM. **First,** we construct an auxiliary task of Attention-weighted Semantic Relation Distillation (ASRD) for distilling category-agnostic pixel semantic relations learned by the SAM encoder into a local feature learning network with attention guide, thus using semantic discriminative to improve local feature description. **Second,** we develop a technique called Weakly Supervised Contrastive Learning Based on Semantic Grouping (WSC) to optimize the metric space of local descriptors using SAM-derived semantic groupings as weakly supervised signals. **Third,** we design an Edge Attention Guidance (EAG) to further improve the localization accuracy and description ability of local features by prompting the network to pay more attention to the edge region. Since the SAM model is only used as a teacher during training, our SAMFeat can efficiently extract local features during inference without burdening the computational consumption of the SAM network.

II. Related Work

A. Local Features and Beyond

Early hand-crafted local features have been investigated for decades and are comprehensively evaluated in [22]. In the wave of deep learning, many data-driven learnable local features have been proposed for improving detectors based on different focuses on [23] and [24], descriptors [25], [26], [27], [28], and end-to-end detection and description [13], [29], [30], [31], [32], [33], [34]. Beyond localized features, some learnable advanced matchers have recently been developed to replace the traditional nearest neighbor matcher (NN) to get more accurate matching results. Sparse matchers such as SuperGlue [35] and LightGlue [36] take off-the-shelf local features as input to predict matches using a GNN or Transformer, however, their time complexity scales quadratically with the number of keypoints. Dense matchers [37], [38], [39] compute the correspondence between pixels end-to-end based on the correlation volume, while they spend more memory and space consumption than sparse matchers [21]. Recently, some

works [40], [41] have been devoted to building large-scale matchers. While they achieved strong performance, the training data and computational resources required are extremely high, and the inference speed for real-time applications still requires improvements. Our work centers on enhancing the efficiency and performance of an end-to-end generalized local feature learning approach. We aim to achieve performance comparable to advanced matchers while only using nearest-neighbor matching across various downstream tasks. This is particularly crucial in resource-constrained scenarios demanding high operational efficiency.

B. Segment Anything Model

The Segment Anything Model (SAM) [16] has achieved remarkable advancements in expanding the scope of segmentation tasks, thereby significantly fostering the evolution of fundamental models in computer vision. SAM incorporates prompt learning techniques in the field of NLP to flexibly implement model building and builds an image engine through interactive annotations, which performs better in techniques such as instance analysis, edge detection, object proposal, and text-to-mask. SAM is specifically designed to address the challenge of segmenting a wide range of objects in complex visual scenes. Unlike traditional approaches that focus on segmenting specific object classes, SAM's primary objective is to segment anything, providing a versatile solution for diverse and challenging scenarios. Many works [42], [43] now build upon SAM for downstream vision tasks such as zero-shot camouflaged object detection [44], open-world segmentation [45], specific niche segmentation [46], *etc* [47]. Unlike them, we advocate for the application of SAM to local feature learning. To the best of our knowledge, our work is the first to apply SAM to feature learning and matching tasks. There is, indeed, other newly proposed state-of-the-art work that incorporates other visual foundation models to tackle feature learning tasks. For example, ROMA [40] proposed to incorporate the encoder from DINO-V2 [48] directly into their feature learning framework and fine-tune it in the training stage. However, the computational cost for training and inference of such a method is extremely high, and only knowledge about the encoded feature can be provided that way. Since there are needs for high operational efficiency requirements in real-time local feature matching applications, we choose not to incorporate SAM directly into the pipeline. To further leverage the full knowledge from SAM while maintaining high efficiency and inference speed, we treat SAM as a teacher to bootstrap local feature learning, thus using SAM only in the training phase.

C. Semantics in Local Feature Learning

Prior to our work, semantic information had solely been incorporated into the visual localization task as a means to mitigate the challenges posed by low-level local features when dealing with severe image variations. Some early works incorporated semantic segmentation into the visual localization pipeline for filtering matching points [49], [50], improving 2D-3D matching [51], [52], and estimating camera position [53]. Some recent works [21], [54] have attempted to introduce semantics into local feature learning to improve the performance of visual localization. Based on the assumption that high-level semantics are insensitive to photometric and geometric, they enhance the robustness of local descriptors on semantic categories by distilling features or outputs from semantic segmentation networks. However, semantic segmentation tasks can only segment certain specific categories (*e.g.*, visual localization-related street scenes), preventing such approaches from generalizing to open-world scenarios and making them effective only on visual localization tasks. In contrast, we introduce SAM for segmenting any scene as a distillation object and propose the category-agnostic Attention-weighted Semantic Relation Distillation (ASRD) scheme to enable local feature learning to enjoy semantic information in scenes beyond visual localization. In addition, we also propose Weakly Supervised Contrastive Learning Based on Semantic Grouping (WSC) and Edge Attention Guidance (EAG) to further motivate the performance of local features based on the special properties of SAM. Based on the above improvements, our SAMFeat makes it possible for local feature learning to more fully utilize semantic information and benefit in a wider range of scenarios.

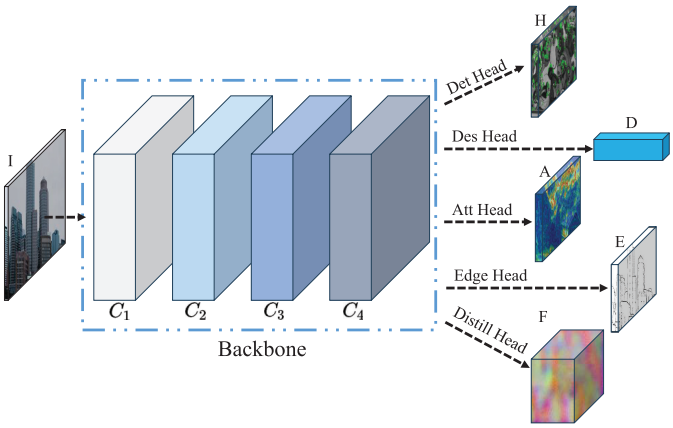


Fig. 2. The overview of our SAMFeat: detection head (Det Head) performs keypoint detection; description head (Des Head) performs feature description; attention head (Att Head) performs attention learning; edge head performs edge depiction, and feature distillation head (Distill Head) performs feature-level distillation. All processes are performed end-to-end.

III. Methodology

A. Overview

In this section, we provide the detailed methodology proposed in this paper. Specifically, in subsection III-B, we first introduce the overall structure of the SAMFeat model: this includes the details of the backbone, three existing heads (keypoint detection head, attention head, feature description head) from the baseline model, and the proposed edge head and the distill head of SAMFeat. Positions, shapes, and relationships between backbones and heads are explained in this subsection. In subsection III-C, we introduce the gifts and knowledge brought by the SAM model and the intuition of why using them in feature learning and matching tasks. In the following subsection III-D, we further introduce, in detail,

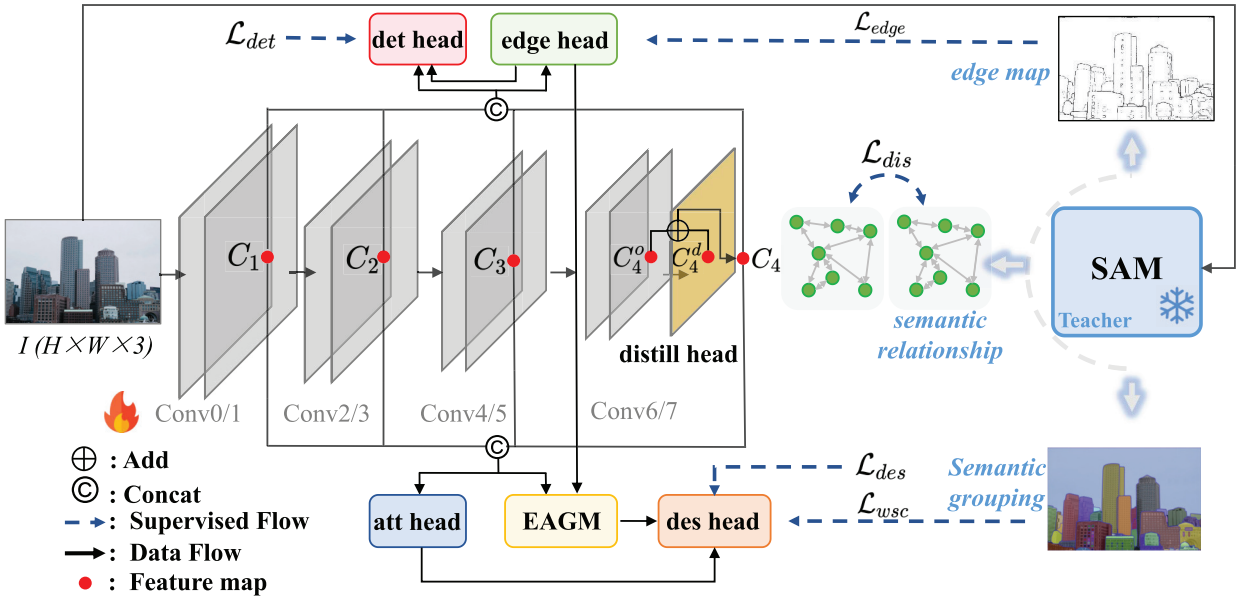


Fig. 3. The detailed overview of SAMFeat. SAMFeat takes an image. I as input, which will be fed into the encoder backbone to generate a multi-scale feature pyramid C_1, C_2, C_3, and C_4. The intermediate features will be used as input to different heads, as described in Sec. III-B, where the detection head (det head) is for keypoint detection; the description head (des head) for feature description; the attention head (att head) performs attention learning; the edge head for edge depiction, and the feature distillation head (distill head) for feature-level distillation. The teacher model SAM provides fruitful knowledge, described in Sec. III-C, including robust features with semantic relationship, edge map, and semantic grouping. This knowledge is used to supervise the training process of SAMFeat via different losses: detection loss $\mathcal{L}_{det}$; edge loss $\mathcal{L}_{edge}$; distillation loss $\mathcal{L}_{dis}$; description loss $\mathcal{L}_{des}$; and Weakly Supervised Contrastive Learning loss $\mathcal{L}_{wsc}$, described in Sec. III-E. Notice that SAM is only applied in the training phase, while there is no computational cost in the inference phase.

how we use them and the logical flow behind them. In the last subsection III-E, we explain how we formulate the previous knowledge from SAM to supervision signals to guide the training of SAMFeat.

B. Overall Structure

Our SAMFeat uses a backbone for feature extraction, along with several heads serving different purposes end-to-end. The simple overview of SAMFeat is shown in Figure 2 and the detailed network structure is shown in Figure 3.

1) Backbone: We use an eight-layer VGG-style backbone encoder following [13] to extract feature maps. The encoder consists of 3×3 convolutional layers, relu layers, and max-pooling layers. For a $H \times W$ image I, we concatenate multiscale feature map outputs on the channel dimension ($C_1 \in \mathbb{R}^{H \times W \times 64}, C_2 \in \mathbb{R}^{\frac{1}{2}H \times \frac{1}{2}W \times 64}, C_3 \in \mathbb{R}^{\frac{1}{4}H \times \frac{1}{4}W \times 128}, C_4 \in \mathbb{R}^{\frac{1}{8}H \times \frac{1}{8}W \times 128}$) by resizing all spatial feature maps to the size $\frac{1}{8}H \times \frac{1}{8}W$, and deliver to different heads that servers different purposes: the detection head (det head) is for keypoint detection by outputting the detected keypoints; the description head (des head) is for feature description by outputting the learned descriptor; the attention head (att head) performs attention learning by outputting the attention heatmap; the edge head is for edge depiction by outputting the learned edge map, and the feature distillation head (distill head) is for feature-level distillation by outputting the learned encoded feature.

2) Keypoint Detection Head: We employ four detection layers to predict keypoint heatmaps at various scales. To integrate these predictions, we upsample the heatmaps to match the image dimensions of $h \times w$. We then use four learnable weights to merge the heatmaps from different scales, thereby predicting the final keypoints and calculating the associated loss. Each detection head receives direct supervision through the detector loss, providing a form of deep supervision as described in [34].

We utilize a weighted binary cross-entropy loss for the detector due to the significant imbalance between keypoints and non-keypoints. With the predicted keypoint heatmap $H \in \mathbb{R}^{h \times w}$ and the pseudo-ground truth label $G \in \mathbb{R}^{h \times w}$, the detector loss $\mathcal{L}_{\text{det}}$ is defined as follows:

$$\mathcal{L}_{\text{bce}}(h, g) = -\lambda g \log(h) - (1 - g) \log(1 - h), \tag{1}$$

$$\mathcal{L}_{\text{det}} = \frac{1}{hw} \sum_{u,v} \mathcal{L}_{\text{bce}}(H_{u,v}, G_{u,v}), \tag{2}$$

where the weight λ is empirically set to 200.

3) Attention Head: The Attention Head is designed to generate the attention map $A \in \mathbb{R}^{\frac{1}{4}h \times \frac{1}{4}w}$. Specifically, we obtain C_{cat} by concatenating the feature maps C_1, C_2, C_3, and C_4 from the backbone network. Next, C_{cat} is averaged across the channel dimension, and an attention map A is generated from this averaged feature map using a 3×3 convolutional layer followed by a softplus activation function.

The attention map proves to be highly useful in descriptor optimization, distillation optimization, and matching processes [55]. In terms of descriptor and distillation optimization, we provide a detailed analyze of motivation in Section III-B.4 and Section III-D. For matching, attention-weighted local descriptors are more effective. Regions with high attention scores in one image are likely to match with similar high-score regions in another image, reducing the matching space and enhancing accuracy. These consistent attention scores

serve as prior information, making local descriptor matching more efficient and reliable. The Attention Head will be jointly optimized during the optimization process of both descriptor generation and feature distillation.

4) Feature Description Head: Initially, we densely extract descriptors for all pixels to form the set $D \in \frac{1}{4}H \times \frac{1}{4}W \times 128$. We then extract descriptors d, a vector with a dimension of 128, for individual pixels at their respective locations. In accordance with previous studies [34], we utilize L2 normalization to obtain the local descriptors, defined as:

$$d = \frac{D_{(i,j)}}{\|D_{(i,j)}\|}, \tag{3}$$

where $\|\cdot\|$ represents the L2 norm and (i, j) denotes the index position of the local descriptor d.

Inspired by [34] and [55], the descriptor loss splits the optimization into two parts: the descriptor's angle and the consistent attention weight. This differs from the standard triplet loss, which only focuses on the angle component. Positive samples have converging attention scores, while negative samples diverge, leading to a consistent distribution of attention scores across image pairs. Higher attention scores significantly influence the gradient, allowing the network to prioritize more relevant samples and avoid optimizing descriptors for less informative pixels, like the sky or grass.

The descriptor Loss is formulated to jointly optimize local descriptors and consistent attention. Building on previous research [34], [55], we use the corresponding point sets (P, P') obtained from ground-truth camera parameters and depths to supervise the training of local descriptors. For an image pair (I, I'), dense descriptors D, D' and attention maps A, A' are extracted using our SAMFeat method.

Given a point set P of size M in image I and the corresponding points P' in image I', the local descriptors of P, P' are denoted by Equation 3 as d_i respectively, where $i \in \{1, \ldots, M\}$. The corresponding score of the descriptor d_i on the attention map A is denoted as ν_i. $\|\cdot\|_2$ denotes the $L2$ distance. Thus, the attention-weighted descriptor is defined as $y_i = \nu_i \cdot d_i$. For y_i, its positive distance $\|y_i\|^+$ is defined as:

$$\|y_i\|^+ = \|\nu_i \cdot d_i - \nu'_i \cdot d'_i\|_2, \tag{4}$$

and its hardest negative distance $\|y_i\|^-$ is defined as:

$$\|y_i\|^- = \min_{j \in \{1,\ldots,M\}, j \neq i} \|\nu_i \cdot d_i - \nu'_j \cdot d'_j\|_2. \tag{5}$$

The overall descriptor loss is the sum of the individual losses:

$$\mathcal{L}_{\text{des}}(y) = \sum_{i=1}^{N} \frac{e^{\nu_i/T}}{\sum_{j=1}^{M} e^{\nu_j/T}} \max(0, \|y_i\|^+ - \|y_i\|^- + 1), \tag{6}$$

where ν is the attention score corresponding to y, and T is a smoothing factor that adjusts the effect of attention weighting on the loss. T is set to 15 following [55].

5) Edge Head: The Edge Head is designed to learn edges, corners, and keypoints information from SAM. The Edge Map E is generated simply via a convolutional and sigmoid layer, taking the concatenated features outputted from the shared backbone. With a simple edge map supervision loss, SAMFeat is able to mimic and generate accurate edge maps learned from SAM edge knowledge as shown in Fig 9. To further utilize the learned edge information, we designed the Edge Attention Guidance Module that enhanced keypoint detection. The corresponding learning loss for Edge Map and the mechanism behind the guidance module will be introduced with details in Section III-D.3

6) Distill Head: The Distill Head is designed to distillate the robust prior feature knowledge from the powerful SAM backbone for better image and scene understanding and recognition. The head is constructed by one convolutional operation, and the learned prior feature will also be fused into the network by another convolutional operation. Supervision loss and feature fusion process will be presented with details in Section III-D.1.

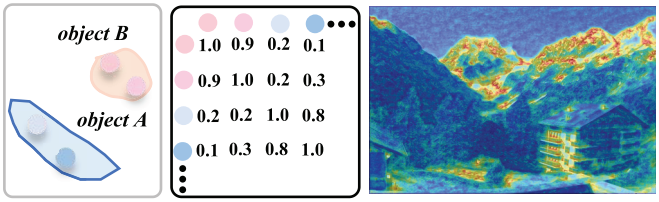
C. Knowledge From SAM

SAM [16] is a newly released visual foundation model for segmenting any objects and has strong zero-shoot generalization due to the fact that it is trained using 11 million images and 1.1 billion masks. Due to its scale, model distillation [56] is deployed in this work. We freeze the weights of SAM and use its output as pseudo-ground truth to guide more accurate and robust local feature learning. In this subsection, we introduce how we utilize three gifts from SAM to enhance our SAMFeat. Shown in Figure 3, we input the image I into the SAM [16] with frozen parameters and then simply proceed to produce the following three outputs for guided local feature learning.

1) Pixel-Wise Representations Relationship: SAM's image encoder trained from 11 million images is used to extract image representations for assigning semantic labels. Thus, SAM's robust zero-shot performance is a helpful enhancement in providing robust and generalized feature representations in complex scenes. The representation of the encoder outputs implies a valuable semantic correspondence, *i.e.*, pixels of the same semantic object are closer together. To eliminate the effect of specific semantic categories on generalizability, we adopt relations between representations as distillation targets. SAM's encoder outputs $\mathcal{F} \in \mathbb{R}^{\frac{1}{8}H \frac{1}{8}W \times C}$, where C is the channel number for feature map. The pixel-wise representations relationship can be defined as $\mathcal{R} \in \mathbb{R}^{\frac{1}{8}H \frac{1}{8}W \times \frac{1}{8}H \frac{1}{8}W}$, where $\mathcal{R}(i, j) = \frac{\mathcal{F}(i) \cdot \mathcal{F}(j)}{|\mathcal{F}(i)||\mathcal{F}(j)|}$.

2) Semantic Grouping: We use the automatically generating masks function[1] of SAM to obtain fine-grained semantic groupings. Specifically, it works by sampling single-point input prompts in a grid over the image, and SAM can predict multiple masks from each of them. Then, masks are filtered for quality and deduplicated using non-maximal suppression [16]. The semantic grouping of the output can be defined as $G \in \mathbb{R}^{H \times W \times N}$, where N is the number of semantic groupings. Notice that semantic grouping differs from semantic segmentation in that each grouping does not correspond to a specific semantic category (*e.g.* buildings, car, and person).

[1] https://github.com/facebookresearch/segment-anything/



(a) Relationship matrix $\mathcal{R}$ from SAM (b) Attention Map

Fig. 4. Schematic diagrams of Relationship matrix and Attention map.

3) Edge Map: The binary edge map $E \in \mathbb{R}^{H\times W\times 1}$ is derived directly[2] from the segmentation results of SAM, which highlights the fine-grained object boundaries.

D. SAMFeat

Thanks to the gifts of the foundation model, SAM, we are able to consider SAM as a knowledgeable teacher with intermediate products and outputs to guide the learning of local features. First, we employ Attention-weighted Semantic Relation Distillation (ASRD) to distill the category-agnostic semantic relations in the SAM encoder into SAMFeat, thereby enhancing the expressive power of local features by introducing semantic distinctiveness. Second, we utilize the high-level semantic grouping of SAM outputs to construct Weakly Supervised Contrastive Learning Based on Semantic Grouping (WSC), which provides cheap and valuable supervision for local descriptor learning. Third, we design an Edge Attention Guidance (EAG) to utilize the low-level edge structure discovered by SAM to guide the network to pay more attention to these edge regions, which are more likely to be detected as keypoints and rich in discriminative information during local feature detection and description.

1) Attention-Weighted Semantic Relation Distillation: SAM aims to obtain the corresponding semantic masks based on the prompt, so the encoder output representation of SAM is rich in semantic discriminative information. Unlike semantic segmentation, SAM does not project pixels to a specified semantic category, so we resort to distilling the semantics contained in the encoder by exploiting the relative relationship between pixels (*i.e.*, pixel representations of the same object are closer together).

However, not every pixel is equally important for local feature learning, and forcing the network to learn a large proportion of background pixels (*e.g.*, the sky) can hinder the learning of discriminative foreground regions. We therefore advocate a greater focus on discriminative foreground regions in the distillation process. In contrast to classical relational distillation [57], we propose Attention-weighted Semantic Relation Distillation (ASRD) to guide the distillation process to focus on valuable relational pairs to further motivate the transfer of knowledge that facilitates local feature matching.

Specifically, C_4^o is exported from *Conv7* layer and then imported into the distillation head to get $C_4^d \in \mathbb{R}^{\frac{1}{8}H\times\frac{1}{8}W\times 128}$. Following the operations reported in Sec. III-C, the semantic relation matrix of C_4^d can be defined as $\mathcal{R}'$. As shown in Figure 4, we distill the semantic relation matrix by imposing L1 loss in order to obtain semantic discriminativeness for C_4^d. $\mathcal{R}'$ and $\mathcal{R}$ are the corresponding student (SAMFeat) and teacher (SAM) relation matrix. As shown in Figure 4 (b), we use the attention map obtained in Section III-B.3 to construct the weight matrix used to weight the relational distillations. We flatten the attention map into a 1-dimensional vector V_A, and then multiply V_A and the transpose V_A^T of V_A by matrix multiplication to obtain the weight matrix $\mathcal{W}$, which can be formally defined as follows:

$$\mathcal{W} = V_A \times V_A^T, \tag{7}$$

where the elements of $\mathcal{W} \in \mathbb{R}^{\frac{1}{8}H\times\frac{1}{8}W}$ and $\mathcal{R}$ correspond to each other. Attention-weighted Semantic Relation Distillation

[2]https://github.com/ymgw55/segment-anything-edge-detection

Algorithm 1 Attention-Weighted Semantic Relation Distillation

Input: Image pair I_1, I_2; Attention Map A; $H = W = 400$; $C = 256$; SAMFeat's encoder E; SAM's encoder E'.

Output: SAMFeat's Relationship Matrix $\mathcal{R}$; SAM's Relationship Matrix $\mathcal{R}'$.

Given I_1, I_2, an encoded image feature $\mathcal{F} \in \mathbb{R}^{\frac{1}{8}H\times\frac{1}{8}W\times C}$ can be obtained via E.

Given I_1, I_2, an encoded image feature $\mathcal{F}' \in \mathbb{R}^{64\times 64\times C}$ can be obtained via E'.

Downsample $\mathcal{F}'$ to $\mathcal{F}'_{down} \in \mathbb{R}^{\frac{1}{8}H\frac{1}{8}W\times C}$

Downsample $\mathcal{A}$ to $\mathcal{A}_{down} \in \mathbb{R}^{\frac{1}{8}H\times\frac{1}{8}W}$, and apply cross multiplication with itself and apply softmax activation function to obtain the Attention Weight $A_w \in \mathbb{R}^{\frac{1}{8}H\frac{1}{8}W\times\frac{1}{8}H\frac{1}{8}W}$

Flatten $\mathcal{F}$ and $\mathcal{F}'_{down}$ then calculate mean on $dim = C$ to obtain $\mathcal{F}_{flatten} \in \mathbb{R}^{\frac{1}{8}H\frac{1}{8}W}$ and $\mathcal{F}'_{flatten} \in \mathbb{R}^{\frac{1}{8}H\frac{1}{8}W}$

Construct Attention Weighted Relationship Matrix $\mathcal{R} \in \mathbb{R}^{\frac{1}{8}H\frac{1}{8}W\times\frac{1}{8}H\frac{1}{8}W}$ and $\mathcal{R}' \in \mathbb{R}^{\frac{1}{8}H\frac{1}{8}W\times\frac{1}{8}H\frac{1}{8}W}$ from $\mathcal{F}_{flatten}$ and $\mathcal{F}'_{flatten}$ respectively, where $\mathcal{R}(i,j) = \frac{\mathcal{F}(i)\cdot\mathcal{F}(j)}{|\mathcal{F}(i)||\mathcal{F}(j)|}\cdot A_w(i,j)$, and same applied for $\mathcal{R}'$

return $\mathcal{L}_{dis} = |\mathcal{R} - \mathcal{R}'|$.

Fig. 5. Example of Semantic Grouping. Different colored stars represent sampling points in different semantic groupings.

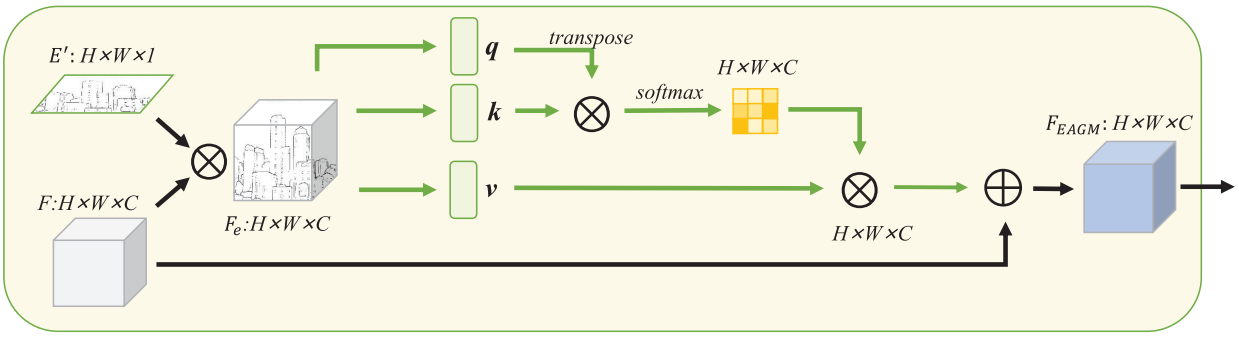


Fig. 6. The Edge Attention Guidance Module (EAGM) takes as input the predicted edge map. E' from the edge head and the fused multi-scale feature maps F extracted from the backbone. We first fuse E' and F through a pixel-wise dot product to generate an edge-oriented feature map F_e. Next, we apply various convolutional transformations to F_e to derive the query q, key k, and value v. The attention score is then computed by taking the dot product of the query and key, followed by applying the *softmax* function to obtain the attention weights. These weights are used to compute the edge-enhanced feature maps by combining them with the value feature vector. Finally, the edge-enhanced feature maps are added to F to produce the output feature maps F_{EAGM}.

Loss $\mathcal{L}_{dis}$ can be defined as:

$$\mathcal{L}_{dis} = \frac{\sum_{i,j}^{(\frac{1}{8}H\times\frac{1}{8}W),(\frac{1}{8}H\times\frac{1}{8}W)} |\mathcal{R}_{i,j} - \mathcal{R}'_{i,j}| \cdot e^{W(i,j)}}{\sum_{i,j} e^{W(i,j)}}, \tag{8}$$

where N is the number of matrix elements, *i.e.*, $\left(\frac{1}{8}H \times \frac{1}{8}W\right) \times \left(\frac{1}{8}H \times \frac{1}{8}W\right)$. Since ASRD is category-agnostic, it is possible to generalize local feature distillation semantic information to generic scenarios. A detailed pseudo-code is illustrated in algorithm 1.

2) Weakly Supervised Contrastive Learning Based on Semantic Grouping: As shown in Figure 5, we use semantic groupings derived from SAM to construct weakly supervised contrastive learning to optimize the description space of local features. Our motivation is very intuitive: *i.e.*, pixels belonging to the same semantic grouping should be closer in the description space, and on the contrary pixels of different groupings should be kept at a distance in the description space. However, since two pixels belonging to the same grouping do not imply that their descriptors are the closest pair, forcing them to align will impair the discriminative properties of pixels within the same grouping. Therefore, semantic grouping can only provide weakly supervised constraints, and we maintain the discriminatory nature within the semantic grouping by setting a margin in optimization. Given the sampling points set $P \in \mathbb{R}^N$, the positive sample average distance $Dist_{pos}$ can be defined as:

$$Dist_{pos} = \frac{1}{J}\sum_{i,j}^{J} \mathrm{dis}(P_i, P_j), where\ G(i) = G(j)\ and\ i \neq j. \tag{9}$$

Here $\mathrm{dis}(\mathrm{P_i}, \mathrm{P_j})$ means calculate the Euclidean distance between the local descriptors corresponding to the two sampling points P_i and P_j. $G(\cdot)$ denotes the indexed semantic grouping category. J denotes the number of positive samples, noting that since J is not consistent for each image, we take the average to denote the positive sample distance. Similarly, the negative sample average distance $Dist_{neg}$ can be defined as:

$$Dist_{\mathrm{neg}} = \frac{1}{K}\sum_{i,j}^{K} \mathrm{dis}(\mathrm{P_i}, \mathrm{P_j}),$$
$$\text{where } G(i) \neq G(j). \tag{10}$$

where K denotes the number of negative samples. Thus, the final $\mathcal{L}_{wsc}$ loss can be defined as:

$$\mathcal{L}_{wsc} = -\log\left(\frac{\exp(\max(Dist_{pos}, \mathrm{M})/\mathrm{T})}{\exp(\max(Dist_{pos}, \mathrm{M}) + Dist_{neg})/\mathrm{T})}\right), \tag{11}$$

where M is a margin parameter used to protect distinctiveness within semantic groupings, and T means the temperature coefficient.

3) Edge Attention Guidance: Edge regions are more worthy of the network's attention than mundane regions. On one hand, corner and edge points in the edge region are more likely to be detected as keypoints. On the other hand, the edge region contains rich information about the geometric structure thus contributing more to the discriminative nature of the local descriptor. To enable the network to better capture the details of edge areas and improve the robustness of descriptors, we propose, as shown in Figure 6, the Edge Attention Guidance Module (EAG), which can guide the network to focus on edge regions. As shown in Figure 3, we first set up an edge head to predict the edge map E' and use the SAM output of the edge map for supervision. The edge loss $\mathcal{L}_{edge}$ is denoted as:

$$\mathcal{L}_{edge} = \sum_{i}^{H\times W} |E_i - E'_i|. \tag{12}$$

We then fuse the predicted edge map E' into the local feature detection and description pipeline to bootstrap the network.

Figure 9 visualize the learning outcome of object boundaries under the guidance of SAM. With our EAG, SAMFeat learns accurate boundaries and edges effectively and efficiently. This prior knowledge of boundaries and edges will then aid feature detection and description.

E. Guidance From SAM

1) Learning Gifts From SAM: As described in section III-C, The carefully designed architecture enables SAMFeat to output the learned knowledge under the supervision of three designed loss modules. Therefore, we could summarize the guidance loss supervision from SAM $\mathcal{L}_g$ as follows:

$$\mathcal{L}_g = \mathcal{L}_{dis} + \mathcal{L}_{edge} + \mathcal{L}_{wsc}. \tag{13}$$

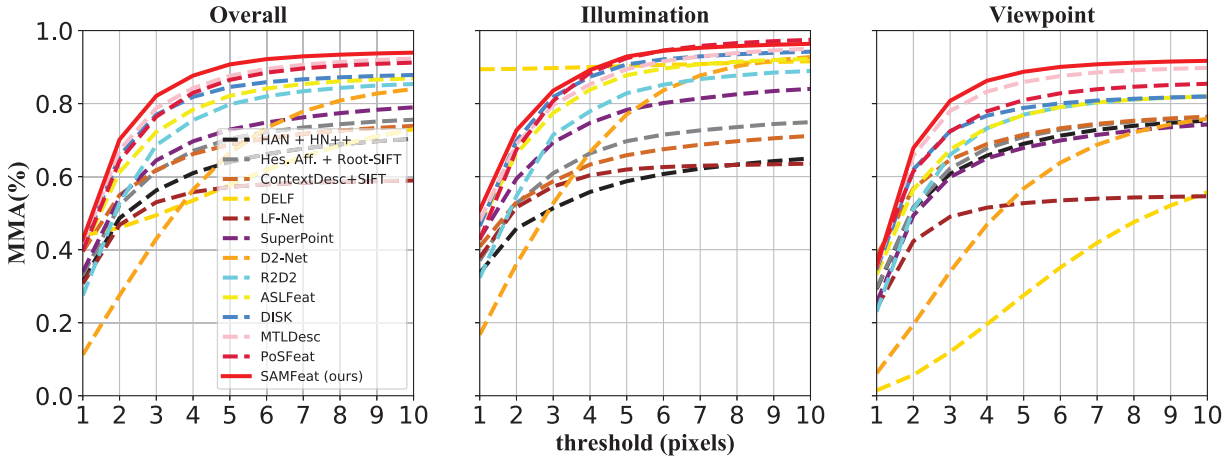


Fig. 7. Comparisons on HPatches dataset with different thresholds Mean Matching Accuracy. Our SAMFeat achieves higher average local feature matching accuracy than other state-of-the-art methods at all thresholds.

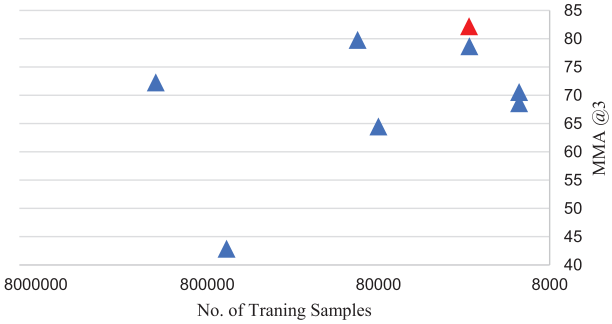


Fig. 8. Comparisons on the number of Training Samples. SAMFeat utilized relatively small training samples while achieving the highest performance.

Fig. 9. Left: Random images selected from HPatches. Right: Learned edge boundaries from SAMFeat under the guide of SAM.

With accurate loss supervision, SAMFeat is able to utilize the learned knowledge in later modules discussed in section III-D to further aid feature learning and matching tasks.

2) Guided Local Feature Detection: To aid feature detection in SAMFeat, the predicted edge map E' from the edge head is performed with a pixel-wise dot product with the middle-level encoded feature representation C_3, shown in Figure 3. The product is added to C_3 for a residual purpose to obtain an edge-enhanced feature C_3. This feature will be used to generate a heatmap via the detection head to provide better local feature detection.

3) Guided Local Feature Description: We filter the edge features by the predicted edge map and model the features of the edge region by a self-attention mechanism to encourage the network to capture the information of the edge region. Specifically, the predicted edge map E' from the edge head, and the multi-scale feature maps F extracted from the backbone are fed into the Edge Attention Guidance Module. As shown in Figure 6, we first fuse E' and F by applying a pixel-wise dot product to obtain an edge-oriented feature map F_e. Then we apply different convolutional transformations to the given F_e to get query q, key k, and value v respectively. We then calculate the attention score using the dot product between query and key. Next, we use the *softmax* function on the attention score to obtain the attention weight, which is used to calculate the edge-enhanced feature maps with the value feature vector. Finally, the edge-enhanced feature maps and the F are added to obtain the output feature maps F_{EAGM}.

4) Total Loss: The total loss $\mathcal{L}$ can be defined as:

$$\mathcal{L} = \mathcal{L}_g + \mathcal{L}_{det} + \mathcal{L}_{des} \tag{14}$$

$\mathcal{L}_g$ is the loss function guided by SAM's knowledge defined in section III-E, while $\mathcal{L}_{det}$ is the cross entropy loss for supervised keypoint detection and $\mathcal{L}_{des}$ is the attention weighted triplet loss from MTLDesc [34] for optimizing the local descriptors. Individual weights for each loss are not assigned: each loss shares equal weights. This independence from hyper-parameters, again, shows the robustness of SAMFeat.

IV. Experiments

A. Implementation

To generate our training data with dense pixel-wise correspondences, we rely on the MegaDepth dataset [15], a rich resource containing image pairs with known pose and depth information from 196 diverse scenes. Specifically, we use MTLDesc [34] [3] released megedepth image and the correspondence ground truth for training. In our experiment, we meticulously configured the parameters to establish a consistent and efficient training process. Hyper-parameters are set as follows. The learning rate of 0.001 enables gradual parameter updates, and the weight decay of 0.0001 helps control model complexity and mitigate overfitting. With a batch size of 14, our model processes 14 samples per iteration, striking a balance between computational efficiency and convergence. M and T are set to 0.07 and 5. Training spans 30 epochs to ensure comprehensive exposure to the data, with a total training time of 3.5 hours. By meticulously defining these parameters and configurations, we establish a robust experimental setup that ensures replicability and accurate evaluation of our model's performance. More detailed information about parameter tuning and ablation experiments can be found in the later subsections.

B. Image Matching

We evaluate the performance of our method in the image-matching and homography estimation tasks on the most popular feature learning and matching benchmark: HPatches [1]. The HPatches dataset consists of 116 sequences of image patches extracted from a diverse range of scenes and objects. Each image patch is associated with ground truth annotations, including key point locations, descriptors, and corresponding homographies. We follow the same evaluation protocol for image matching evaluation as in D2-Net [32], where eight unreliable scenes are excluded. To ensure an equitable comparison, we align the features extracted by each method through nearest-neighbor matching. A match is deemed accurate if its estimated reprojection error is lower than a predetermined matching threshold. The threshold is systematically varied from 1 to 10 pixels, and the mean matching accuracy (MMA) across all pairs is recorded, indicating the proportion of correct matches relative to potential matches. Subsequently, the area under the curve (AUC) is computed at 5px based on the MMA. The comparison between SAMFeat and other state-of-the-art methods on HPatches image matching is visualized in Figure 7. The MMA @3, the area under the curve at thresholds of 2 and 3 against other state-of-the-art methods under each threshold is listed in the left part in Table I. Homography estimation results on HPatches [1] are shown in the right part in Table I. To ensure a fair comparison, we follow the evaluation protocol of [37] and [67], where we compute the corner error between the images warped by the estimated homography $\hat{\mathcal{H}}$ and the ground truth homography $\mathcal{H}$ as a measure of correctness, as outlined in Table I. Following the protocol in [37] and [67], we report the area under the cumulative curve (AUC) of the corner error for a threshold value of 1 and 3, with the number of top matches. SAMFeat achieved the highest score in various metrics and tasks compared to the most updated feature learning model in top-tier conferences.

TABLE I

Performance Comparison on HPatches Dataset and Homography Estimation. The Table Includes Image Matching Performance (MMA @3, AUC @2, AUC @5), and Homography Estimation Results (AUC @1, AUC @3, Number of Matches)

Image Matching			
Methods	**MMA @3**	**AUC @2**	**AUC @5**
SIFT $_{IJCV'2012}$ [59]	50.1	39.5	49.6
HardNet $_{NeurIPS'2017}$ [60]	62.1	42.6	56.9
DELF $_{ICCV'2017}$ [61]	50.7	44.7	48.7
SuperPoint $_{CVPRW'2018}$ [13]	65.7	44.1	59.0
Lf-net $_{NeurIPS'2018}$ [62]	53.2	38.7	48.7
ContextDesc $_{CVPR'2019}$ [28]	63.2	47.2	58.3
D2Net $_{CVPR'2019}$ [63]	40.3	19.5	37.8
R2D2 $_{NeurIPS'2019}$ [64]	72.1	43.4	64.2
DISK $_{NeurIPS'2020}$ [14]	72.2	52.3	69.8
ASLFeat $_{CVPR'2020}$ [65]	72.2	50.1	66.9
LLF $_{WACV'2021}$ [66]	74.0	52.1	66.8
Key.Net $_{TPAMI'2022}$ [67]	72.1	40.9	56.0
ALIKE $_{TMM'2022}$ [68]	70.5	51.7	69.0
MTLDesc $_{AAAI'2022}$ [35]	78.7	55.0	71.4
PoSFeat $_{CVPR'2022}$ [69]	75.3	50.2	69.2
SFD2 $_{CVPR'2023}$ [21]	70.6	50.1	64.8
TPR $_{CVPR'2023}$ [70]	79.8	57.1	73.0
SAMFeat (Ours)	**82.2**	**59.2**	**74.4**

Homography Estimation			
Methods	**AUC @1**	**AUC @3**	**# Matches**
SuperPoint $_{CVPRW'2018}$ [13]	26.3	52.2	1.1K
D2Net $_{CVPR'2019}$ [63]	-	23.2	0.2K
R2D2 $_{NeurIPS'2019}$ [64]	20.4	50.6	0.5K
DISK $_{NeurIPS'2020}$ [14]	22.1	52.2	1.1K
SuperGlue $_{CVPR'2020}$ [36]	-	53.9	0.6K
SiLK $_{CVPR'2022}$ [71]	32.0	58.1	1K
XFeat $_{CVPR'2023}$ [72]	34.5	58.5	1K
SAMFeat (Ours)	**34.9**	**58.8**	1K

C. Visual Localization

To further validate the efficacy of our approach when dealing with intricate tasks, we assess its performance in the area of visual localization. This task involves estimating the camera's position within a scene using an image sequence and serves as an evaluation benchmark for local feature performance in long-term localization scenarios, without requiring a dedicated localization pipeline. We utilize the Aachen Day-Night v1.1 dataset [4] to showcase the impact on visual localization tasks. To ensure fairness in the assessment, we employ a predefined visual localization pipeline[4] based on

[3] https://github.com/vignywang/MTLDesc

[4] https://github.com/GrumpyZhou/image-matching-toolbox

TABLE II

VISUAL LOCALIZATION PERFORMANCE COMPARISON ON AACHEN V1.1. CATEGORY "L" MEANS LOCAL FEATURE METHODS SPECIFICALLY DESIGNED FOR VISUAL LOCALIZATION TASKS, AND "G" MEANS GENERALIZED LOCAL FEATURE METHODS

Category	Method	Accuracy @ Thresholds (%) ↑	
		Day	Night
		0.25m,2°/0.5m,5°/5m,10°	
L	SeLF $_{TIP'22}$ [55]	–	75.0 / 86.8 / 97.6
	SFD2 $_{CVPR'2023}$ [21]	88.2 / 96.0 / 98.7	78.0 / 92.1 / 99.5
G	SIFT $_{IJCV'12}$ [6]	72.2 / 78.4 / 81.7	19.4 / 23.0 / 27.2
	SuperPoint $_{CVPRW'18}$ [31]	87.9 / 93.6 / 96.8	70.2 / 84.8 / 93.7
	D2-Net $_{CVPR'19}$ [33]	84.1 / 91.0 / 95.5	63.4 / 83.8 / 92.1
	R2D2 $_{NeurIPS'19}$ [32]	88.8 / 95.3 / 97.8	72.3 / 88.5 / 94.2
	ASLFeat $_{CVPR'20}$ [34]	88.0 / 95.4 / 98.2	70.7 / 84.3 / 94.2
	CAPS $_{ECCV'20}$ [74]	82.4 / 91.3 / 95.9	61.3 / 83.8 / 95.3
	LISRD $_{ECCV'20}$ [75]	–	73.3/ 86.9 / 97.9
	DISK $_{NeurIPS'22}$ [14]	–	73.8 / 86.2 / 97.4
	PoSFeat $_{CVPR'22}$ [69]	–	73.8 / 87.4 / **98.4**
	MTLDesc $_{AAAI'22}$ [35]	–	74.3 / 86.9 / 96.9
	TR $_{CVPR'23}$ [70]	–	74.3 / 89.0 / **98.4**
	SAMFeat (Ours)	**90.2** / **96.0** / **98.5**	**75.9** / **89.5** / 95.8

colmap provided by benchmark.[5] This pipeline operates as follows: Initially, custom features extracted from the database's images are employed to construct a structure-from-motion model. Subsequently, the query images are registered within this model using the same custom features. For keypoints matching, we utilize the mutual nearest neighbor approach to effectively filter out outliers. And we set the number of features in this experiment to 10000. We tally the number of accurately localized images under three distinct error thresholds, namely (0.25m, 2°), (0.5m, 5°), and (5m, 10°), signifying the maximum allowable position error in both meters and degrees. Referring to Table II, we categorize current state-of-the-art methods into two categories: *G* contains methods that are designed for general feature learning tasks; *L* contains methods that are designed, tuned, and tested specifically for localization tasks, and they typically perform poorly outside of specific localization scenarios, as shown in Table I. SAMFeat achieved the top performance among all general methods, while also revealing a competitive performance among methods that are designed specifically for visualization.

D. 3D Reconstruction

We utilize the ETH benchmark [71] to evaluate the performance of our method on the 3D reconstruction task. Three medium-scale datasets from the benchmark are used for this purpose. We perform exhaustive image matching on all three collections and apply ratio test filtering with a default threshold of 0.8 to eliminate incorrect matches. The reconstruction protocol follows the COLMAP [2] pipeline, which involves running Structure-from-Motion (SfM) first and then Multi-View Stereo (MVS) to generate the dense point cloud models. As shown in Table III, SAMFeat consistently produces the highest number of registered images and sparse points, along with competitive track length and reprojection error, demonstrating its robustness and accuracy in 3D reconstruction tasks.

[5]https://www.visuallocalization.net

TABLE III

EVALUATION ON ETH 3D RECONSTRUCTION BENCHMARK [71]. THE TOP TWO RESULTS ARE MARKED WITH BOLD (1ST) AND UNDERLINE (2ND)

ETH benchmark						
Datasets	Methods	#Reg. Images	#Sparse Points	Track Length	Reproj. Error	#Dense Points
Madrid Metropolis 1344 images	SuperPoint [31]	438	29K	9.03	1.02px	1.55M
	D2-Net [33]	495	144k	6.39	1.35px	1.46M
	ASLFeat [34]	613	96k	8.76	0.90px	**2.00M**
	DISK [14]	677	213K	7.89	1.14px	1.87M
	AWDesc-CA [56]	864	278K	9.52	**0.96px**	1.65M
	SAMFeat	**892**	**282K**	**9.84**	0.93px	1.90M
Gendarmenmarkt 1463 images	SuperPoint [31]	967	93k	7.22	1.03px	3.81M
	D2-Net [33]	965	310K	5.55	1.28px	3.15M
	ASLFeat [34]	1040	221K	**8.72**	1.00px	**4.01M**
	DISK [14]	1218	588K	6.02	0.98px	3.62M
	AWDesc-CA [56]	1354	548K	6.94	0.95px	3.86M
	SAMFeat	**1370**	**596K**	7.02	**0.93px**	3.91M
Tower of London 1576 images	SuperPoint [31]	681	52K	8.76	0.96px	2.77M
	D2-Net [33]	708	287K	5.20	1.34px	2.86M
	ASLFeat [34]	821	222K	12.52	0.92px	**3.06M**
	DISK [14]	985	517K	5.90	1.02px	3.00M
	AWDesc-CA [56]	1414	563K	12.88	0.88px	2.89M
	SAMFeat	**1443**	**587K**	**13.01**	**0.84px**	2.91M

TABLE IV

RELATIVE POSE ESTIMATION ACCURACY ON THE SCANNET DATASET. LEFT: ACCURACY OF ROTATION ESTIMATION; RIGHT: ACCURACY OF TRANSLATION ESTIMATION. THE TERM d_{frame} REPRESENTS THE INTERVAL BETWEEN FRAMES, WITH LARGER FRAME INTERVALS CORRESPONDING TO MORE CHALLENGING IMAGE PAIRS FOR MATCHING

Average Accuracy on ScanNet (%)			
Methods	d_{frame}=10	d_{frame}=30	d_{frame}=60
LF-Net [30]	91.0 15.2	57.7 11.6	42.5 12.8
SuperPoint [13]	92.8 14.7	58.3 13.4	41.2 15.8
SAMFeat (Ours)	**98.5 21.2**	**68.8 24.0**	**51.0 23.1**

E. Pose Estimation

We further evaluate the performance of our approach on the ScanNet [72] pose estimation benchmark. The ScanNet dataset is a large-scale indoor dataset containing ground truth poses and depth images, with well-defined splits for training, validation, and testing across 1613 monocular sequences. We followed the approach of [30] and randomly sampled 1,000 image pairs from three different frame intervals (10, 30, and 60). We assess the estimated camera poses using angular deviation for both rotation and translation. Specifically, we consider a rotation or translation to be correct if the angular deviation is below 5° and report the average accuracy.

F. Ablation Study

We conduct ablation studies on different aspects to support our claim and illustrate the necessity of each of our contributed modules.

1) Ablation on Designed Modules: Table V demonstrates the efficacy of the components within our network as we progressively incorporate Attention-weighted Semantic Relation Distillation (ASRD), Weakly Supervised Contrastive Learning Based on Semantic Grouping (WSC), and Edge Attention Guidance (EAG). The effectiveness of each component is

TABLE V

DETAILED ABLATION STUDY ON SAMFEAT. ✓ DENOTES APPLIED COMPONENTS. THE RESULTS OF MMA@2 AND MMA@3 ON HPATCHES, WITH THE REMOVAL OF EACH COMPONENT INDIVIDUALLY AND THE SEQUENTIAL APPLICATION OF COMPONENTS, ARE REPORTED

ASRD	EAG	WSC	MMA @2	MMA @3
			63.5	75.7
✓			68.1	78.6
✓	✓		68.9	80.9
✓	✓	✓	**70.4**	**82.2**
✓		✓	69.3	81.2
	✓	✓	69.1	79.4

TABLE VI

ABLATION TEST ON THE ATTENTION-WEIGHTED SEMANTIC RELATION DISTILLATION (ASRD). WE COMPARE THE EFFECT OF OUR PROPOSED ASRD MODULE AGAINST DIRECT SEMANTIC FEATURE DISTILLATION FROM SAM (DSFD) [56], THE CLASSICAL RELATIONAL KNOWLEDGE DISTILLATION (RKD) [57], AND DIRECT FEATURE DISTILLATION FROM DINO-V2 LARGE (DINO) [48]. OUR PROPOSED APPROACH BROUGHT THE MOST POSITIVE EFFECT ON MATCHING ACCURACY

Module Selected	MMA @3
DSFD [57]	76.9
RKD [58]	77.4
DINO [49]	78.1
ASRD (Ours)	**78.6**

reflected by the Mean Matching Accuracy at the pixel three threshold on the HPatches Image Matching task. Our baseline is trained using SuperPoint [13] structure along with its detector supervision and attention-weighted triplet loss [34] for descriptor learning. Following the addition of the ASRD, the model's performance notably improves due to better image feature learning. The introduction of the WSC further enhances accuracy by augmenting the discriminative power of descriptors with semantics. It demonstrates superior performance as it better preserves the inner diversity of objects by optimizing sample ranks. Lastly, the inclusion of the EAG bolsters the network's capability to embed object edge and boundary information, resulting in further enhancements in accuracy.

2) Attention-Weighted Semantic Relation Distillation Versus Direct Semantic Feature Distillation: Table VI highlights the performance differences between two approaches for distilling image features from the SAM encoder: our proposed Attention-weighted Semantic Relation Distillation (ASRD) and Direct Semantic Feature Distillation (DSFD). The effectiveness of these approaches is evaluated using the Mean Matching Accuracy at the pixel three threshold on the HPatches Image Matching task. The results demonstrate that while DSFD achieves a Mean Matching Accuracy (MMA) of 76.9, the ASRD method significantly enhances performance, achieving an MMA of 78.6. This indicates that ASRD provides superior feature learning by effectively capturing and distilling the semantic relationships within the image features.

TABLE VII

ABLATION TEST ON HYPER-PARAMETERS ON WSC

(M, T)	MMA @3
0, 1	80.9
0.03, 1	81.2
0.05, 1	80.3
0.07, 1	**81.3**
0.09, 1	80.8
0.11, 1	80.5
0.07, 3	81.6
0.07, 5	**82.2**
0.07, 7	82.0
0.07, 9	81.8

TABLE VIII

A QUANTITATIVE ANALYSIS ON THE OVERHEAD TRAINING TIME COSTS FOR ADDING THE EXTRA LOSS FUNCTIONS. ✓ MEANS DENOTES APPLIED LOSS COMPONENTS

ASRD	EAG	WSC	Training time in Hours
			3.6
✓			4.7
✓	✓		5.1
✓	✓	✓	6.0

3) Hyper-Parameters in Weakly Supervised Contrastive Module: Table VII presents the impact of different hyper-parameter values on the performance of Weakly Supervised Contrastive Learning Based on Semantic Grouping (WSC). Specifically, the margin parameter M is used to maintain distinctiveness within semantic groupings, while T represents the temperature coefficient. The effectiveness of these hyper-parameters is evaluated using the Mean Matching Accuracy at the pixel three threshold on the HPatches Image Matching task. The results demonstrate that varying M and T values impact performance. Notably, the combination of $M = 0.07$ and $T = 5$ achieves the highest accuracy, with an MMA of 82.2. This configuration provides an optimal balance, enhancing the discriminative power of the descriptors by effectively preserving the inner diversity of objects.

4) Training Samples: Table 8 provides a comparative analysis of various methods based on the number of training samples used and their Mean Matching Accuracy (MMA) at the pixel three threshold on the HPatches dataset. Despite being trained on a relatively small dataset of 23,600 images, SAMFeat achieves an outstanding MMA@3 of 82.2, surpassing all other methods. This result underscores the efficiency and robustness of SAMFeat in achieving superior performance with limited training data. For instance, SuperPoint, trained on 80,000 images, achieves an MMA@3 of 64.5, while ASLFeat, utilizing a significantly larger dataset of 1,600,000 images, achieves an MMA@3 of 72.3. Similarly, methods like D2Net and R2D2, despite having access to larger or comparable training datasets, attain lower MMA@3 scores of 42.9 and 68.6, respectively. These comparisons highlight

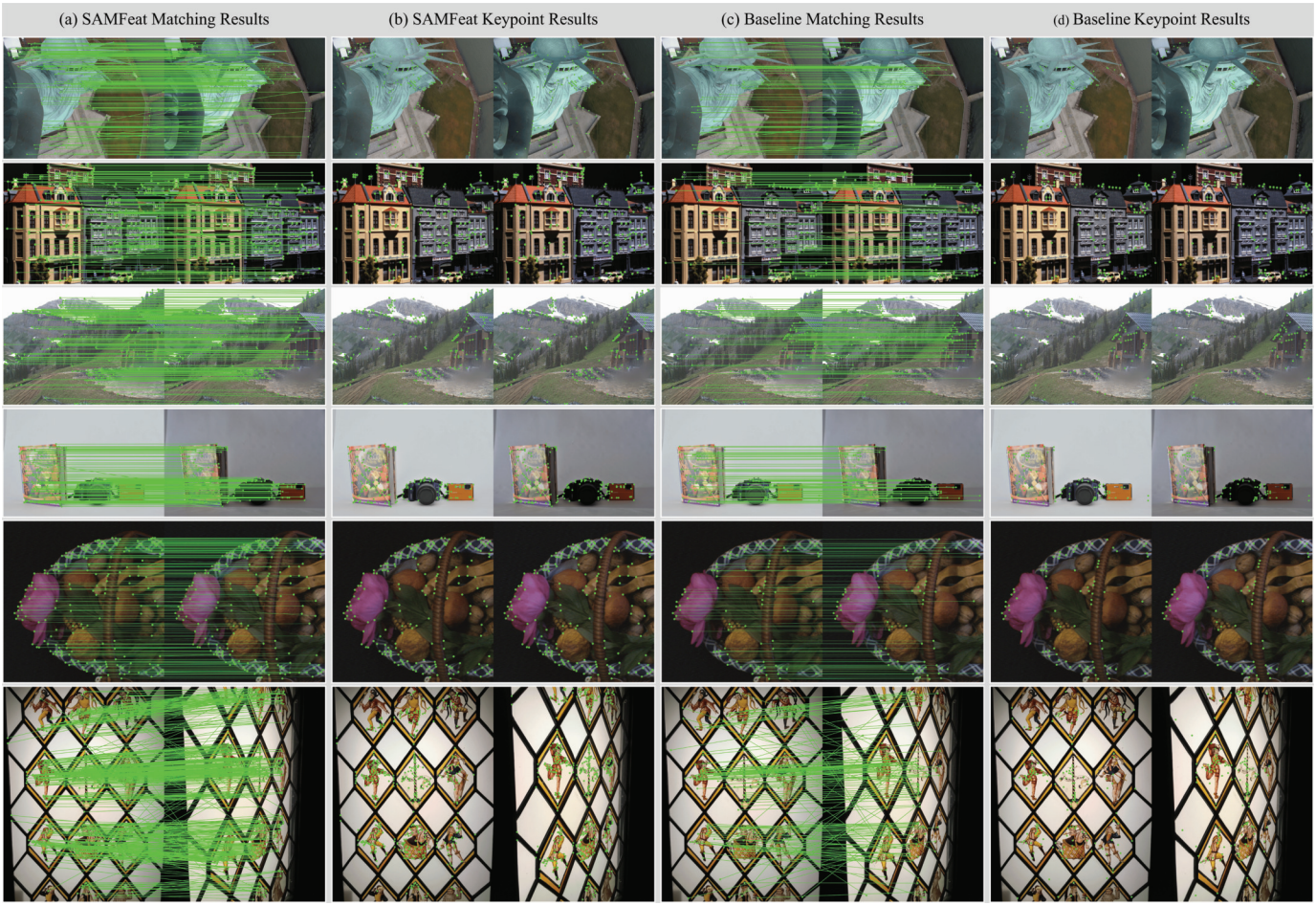


Fig. 10. A visualization of feature matching (a) and keypoint detection (b) results, and comparison against the baseline approach (c and d). Learned and inspired by SAM's generalized feature and semantic grouping, our approach is able to learn and pair more valid matches; enhanced by SAM's edge knowledge, our approach is also able to detect more valid keypoints along object boundaries, which further helps feature matching. Methods are compared against the same settings and hyper-parameters.

the effectiveness of SAMFeat's design in leveraging a modest amount of training data to achieve top-tier performance.

5) Training Time: Table VIII presents a quantitative analysis of the overhead training time costs associated with incorporating additional loss functions into our method. The table details the incremental training time required for each combination of the Attention-weighted Semantic Relation Distillation (ASRD), Edge Attention Guidance (EAG), and Weakly Supervised Contrastive Learning Based on Semantic Grouping (WSC) loss components. Note that our method only requires training for 6 hours using two Nvidia RTX 3090 GPUs. Compared to other work like ASLFeat (42 hours on a single NVIDIA RTX 2080Ti) and TRR (30 hours for training with two NVIDIA-A100 GPUs), this demonstrates a totally reproducible cost for individual researchers.

Even though each additional loss function introduces some extra training time, the tradeoff is justified by the minimal increase in time and the corresponding improvement in accuracy, and the final training time remains acceptable. In addition, our approach remains fast and lightweight in deployment. The size of the model checkpoint is only 8 MB, the training inference time is 21 FPS, and the GPU memory requirement is around 20 GB for training on Nvidia RTX 3090 GPU. These demonstrate the lightweight nature of our approach. The high efficiency makes our method easily implementable and resource-efficient, highlighting its reproducibility and practicality for individual researchers in the field of feature learning and description.

TABLE IX

ABLATION TEST ON ADJUSTING THE LOSS WEIGHT OF EAG WITHOUT WSC. THE MMA @3 ON HPATCHES ARE RECORDED, SHOWING THAT IT IS DIFFICULT TO ACHIEVE THE EFFECT OF IMPOSING WSC BY ONLY ADJUSTING THE LOSS WEIGHTS

Weights of ASRD	Weights of EAG	MMA @3
1.0	0.5	80.7
1.0	1.0	80.9
1.0	1.5	81.0

6) Ablation on Loss Weights: Table IX presents the results of an ablation test that explores the impact of adjusting the loss weight of Edge Attention Guidance (EAG) without incorporating Weakly Supervised Contrastive Learning Based on Semantic Grouping (WSC). The effectiveness of these adjustments is measured using the Mean Matching Accuracy

TABLE X

COMPARISONS ON THE INFERENCE SPEED. THE SPEED IS CALCULATED AS THE AVERAGE FEATURE EXTRACTION INFERENCE SPEED ON HPATCHES (480 × 640) WITH THE SAME SETTING. ALL HEADS ARE ACTIVE DURING INFERENCE SPEED TESTING FOR OUR APPROACH

Methods	Superpoint	D2-Net	SFD2	MTLDesc	R2D2	SAMFeat
Inference Speed	31FPS	6FPS	11FPS	24FPS	8FPS	21FPS

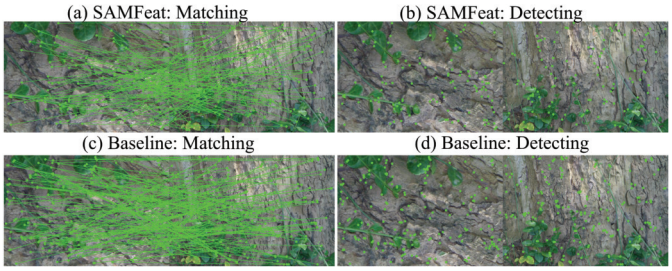


Fig. 11. Failure case of SAMFeat. When dealing with challenging cases such as huge view displacement or weak and repetitive texture, SAMFeat generates wrong matches (a) and keypoints (b). No improvements are provided comparing the baseline results (c and d).

at the pixel three threshold on the HPatches dataset. The results indicate that merely adjusting the loss weights of EAG does not achieve the same effectiveness as incorporating WSC. Specifically, with a fixed ASRD weight of 1.0, varying the EAG weight from 0.5 to 1.5 yields incremental improvements in MMA@3, peaking at 81.0. However, this peak still falls short of the performance enhancements observed when WSC is included, highlighting the unique contribution of WSC to the overall accuracy.

G. Matching Visualization

We provide qualitative visualizations in this subsection. A visualization of the feature matching (a) and keypoint detection (b) results, along with a comparison to the baseline approach (c and d) is shown in Figure 10. Leveraging SAM's generalized feature and semantic grouping, our method effectively learns and pairs more valid matches. Additionally, with the integration of SAM's edge knowledge, our approach detects more accurate keypoints along object boundaries, which further improves feature matching. All methods are evaluated under the same settings and hyperparameters for a fair comparison.

As mentioned in Section III-C in the full paper, SAMFeat learns edge maps from SAM and utilizes Edge Attention Guidance (EAG) to further enhance the precision of local feature detection and description by encouraging the network to prioritize attention to the edge region. Figure 9 demonstrates the learning outcome of SAMFeat. With the fine-grained object boundaries from SAM, SAMFeat is able to learn clear object edges. This illustrates two things: first, the encoded feature that is used to generate the edge map contains rich edge information, and second, with a clear and accurate generated edge map and EAGM, SAMFeat can better capture the details of edge areas and improve the robustness of local descriptors.

H. Inference Speed

To further demonstrate SAMFeat's high efficiency and fast inference speed, we conduct a comparison of inference time between other state-of-the-art methods in table X. We assessed the running speed of various methods using open-source code. In Table X, our approach demonstrated exceptionally competitive performance while maintaining a fast inference speed among many lightweight methods.

V. CONCLUSION AND FUTURE WORK

In this study, We introduce SAMFeat, a local feature learning method that harnesses the power of the Segment Anything Model (SAM). SAMFeat encompasses three innovations. Firstly, we introduce Attention-weighted Semantic Relation Distillation (ASRD), an auxiliary task aimed at distilling the category-agnostic semantic information acquired by the SAM encoder into the local feature learning network. Secondly, we present Weakly Supervised Contrastive Learning Based on Semantic Grouping (WSC), a technique that leverages the semantic groupings derived from SAM as weakly supervised signals to optimize the metric space of local descriptors. Furthermore, we engineer the Edge Attention Guidance (EAG) mechanism to elevate the accuracy of local feature detection and description. Our comprehensive evaluation of tasks such as image matching on HPatches and long-term visual localization on Aachen Day-Night consistently underscores the remarkable performance of SAMFeat, surpassing previous methods.

Although other visual foundation models like DINO-V2 [48] or SEEM [73] could potentially serve as alternative teachers, the focus of our study was specifically on SAM. Our methodology was designed around SAM's unique capabilities. Therefore, further investigation into alternative teachers was not pursued in this study. Future research could explore the applicability and potential advantages of employing other visual foundation models as teachers for local feature learning tasks. In addition, SAMFeat, like all other methods, generates wrong matching when facing challenging cases. For example, in Figure 11, when dealing with larger view displacement and repetitive texture. Solving these problems could also be a future direction. Potential solutions worth exploring include introducing spatial geometry information or spatial encoding techniques.

REFERENCES

[1] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, "HPatches: A benchmark and evaluation of handcrafted and learned local descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5173–5182.

[2] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2016, pp. 4104–4113.

[3] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[4] T. Sattler et al., "Benchmarking 6DOF outdoor visual localization in changing conditions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 8601–8610.

[5] X. Wang, Y. Hua, E. Kodirov, G. Hu, R. Garnier, and N. M. Robertson, "Ranked list loss for deep metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5202–5211.

[6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[7] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2564–2571.

[8] S. Xu, S. Chen, R. Xu, C. Wang, P. Lu, and L. Guo, "Local feature matching using deep learning: A survey," *Inf. Fusion*, vol. 107, Jul. 2024, Art. no. 102344.

[9] R. Xu et al., "MRFTrans: Multimodal representation fusion transformer for monocular 3D semantic scene completion," *Inf. Fusion*, vol. 111, Nov. 2024, Art. no. 102493.

[10] R. Xu et al., "DefFusion: Deformable multimodal representation fusion for 3D semantic segmentation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2024, pp. 7732–7739.

[11] Y. Yan, R. Xu, J. Zhang, P. Li, X. Liang, and J. Yin, "InstruGen: Automatic instruction generation for vision-and-language navigation via large multimodal models," 2024, *arXiv:2411.11394*.

[12] P. Ren et al., "InfiniteWorld: A unified scalable simulation framework for general visual-language robot interaction," 2024, *arXiv:2412.05789*.

[13] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-supervised interest point detection and description," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 224–236.

[14] M. J. Tyszkiewicz, P. Fua, and E. Trulls, "DISK: Learning local features with policy gradient," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2020, pp. 14254–14265.

[15] Z. Li and N. Snavely, "MegaDepth: Learning single-view depth prediction from internet photos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2041–2050.

[16] A. Kirillov et al., "Segment anything," 2023, *arXiv:2304.02643*.

[17] R. Bommasani et al., "On the opportunities and risks of foundation models," 2021, *arXiv:2108.07258*.

[18] J. Zhang et al., "NaVid: Video-based VLM plans the next step for vision-and-language navigation," 2024, *arXiv:2402.15852*.

[19] F. Xue, I. Budvytis, D. O. Reino, and R. Cipolla, "Efficient large-scale localization by global instance recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 17327–17336.

[20] J. L. Schönberger, M. Pollefeys, A. Geiger, and T. Sattler, "Semantic visual localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6896–6906.

[21] F. Xue, I. Budvytis, and R. Cipolla, "SFD2: Semantic-guided feature detection and description," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 5206–5216.

[22] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.

[23] D. Mishkin, F. Radenovic, and J. Matas, "Repeatability is not enough: Learning affine regions via discriminability," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 284–300.

[24] A. B. Laguna, E. Riba, D. Ponsa, and K. Mikolajczyk, "Key.Net: Keypoint detection by handcrafted and learned CNN filters," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5835–5843.

[25] Y. Tian, B. Fan, and F. Wu, "L2-Net: Deep learning of discriminative patch descriptor in Euclidean space," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 661–669.

[26] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas, "Working hard to know your neighbor's margins: Local descriptor learning loss," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4826–4837.

[27] Y. Tian, X. Yu, B. Fan, F. Wu, H. Heijnen, and V. Balntas, "SOSNet: Second order similarity regularization for local descriptor learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11016–11025.

[28] Z. Luo et al., "ContextDesc: Local descriptor augmentation with cross-modality context," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2527–2536.

[29] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "LIFT: Learned invariant feature transform," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 467–483.

[30] Y. Ono, E. Trulls, P. Fua, and K. M. Yi, "LF-net: Learning local features from images," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2018, pp. 6234–6244.

[31] J. Revaud, C. R. d. Souza, M. Humenberger, and P. Weinzaepfel, "R2D2: Reliable and repeatable detector and descriptor," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, Sep. 2019, pp. 12405–12415. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/3198dfd0aef271d22f7bcddd6f12f5cb-Paper.pdf

[32] M. Dusmanu et al., "D2-Net: A trainable CNN for joint description and detection of local features," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8084–8093.

[33] Z. Luo et al., "ASLFeat: Learning local features of accurate shape and localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6589–6598.

[34] C. Wang et al., "MTLDesc: Looking wider to describe better," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 2, 2022, pp. 2388–2396.

[35] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning feature matching with graph neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4938–4947.

[36] P. Lindenberger, P.-E. Sarlin, and M. Pollefeys, "LightGlue: Local feature matching at light speed," 2023, *arXiv:2306.13643*.

[37] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, "LoFTR: Detector-free local feature matching with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8922–8931.

[38] J. Yu, J. Chang, J. He, T. Zhang, J. Yu, and F. Wu, "Adaptive spot-guided transformer for consistent local feature matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 21898–21908.

[39] Q. Wang, J. Zhang, K. Yang, K. Peng, and R. Stiefelhagen, "MatchFormer: Interleaving attention in transformers for feature matching," in *Proc. Asi. Conf. Comput. Vis. (ACCV)*, 2022, pp. 2746–2762.

[40] J. Edstedt, Q. Sun, G. Bökman, M. Wadenbäck, and M. Felsberg, "RoMa: Robust dense feature matching," 2023, *arXiv:2305.15404*.

[41] X. He et al., "MatchAnything: Universal cross-modality image matching with large-scale pre-training," 2025, *arXiv:2501.07556*.

[42] H. He, J. Zhang, M. Xu, J. Liu, B. Du, and D. Tao, "Scalable mask annotation for video text spotting," 2023, *arXiv:2305.01443*.

[43] M. Kristan et al., "The ninth visual object tracking VOT2021 challenge results," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 2711–2738.

[44] L. Tang, P.-T. Jiang, Z.-H. Shen, H. Zhang, J.-W. Chen, and B. Li, "Chain of visual perception: Harnessing multimodal large language models for zero-shot camouflaged object detection," in *Proc. 32nd ACM Int. Conf. Multimedia*, Oct. 2024, pp. 8805–8814.

[45] L. Tang, P.-T. Jiang, H. Xiao, and B. Li, "Towards training-free open-world segmentation via image prompt foundation models," *Int. J. Comput. Vis.*, vol. 133, no. 1, pp. 1–15, Jan. 2025.

[46] B. Li, H. Xiao, and L. Tang, "ASAM: Boosting segment anything model with adversarial tuning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 3699–3710.

[47] C. Zhang et al., "A comprehensive survey on segment anything model for vision and beyond," 2023, *arXiv:2305.08196*.

[48] M. Oquab et al., "DINOv2: Learning robust visual features without supervision," 2023, *arXiv:2304.07193*.

[49] Z. Huang et al., "VS-Net: Voting with segmentation for visual localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6097–6107.

[50] H. Hu, Z. Qiao, M. Cheng, Z. Liu, and H. Wang, "DASGIL: Domain adaptation for semantic and geometric-aware image-based localization," *IEEE Trans. Image Process.*, vol. 30, pp. 1342–1353, 2021.

[51] C. Toft et al., "Semantic match consistency for long-term visual localization," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Jan. 2018, pp. 391–408.

[52] T. Shi, H. Cui, Z. Song, and S. Shen, "Dense semantic 3D map based long-term visual localization with hybrid features," 2020, *arXiv:2005.10766*.

[53] C. Toft, C. Olsson, and F. Kahl, "Long-term 3D localization and pose from semantic labellings," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 650–659.

[54] B. Fan et al., "Learning semantic-aware local features for long term visual localization," *IEEE Trans. Image Process.*, vol. 31, pp. 4842–4855, 2022.

[55] C. Wang et al., "Attention weighted local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 9, pp. 10632–10649, Sep. 2023.

[56] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[57] W. Park, D. Kim, Y. Lu, and M. Cho, "Relational knowledge distillation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3967–3976.

[58] T. Lindeberg, "Scale invariant feature transform," *Scholarpedia*, vol. 7, no. 5, p. 10491, 2012.
[59] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, "Large-scale image retrieval with attentive deep local features," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3456–3465.
[60] Y. Ono, E. Trulls, P. Fua, and K. M. Yi, "LF-Net: Learning local features from images," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2018, pp. 1–13.
[61] J. Revaud et al., "R2D2: Repeatable and reliable detector and descriptor," 2019, *arXiv:1906.06195*.
[62] S. Suwanwimolkul, S. Komorita, and K. Tasaka, "Learning of low-level feature keypoints for accurate and robust detection," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 2261–2270.
[63] A. Barroso-Laguna and K. Mikolajczyk, "Key.Net: Keypoint detection by handcrafted and learned CNN filters revisited," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 698–711, Jan. 2023.
[64] X. Zhao, X. Wu, J. Miao, W. Chen, P. C. Y. Chen, and Z. Li, "ALIKE: Accurate and lightweight keypoint detection and descriptor extraction," *IEEE Trans. Multimedia*, vol. 25, pp. 3101–3112, 2022.
[65] K. Li, L. Wang, L. Liu, Q. Ran, K. Xu, and Y. Guo, "Decoupling makes weakly supervised local feature better," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 15817–15827.
[66] Z. Wang, C. Wu, Y. Yang, and Z. Li, "Learning transformation-predictive representations for detection and description of local features," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 11464–11473.
[67] P. Gleize, W. Wang, and M. Feiszli, "SiLK-simple learned keypoints," 2023, *arXiv:2304.06194*.
[68] G. Potje, F. Cadar, A. Araujo, R. Martins, and E. R. Nascimento, "XFeat: Accelerated features for lightweight image matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 2682–2691.
[69] Q. Wang, X. Zhou, B. Hariharan, and N. Snavely, "Learning feature descriptors using camera pose supervision," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Jan. 2020, pp. 1–12.
[70] R. Pautrat, V. Larsson, M. R. Oswald, and M. Pollefeys, "Online invariance selection for local feature descriptors," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 707–724.
[71] J. L. Schonberger, H. Hardmeier, T. Sattler, and M. Pollefeys, "Comparative evaluation of hand-crafted and learned local features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1482–1491.
[72] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5828–5839.
[73] X. Zou et al., "Segment everything everywhere all at once," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2023, pp. 1–43.

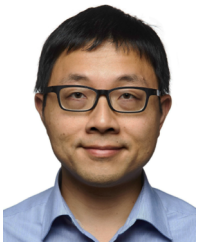
Jingqian Wu received the B.S. degree from Wake Forest University, USA, in 2022, and the M.S. degree from Northwestern University, USA, in 2024. He is currently pursuing the Ph.D. degree with the Department of Electrical and Electronic Engineering, The University of Hong Kong. His research interests include computational imaging, computer vision, and machine learning.

Rongtao Xu (Member, IEEE) received the B.S. degree in information and computing science from the Huazhong University of Science and Technology, China, in July 2019, and the joint Ph.D. degree from the State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, and the School of Artificial Intelligence, University of Chinese Academy of Sciences. Since 2024, he has been with the Institute of Automation, Chinese Academy of Sciences. His research interests include embodied AI, multimodal learning, and robotic vision.

Zach Wood-Doughty received the B.A. degree from the Carleton College in 2014 and the Ph.D. degree from Johns Hopkins University in 2021. He is currently an Assistant Professor of instruction with the Department of Computer Science, Northwestern University. His teaching and research focus on machine learning and causal inference.

Changwei Wang received the B.S. degree in software engineering from Tiangong University, China, in July 2019, and the joint Ph.D. degree from the State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, and the School of Artificial Intelligence, University of Chinese Academy of Sciences. Since 2024, he has been with Shandong Computer Science Center (National Supercomputer Center in Jinan). His research interests include multimodal learning, robotic vision, embodied intelligence, and edge intelligence.

Shibiao Xu (Member, IEEE) received the B.S. degree in information engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 2009, and the Ph.D. degree in computer science from the Institute of Automation, Chinese Academy of Sciences, Beijing, in 2014. He is currently a Professor at the School of Artificial Intelligence, Beijing University of Posts and Telecommunications. His current research interests include computer vision and image-based three-dimensional scene reconstruction and understanding.



Edmund Y. Lam (Fellow, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Stanford University. He was a Visiting Associate Professor with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology. He is currently a Professor of electrical and electronic engineering at The University of Hong Kong. He is the Computer Engineering Program Director and a Research Program Coordinator with the AI Chip Center for Emerging Smart Systems. He is a Founding Member of The Hong Kong Young Academy of Sciences. His research interests include computational imaging algorithms, systems, and applications. He is a fellow of *Optica*, SPIE, IS&T, and HKIE.