Query-Driven Feature Learning for Cross-View Geo-Localization

Shuyu Hu, Zelin Shi, Tong Jin, and Yunpeng Liu, *Member, IEEE*

***Abstract*— The cross-view geo-localization (CVGL) task aims to accurately retrieve a location using images captured from different platforms, such as satellites and drones, which is particularly challenging due to a large variation in viewpoint. Current methods mainly focus on rigid strategies such as partitioning or sorting local features, which may be ill-suited to accommodate the variance of viewpoint and distance scale in different camera perspectives. To address these issues, we propose a novel method called query-driven feature learning (QDFL) to query viewpoint-invariant feature vectors autonomously. Our method incorporates an adaptive query embedding unit (AQEU) and a feature fusion unit (FFU). AQEU adjusts feature map and implements a coarse query process to extract the contextual clues. FFU further refines feature map, fusing it at spatial and channel dimensions, tending to withdraw more fine-grained features. Subsequently, AQEU executes a fine query on salient landmarks in the fused feature map, enhancing the minutia descriptive power of query vectors. In addition, we use parameter-efficient transfer learning (PETL) manner by integrating tunable adapters into the frozen pretrained backbone, maintaining feature representation capabilities of foundation models while enabling seamless adaptation to CVGL task. Extensive experiments show that our method achieves state-of-the-art (SOTA) performances on two well-known datasets, University-1652 and SUES-200. Moreover, our method exhibits an excellent generalizability compared with current SOTA methods in cross-dataset experiments. The code is available at https://github.com/Shuyu-Hu/QDFL**

***Index Terms*— Contrastive representation learning, cross-view, geo-localization, image retrieval, unmanned aerial vehicle (UAV) navigation.**

I. Introduction

CROSS-VIEW geo-localization (CVGL) task aims to address the challenge of relocating an unmanned aerial vehicle (UAV) using airborne cameras and databases, in circumstances where global navigation satellite system (GNSS) signals are weak or unavailable. It has numerous applications, including autonomous delivery [1], search and rescue missions [2], environmental monitoring [3], precision agriculture [4], and autonomous 3-D scene reconstruction [5].

The application of UAV-based CVGL is classified into two branches, drone-view localization (drone–satellite) and drone navigation (satellite–drone). In the drone-view localization task, drone-view images are used as query inputs to search for the most closely matched satellite images in a database (referred to as the gallery). Conversely, drone navigation focuses on using satellite-view images as queries to find the most closely matched drone-view images in the gallery, to guide the drone to a specific location. By enabling accurate localization without relying on GNSS, CVGL task is crucial for ensuring the reliability and efficiency of UAV operations in complex ambiance. Unlike ground-based cameras, UAV-mounted cameras capture more contextual information and geometric features shared with satellite images due to their downward angle [6], [7]. However, CVGL task remains arduous due to technical challenges such as variations in viewpoint, illumination, visibility, and rotation of photographic apparatus.

While researchers have explored various methods to enhance the performance of CVGL [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21] task, there is still a gap in practice. LPN [13] mainly focuses on parting feature map using a manual partitioning strategy, which is typically based on a strong assumption that the objective is in the center of the image and the contextual information is at the edges. Usually this assumption is seldom guaranteed. Although shifting-dense partition learning (SDPL) [14] relaxes this assumption by improving manual partitioning strategy, there still exist disruption issues. It significantly undermines the continuity of the semantic structure within feature map. Meanwhile, CCR [15] uses counterfactual causal reasoning to enhance performance by emphasizing the features of target buildings. CCR overlooks valid background elements such as vegetation or roads. As a result, its performance may deteriorate when the target buildings occupy a lesser portion of the image. In addition, foundation models [22], [23], [24] trained on large-scale datasets with supervised or self-supervised manner have emerged recently and demonstrated exceptional representational capabilities. However, directly applying these models to CVGL task and fully fine-tuning them can overwrite prior knowledge, potentially leading to catastrophic forgetting problem [25]. Furthermore, foundation models typically have a large number of parameters, making full fine-tuning computationally expensive, memory-intensive, and laborious.

Overall, there are still two unresolved issues.

1) Nearly all the existing methods predominantly focus on strategies that partition, segment, or constrain feature maps, overlooking the variance of perspective, illumination, and contextual changes brought by different camera viewpoints and distance scales.

Received 13 December 2024; revised 25 February 2025 and 26 March 2025; accepted 2 April 2025. Date of publication 8 April 2025; date of current version 30 April 2025. *(Corresponding authors: Zelin Shi; Yunpeng Liu.)*

Shuyu Hu and Tong Jin are with the Key Laboratory of Opto-Electronic Information Processing and Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China, and also with the School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: hushuyu@sia.cn; jintong@sia.cn).

Zelin Shi and Yunpeng Liu are with Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China (e-mail: zlshi@sia.cn; ypliu@sia.cn).

Digital Object Identifier 10.1109/TGRS.2025.3558924

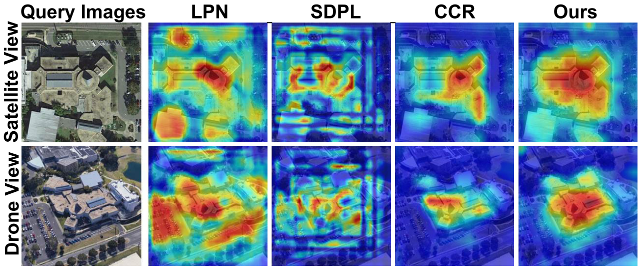


Fig. 1. Feature map comparison of LPN [13], SDPL [14], CCR [15], and our proposed method. The first column shows the query image of the same location but from different perspectives. The second, third, and fourth columns display the backbone feature maps of LPN, SDPL, and CCR, respectively. The fifth column presents feature map generated by our method (QDFL). It is evident that QDFL focuses more accurately on the target building compared with the other approaches.

2) Full fine-tuning foundation models for CVGL may risk catastrophic forgetting [25] and require high computational costs that hinders training on resource-limited devices.

To address 1), we propose a simple yet effective method called query-driven feature learning (QDFL). QDFL autonomously learns a set of trainable queries, establishing critical mutual priors in between perspectives and applying attention mechanisms to query robust representations within feature maps. The QDFL comprises two core units: an adaptive query embedding unit (AQEU) and a feature fusion unit (FFU). AQEU initializes learnable queries and implements queries on feature map. It extracts viewpoint-invariant feature vectors from coarse to fine. These learned query vectors seek valid information consistently across input images, effectively bridging the differences caused by viewpoint and scale variations. Meanwhile, FFU enables the model to focus on salient landmarks within feature map, gaining spatial and channel enhancement. QDFL actively queries for both global and fine-grained features from feature maps, effectively aligning descriptors extracted from drone and satellite views. Furthermore, we leverage the multi-similarity loss (MS-Loss) [26] into CVGL task and adapt it specifically for it, thus further enhancing the model's performance.

To address 2), we implement parameter-efficient transfer learning (PETL) using bottleneck adapters [27], [28]. By integrating tunable adapters into the frozen pretrained backbone, it preserves the original parameters of foundation model to maintain the model's feature extracting capabilities. Adapters empower the foundation model to seamlessly adapt to downstream tasks, bridging the gap between pretraining and the CVGL task.

Feature maps extracted from different methods using the same pair of drone–satellite images are illustrated in Fig. 1. The figure shows that both LPN and SDPL, which rely on manual partitioning strategies, lead to weight discontinuities at the squared segment boundaries. CCR focuses on target building excessively, overlooking valid background information. In comparison, QDFL concentrates on the primary building while concerning relevant background details, further enriching the overall feature representation.

In summary, our contributions are highlighted as follows.

1) A novel QDFL framework is proposed for CVGL task, which incorporates two innovative units: AQEU and FFU. With the collaborative interaction of two units, QDFL consistently queries feature vectors across different viewpoints, generating robust viewpoint-invariant representations.
2) The proposed AQEU is dedicated to learning a set of query vectors to capture mutual feature patterns from cross-view images and extracting robust viewpoint-invariant features in a coarse-to-fine manner. The FFU is designed to enhance fine-grained features by fusing feature maps across spatial and channel dimensions, using lightweight attention weights along dimensions to emphasize salient landmarks.
3) To maintain the representation power of foundation models in CVGL task, we introduce an adapter-based PETL paradigm by integrating tunable adapters into the frozen DINOv2-B backbone, which alleviates the training burden and avoids catastrophic forgetting.
4) For effective feature optimization, we pioneer the application of MS-Loss in CVGL task, exploiting similarity relations within mini-batches. Furthermore, we complement JS-Loss to align feature distributions across different perspectives. Extensive experiments on the University-1652 and SUES-200 datasets demonstrate that our method achieves state-of-the-art (SOTA) performance with a superior generalization capability across different datasets.

II. Related Work

This section briefly reviews related works on CVGL, PETL, and deep metric learning loss.

A. Cross-View Geo-Localization

Early research in CVGL mainly focuses on matching panorama ground and satellite image pairs [29], [30], [31], [32], [33], [34], [35], with several well-known datasets [36], [37], [38], [39] being introduced. With the introduction of the drone–satellite dataset [8] and the development of drone technology, UAV-based CVGL gradually gained attention. Zhu et al. [40] addressed differences in aerial photography captured by drones flying at different altitudes, leading to the construction of a multialtitude dataset known as SUES-200. With the proposal of large-scale UAV-based CVGL datasets, many UAV-based CVGL tasks have been conducted. These methods can be categorized into two main branches: CNN-based methods and ViT-based methods.

1) CNN-Based Methods: Zheng et al. [8] used instance loss to train a multibranch CNN model on University-1652, achieving considerable improvement over other baseline models. Wang et al. [13] proposed LPN, dividing the ResNet-50 feature map into several square rings to generate feature descriptors for geographic targets and background information. Lin et al. [9] gave a solution that considers both discriminative representation and keypoints detection in an end-to-end manner, and the proposed RK-Net achieves higher performance than the vanilla LPN by adding a modicum of parameters. Chen et al. [14] observed that simply parting feature maps like LPN inevitably damages the effective information in feature map and proposed an SDPL strategy, enhancing its anti-offset ability and maintaining the local representation ability of block partitioning. Shen et al. [41] proposed MCCG, which used an improved triplet attention module to balance spatial and channel attention within the ConvNeXt-Tiny feature map distribution and used multiple-classifier blocks, extract robust descriptors from feature map. Ge et al. [20] proposed a multibranch joint representation learning based on information fusion strategy (MJRLIFS), using information fusion strategies to extract global and local features while preserving local features to assist in learning features more robustly. Deuser et al. [42] proposed a simple cross-view matching training pipeline called Sample4Geo, which uses ConvNeXt-Base [43] model as the backbone and a custom sampling strategy to avoid multiple drone images of the same category in a batch, and using symmetric InfoNCE loss for model training. A counterfactual causal reasoning-based method has been proposed by Du et al. [15], which ensures the emphasis on the main details of the target structure and the robustness of the extracted semantic information. Xia et al. [44] proposed domain alignment and scene consistency (DAC), using contrastive learning loss, and adopted the DAC module to perform granular alignment on patch tokens. Wu et al. [21] proposed contrastive attributes mining and position-aware partitioning (CAMP), establishing negative sample supervision using the CAM strategy.

2) ViT-Based Methods: Apart from CNN-based methods that have been mentioned above, there are also various methods to use ViTs as the backbone. FSRA [16] performs heatmap segmentation and alignment on feature maps output by the ViT-S model, enabling the network to focus more on identifying effective semantic features and achieving better performance. Zhao et al. [45] proposed TransFG, using the feature aggregation module and gradient guidance module to extract effective cross-view feature descriptions from feature maps output by the ViT backbone. Li et al. [19] introduced a two-stage method, GeoFormer, incorporating linear attention and multiscale feature aggregation at the first stage and using semantic-guided region segmentation and hierarchical rotation matching with SuperPoint and LightGlue at the second stage. Liu et al. [18] adopted Swinv2-T as the backbone and introduced a semantic-aware graph convolutional network (SeGCN) to perform operations on graphs constructed with the same semantic features. Lv et al. [46] proposed SRLN, a pruned Swin-transformer-based framework that integrates multiscale feature aggregation to harmonize drone directional information and satellite environmental features for enhanced performance in CVGL task. Compared with CNN-based methods, ViT-based methods tend to have better contextual consistency and can achieve performances close to or even better than CNN-based methods at lower input resolutions. Based on the above reasons, we choose ViT as the backbone to use the powerful context extraction capability provided by the attention mechanism.

B. Parameter-Efficient Transfer Learning

ViT and its variants [22], [24], [47] pretrained from large-scale datasets, such as the ImageNet [48], have been proven to be robust foundation models for many computer vision tasks. Fully leveraging the power of these backbones and adapting to the task-specific distribution is instrumental; however, simply fully fine-tuning the backbone may disrupt the prior knowledge, leading to catastrophic forgetting problems [25]. Fine-tuning only the last few layers of ViT could alleviate the forgetting problem, but this makes fine-tuning parameters inefficient and only addresses the symptoms rather than the root cause. Researchers have explored numerous PETL methods in the natural language processing field [27], [49], [50]. Houlsby et al. [27] introduce the "adapter" approach, a lightweight and modular method for fine-tuning pretrained language models by inserting small trainable layers into the transformer. Lester et al. [50] show that prompt-tuning can achieve competitive performance by optimizing soft prompts, while keeping the pretrained model frozen. This approach outperforms fully fine-tuning the entire model, especially as the model size increases. Informed by the concepts and methods presented above, works have extended these approaches to ViT-based computer vision tasks [28], [51], [52], [53], [54], [55]. AdaptFormer [51] is a PETL approach for ViTs that inserts lightweight adaptation modules into feedforward layers, achieving competitive performance with minimal computational overhead. Lu et al. [28] proposed a dual-stage visual place recognition method that uses a hybrid adaptation approach using lightweight adapters and a mutual nearest neighbor local feature loss to achieve SOTA performance. Leveraging rich prior knowledge embedded in ViT and fine-tuning it in a parameter-efficient manner is essential for enhancing performance in the CVGL task. Inspired by [27] and [28], we add learnable adapters into the ViT-based foundation backbone to implement a seamless PETL process to the CVGL task.

C. Deep Metric Learning Loss

Since the construction of CVGL data is based on pairs of samples, and the objective is to learn viewpoint-invariant representations, this concept is highly aligned with the goals of deep metric learning. Specifically, deep metric learning aims to learn representations that preserve the similarity between samples of the same class while distinguishing between samples from different classes. Hadsell et al. [56] proposed the contrastive loss, maximizing the similarity between positive pairs while minimizing the similarity of negative pairs. Similarly, Hoffer and Ailon [57] introduced triplet loss, which uses triplets as training samples, learning embeddings by minimizing the distance between an anchor, a positive sample, and a negative sample. Moreover, Wang et al. [26] introduced the MS-Loss, which integrates pair sampling and weighing schemes, using a weighted combination of multiple similarity measures between pairs of samples. In the context of CVGL

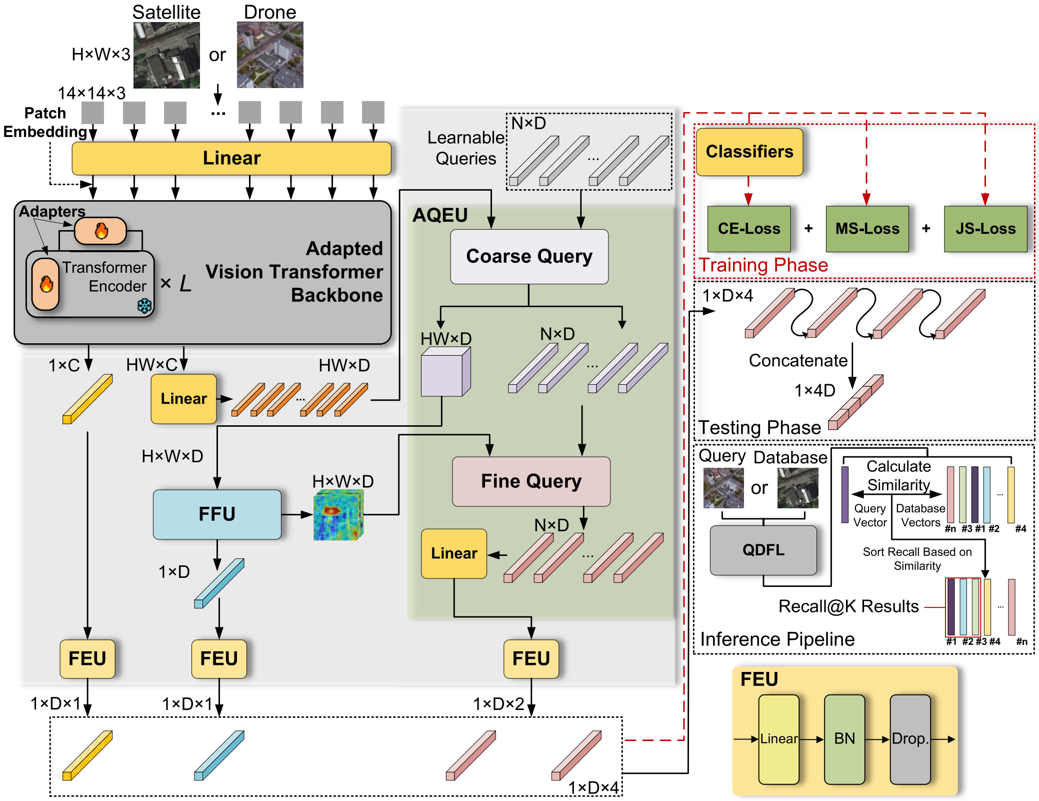


Fig. 2. Overall view of the QDFL. The QDFL uses shared weights in processing with both satellite- and drone-view images. Input images are processed by the adapted vision-transformer (ViT) backbone (e.g. DINOv2), with QDFL separating patch and class tokens. AQEU (Fig. 4) initializes learnable queries, performing a coarse query on the processed patch tokens, while FFU (Fig. 5) refines feature map output by AQEU for fine querying. Vectors are then processed by the FEU (bottom right) to produce 4 × D feature vectors. During inference, these are concatenated into a 1 × 4D descriptor.

task, various metric learning losses have been used to improve performance. Liu et al. [18] considered images from different viewpoints and applied triplet loss to constrain the extracted features. Similarly, Zhao et al. [45] used triplet-center loss to align the data distribution of two perspectives, further demonstrating the effectiveness of metric learning in CVGL task. In this work, we adopt the MS-Loss into CVGL task, effectively balancing the tradeoff between separating dissimilar samples and pulling together similar ones. This ensures that the learned features exhibit both viewpoint invariance for the same location and high separability for different locations.

III. Proposed Method

This section comprises feature extraction and adaptation process (Section III-A), QDFL framework (Section III-B), and loss function (Section III-C). The overall architecture of QDFL is illustrated in Fig. 2.

Problem Formulation: The essence of CVGL task lies in solving an image retrieval problem. Given a set of cross-view image pairs $\{x_i^q, x_i^g\}$, where q denotes query image, and g denotes gallery images. Each pair is associated with a unique location label i, where $i \in \mathcal{L}$ and $\mathcal{L}$ denotes the set of all the locations. The objective is to construct a robust feature extractor F, which transforms an input image $x_i^\cdot$ into a descriptor $d_{x_i^\cdot}$. Here, $x_i^\cdot$ represents the query or gallery input from location i

$$d_{x_i^\cdot} = F\left(x_i^\cdot\right), \quad i \in \mathcal{L}. \tag{1}$$

The task aims to extract $d_{x_i^\cdot}$ from images captured on different platforms (e.g., drone/satellite) and map them into a shared latent space, where cross-view images from the same location are projected closely, while those from different locations are projected further apart.

A. Feature Extraction and Adaptation Process

Extracting contextual information that remains highly invariant to viewpoint changes and occlusions is crucial. CNNs enlarge the receptive field with stacked convolution and pooling processes layer by layer, while ViT benefits from its self-attention (SA) mechanism, allowing it to obtain large receptive fields even in the shallowest layers. This makes ViT excel over CNN in extracting and preserving contextual-sensitive features. Thus, we use the ViT-based DINOv2 model [24] as the backbone.

The DINOv2-B model is built on the ViT-B/14 architecture. Similar to the standard ViT pipeline processing an input image by first dividing it into 16 × 16 nonoverlapping patches, DINOv2 processes it into 14 × 14 patches. Each of these patches is then flattened into a 1-D vector, which is subsequently linearly projected into a C-dimensional embedding space $x_p \in \mathbb{R}^{N \times C}$. ViT introduces a learnable [CLS] token, which is concatenated at the beginning of the patch embedding sequence $x_0 = [\text{CLS}], x = [x_0, x_p^1, x_p^2, \ldots, x_p^N]$, serving as a global representation of the image. Moreover, like positional embeddings commonly used in the natural language

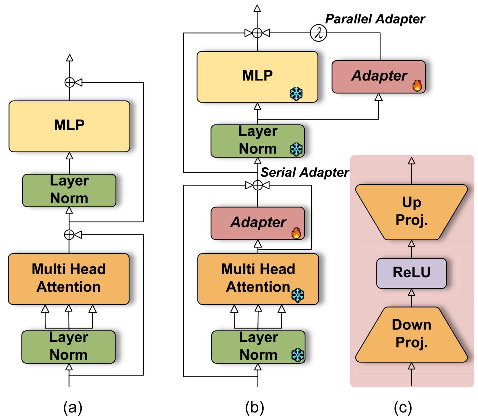


Fig. 3. Overall view of adaptation. (a) Vanilla transformer block. (b) Adapted transformer block. (c) Architecture of the adapter. As shown in (b), adapters are added after the MHA and in parallel with the feedforward network within the encoder block.

processing field, the ViT uses patch embedding to introduce positional information to these patches to maintain spatial context. Then, processed patches are fed into transformer encoder layers to extract the effective information of the image. As shown in Fig. 3(a), the vanilla transformer encoder block can be expressed as

$$\begin{aligned} x_l^0 &= \mathrm{MHA}(\mathrm{LN}(x_{l-1})) + x_{l-1} \\ x_l &= \mathrm{MLP}\left(\mathrm{LN}\left(x_l^0\right)\right) + x_l^0 \end{aligned} \tag{2}$$

where x_l denotes the output of the lth transformer block, MHA denotes the multihead attention operation, $\mathrm{LN}(\cdot)$ represents the layer normalization operation, and MLP represents the multilayer perceptron.

Inspired by previous works [27], [28], [54], [55], we insert bottleneck-style adapters [28], [51] into ViT blocks and freeze all the vanilla ViT parameters during training except for adapters, enabling PETL by fine-tuning only a few number of parameters. The adapted transformer encoder block, illustrated in Fig. 3(b), incorporates adapters with the bottleneck MLP structure [Fig. 3(c)]. The serial adapter placed after the MHA operation uses a residual connection within the adapter. This ensures that adapters do not dominate the forward or backward propagation until they have converged. The parallel adapter acts as an auxiliary bypass for the MLP. It can be expressed as the following:

$$\begin{aligned} x_l^{0'} &= \mathrm{Ada}(\mathrm{MHA}(\mathrm{LN}(x_{l-1}))) + x_{l-1} \\ x_l &= \lambda \cdot \mathrm{Ada}\left(\mathrm{MLP}\left(\mathrm{LN}\left(x_l^0\right)\right)\right) \\ &\quad + \mathrm{MLP}\left(\mathrm{LN}\left(x_l^{0'}\right)\right) + x_l^{0'} \end{aligned} \tag{3}$$

where Ada represents the adapter block, and λ denotes the fixed threshold that controls the output of parallel adapter.

Same as the vanilla ViT, the adapted ViT outputs $x_{p_i} \in \mathbb{R}^{B\times(N+1)\times C}$, where B denotes the batch size, and $(N+1)$ denotes the length of [CLS] token $c_i \in \mathbb{R}^{B\times 1\times C}$ and patch tokens $x_i \in \mathbb{R}^{B\times N\times C}$. After extracting patch tokens from backbone, we adopt a linear layer to reduce their dimensionality from C to D $(D < C)$, obtaining compressed patch tokens x_i'. In addition, we also absorb feature extraction unit (FEU) from MCCG [41], as shown in the bottom right of Fig. 2. The [CLS] token c_i from the backbone along with two feature vectors from AQEU (Section III-B1) and one from FFU (Section III-B2) are processed to extract four refined feature vectors $V_n \in \mathbb{R}^{1\times D}$, where $n \in \{0, 1, 2, 3\}$. The c_i is processed by FEU to generate the first refined feature vector, V_0.

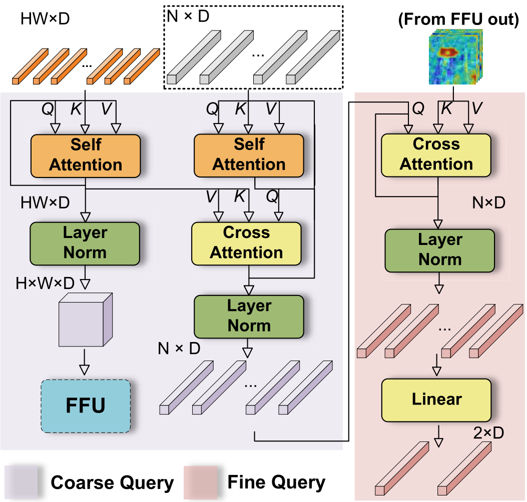


Fig. 4. Overall view of AQEU. AQEU first applies an SA mechanism to preprocess feature map. Simultaneously, it initializes query vectors, and then is activated by the SA mechanism. Coarse queries are performed on the preprocessed feature map using CA mechanism. A second CA mechanism is then used for fine queries on the refined feature maps from FFU. Finally, the query vectors are projected into a fixed number of refined vectors.

B. QDFL Framework

Features extracted from the backbone contain copious semantic information. However, relying solely on the backbone cannot fully interpret this information. To minimize the omission of semantic structures essential for CVGL, we propose a QDFL method, which is capable of autonomously extracting features with viewpoint invariance using a set of mutual prior pattern query vectors.

1) Adaptive Query Embedding Unit: Inspired by [58] and [59], we design a novel coarse-to-fine feature query unit, called AQEU, as illustrated in Fig. 4. In our proposed method, we introduce a set of learnable query vectors as mutual prior patterns between perspectives, enabling the model to focus on extracting robust, viewpoint-invariant features from feature maps. The core component of AQEU is the MHA mechanism [60]. MHA includes three inputs, query, key, and value. For computational efficiency, these inputs are projected into multiple parallel attention heads. The MHA mechanism enables two fundamental operations. Firstly, SA captures relationships within a single input [e.g., MHA(q,q,q)], and second, cross-attention (CA) allows interactions between two different sequences, where one attends to the features of the other [e.g., MHA(q,x,x)].

After introducing the MHA mechanism, we introduce the proposed AQEU. We first process x_i' by an SA mechanism with an output of $x_{\mathrm{sa}} \in \mathbb{R}^{B\times N\times D}$, which can be expressed as follows:

$$x_{\text{sa}} = \text{MHA}_x\left(x_i{}', x_i{}', x_i{}'\right) + x_i{}' \tag{4}$$

where, $x_i{}'$ denotes the processed patch tokens, x_{sa} denotes the output transformed by the SA mechanism, and MHA_x denotes the SA mechanism for feature map. Synchronized with the above process, a fixed set of learnable query vectors $Q_0 = [q_1^0, q_2^0, \ldots, q_n^0]$ is initialized. These query vectors are independent of the input feature map, serving as prior representations that the model can adaptively refine throughout training and extract viewpoint-invariant features from feature maps during the inference process. The SA mechanism is applied to Q_0 before it is used for querying

$$Q_1 = \text{MHA}(Q_0, Q_0, Q_0) + Q_0. \tag{5}$$

After which, we apply the first CA mechanism to perform feature queries on x_{sa} using learnable queries Q_1

$$Q_2 = \text{LN}(\text{MHA}(Q_1, x_{\text{sa}}, x_{\text{sa}}) + Q_1) \tag{6}$$

where Q_2 has the same size of Q_0, and $\text{LN}(\cdot)$ denotes the layer norm. By remaining decoupled from the input, these learnable queries serve as flexible filters and shared priors, allowing the model to focus on the most discriminative features.

The coarse query vectors that are queried by the x_{sa} are generated. However, using a standalone CA mechanism for generating attention-weighted feature maps tends to overlook fine-grained local information in feature map. To mitigate this issue, we propose routing the output x_{sa} via a FFU (in Section III-B2) to apply further spatial–channel attention weighing. Subsequently, we use the second CA mechanism to perform a fine query on FFU-weighted feature map. The learning query Q_{aqeu} can calculate and apply attention weights to the fused feature map x_{ffu}. This process can be expressed as the following equation:

$$Q_3 = \text{MHA}(Q_2, \text{LN}(x_{\text{ffu}}), \text{LN}(x_{\text{ffu}})) + Q_2 \tag{7}$$

where x_{ffu} denotes the FFU-weighted feature map, and $\text{LN}(\cdot)$ denotes the layer norm operation. Details of FFU will be elaborated in Section III-B2. After the fine query process, lossless encoding of these query features is crucial. Directly adding or concatenating the final query results does not guarantee the generation of a sufficient and compact hierarchical representation. A natural solution is to use linear layers for continuous scaling, which further refines the features. The query vectors Q_{aqeu} are remapped by a linear layer, generating two refined feature vectors V_1 and V_2.

2) FFU: Simply querying feature vectors from a preprocessed feature map can lead to overlooking salient landmarks that contain crucial viewpoint-invariant features. Thus, a more nuanced approach is essential to ensure that all the relevant spatial features are considered during the querying process. To address this, we propose FFU, which incorporates 2-D fusion mechanisms and spatial–channel attention arranged sequentially to compute attention weights along both the height and width dimensions on feature map.

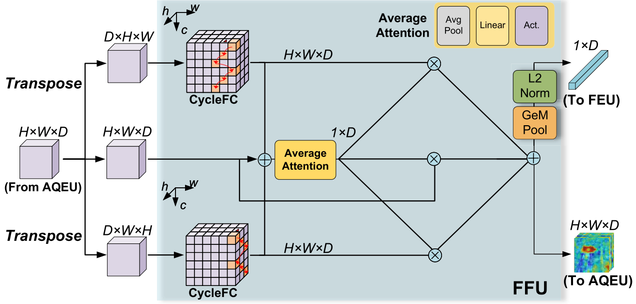


Fig. 5. Overall view of FFU. The preprocessed feature map is reshaped into two orientations, followed by width and height fusion using two CycleFC [61] layers. An attention matrix is then computed to weigh the tensors along the height, width, and channels. After summing the weighted feature map, it splits into two branches: one returns to AQEU for fine querying, while the other undergoes GeM pooling, and L2 normalization and is passed to the FEU.

As depicted in Fig. 5, given a feature map $x_{\text{sa}} \in \mathbb{R}^{N \times D}$ output by AQEU, we reshape feature map to two orientations, $x_{\text{sa}}^h \in \mathbb{R}^{D \times H \times W}$ and $x_{\text{sa}}^w \in \mathbb{R}^{D \times W \times H}$. Then, we use CycleFC [61] layer to fuse information across both the spatial and channel dimensions, performing effective context aggregation with linear complexity. Two CycleFC layers perform width and height fusion, respectively. The CycleFC introduces a receptive field with step sizes of Step_H and Step_W, corresponding to the height and width dimensions. The operation of two fusion mechanisms at a fixed position (i, j) is expressed by the following equation:

$$\text{CycleFC}(X)_{i,j,:} = \sum_{d=0}^{D_{\text{in}}} X_{i+\delta_i(d), j+\delta_j(d), d} \cdot W_{d,:}^{\text{mlp}} + b \tag{8}$$

where $(X)_{i,j,:}$ represents the values of all the channels at spatial location (i, j), and $W_{d,:}^{\text{mlp}} \in \mathbb{R}^{D_{\text{in}} \times D_{\text{out}}}$ and $b \in \mathbb{R}^{D_{\text{out}}}$ are the learnable parameters of the CycleFC layer. The spatial offsets $\delta_i(d)$ and $\delta_j(d)$ for the height and width dimensions, respectively, in the dth channel are calculated as follows:

$$\begin{aligned}\delta_i(d) &= \left(d \bmod \text{Step}_H\right) - 1\\ \delta_j(d) &= \left\lfloor \frac{d}{\text{Step}_H} \right\rfloor \bmod \text{Step}_W - 1.\end{aligned} \tag{9}$$

After the fusion operation, we apply an averaging-attention mechanism, which consists of average pooling, a linear layer, and a Softmax activation. By averaging the attention across the fused spatial–channel feature maps, viewpoint-invariant features from prominent landmarks are emphasized, while contextual regions are downweighted. We generate the attention matrix A from the output of the preceding operations. This attention matrix is then back to reweight feature maps along the height (h), width (w), and channel (c) dimensions. The entire fusion process can be formally represented by the following equations:

$$\begin{aligned}A &= \text{Softmax}\left(\text{MLP}\left(\text{Avg}\left(x_{\text{sa}}^h + x_{\text{sa}}^w + x_{\text{sa}}\right)\right)\right)\\ x_{\text{ffu}} &= A_h \cdot x_{\text{sa}}^h + A_w \cdot x_{\text{sa}}^w + A_c \cdot x_{\text{sa}}\end{aligned} \tag{10}$$

where $A = [A_h, A_w, A_c]$, and A_h, A_w, and A_c denote the attention matrices corresponding to the height, width, and channel dimensions, respectively. x_{sa}^h and x_{sa}^w are the feature maps along the height and width dimensions, respectively. x_{sa} represents the original input along the channel dimension.

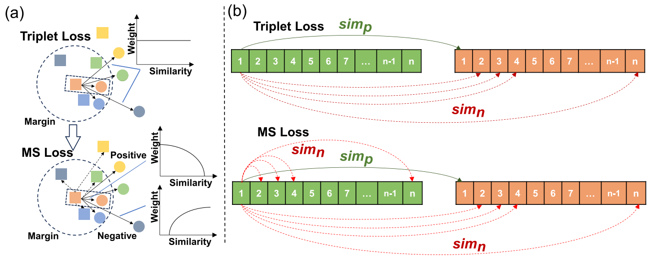


Fig. 6. (a) Rectangle and circle blocks represent samples from different views in a mini-batch. Colors represent the classes. Triplet loss considers all the negatives equally and focuses solely on distinguishing images from different views. In contrast, the MS-Loss considers all the samples in a mini-batch, weighing them considering their attributes (positive or negative) and their similarity to other samples. (b) Unlike previous works, when calculating the MS-Loss, we not only compute the similarity between samples from different perspectives relative to the anchor but also assess the similarity between different samples from the same perspective.

To fully leverage feature map processed by FFU, we designed two branches. One branch returns feature map to AQEU for the fine query process, while the other undergoes global average pooling (GeM) [62] followed by L2 normalization. Finally, the vectors are processed by FEU, obtaining feature vector V_3.

C. Loss Function

1) MS-Loss: The soft-margin triplet loss [57] has demonstrated its effectiveness in CVGL domain [15], [16], [18], [41], [45], but it mainly focuses relative similarity on cross-view pairs. As illustrated in Fig. 6(a), triplet loss considers all the negative samples with the same weight, constraining the model to learn in a more informative manner. Besides, as shown in Fig. 6(b), previous works [15], [16], [18] only trained models to distinguish images from different views may result in insufficient separability of samples from the same perspective in the latent space, affecting the performance. To address this limitation, we use and adapt MS-Loss [26] into CVGL task, which takes into account both relative and self-similarity in mini-batches, while weighing samples dynamically. The MS-Loss can be computed as

$$L = \frac{1}{N}\sum_{i=1}^{N}\left\{\frac{1}{\alpha}\log\left[1+\sum_{k\in P_i}e^{-\alpha(S_{ik}-\lambda)}\right]\right. \\ \left. + \frac{1}{\beta}\log\left[1+\sum_{k\in N_i}e^{-\beta(S_{ik}-\lambda)}\right]\right\}. \quad (11)$$

The above equation describes an anchor sample s_i within a batch, where $i \in \{1, \ldots, n\}$, with the corresponding positive and negative index sets, denoted as P_i and N_i, respectively. The terms α, β, and λ represent the loss hyperparameters. The first term in the N summation symbol represents the weighted component for positive samples, while the second term represents the weighted component for negative samples. The exponent in both the terms indicates the weight associated with each sample, which is determined by the sample's attributes (positive or negative) and its similarity to the anchor sample.

2) JS-Loss: The introduction of JS-Loss in TransFG [45] for CVGL task has proven to be effective. Unlike the KL-Loss, the JS-Loss offers calculation symmetry, which allows for better alignment of features between cross-view samples. The JS-Loss is expressed as follows:

$$\mathrm{JS}(Q\|G) = \frac{1}{2}\left(\mathrm{KL}\left(Q\middle\|\frac{Q+G}{2}\right)+\mathrm{KL}\left(G\middle\|\frac{Q+G}{2}\right)\right) \quad (12)$$

where Q represents the query output, while G represents the output of gallery.

3) Total Loss: The loss function is defined as follows:

$$L_{\text{total}} = L_{\text{CE}} + L_{\text{MS}} + L_{\text{JS}}. \quad (13)$$

Here, L_{CE} represents the cross-entropy loss, L_{MS} represents the MS-Loss, and L_{JS} represents the JS-Loss.

IV. Experiments and Discussions

A. Evaluation Protocols

To evaluate the effectiveness of our proposed method, we follow the same evaluation metric of existing literatures [13], [14], [15], [16], [17], [18], [19], [20], [45], where Recall@K (R@K) and average precision (AP) are measured. R@K measures the percentage of query images where at least one of the top-k recalled gallery images is correctly retrieved. AP assesses retrieval precision by calculating the area under the precision–recall curve, capturing both precision and recall to evaluate overall performance.

B. Datasets

In this work, we use two well-known CVGL datasets, University-1652 and SUES-200.

University-1652 [8] is a large-scale multiview, multisource dataset consisting of satellite-, drone-, and street-view images. It contains samples from 1652 university campus buildings across 72 universities worldwide and is widely used for research in CVGL. Notably, it is the first dataset to introduce drone-view images into CVGL, posing new challenges for models. These challenges include how to accurately retrieve drone images using satellite images and how to perform precise geo-localization of drone images using satellite images.

SUES-200 [40] is a well-designed CVGL dataset that samples images of different buildings at various altitudes on the campus of Shanghai University of Engineering Science and nearby parks. For each location and altitude, there is one corresponding satellite image and 50 drone images. SUES200 captures drone images at different heights, presenting new challenges for model representation ability. In particular, at an altitude of 150 m, the narrow field of view of the drone means that it cannot capture the entire building in a single frame, demanding higher adaptability from the models to handle complex and dynamic environments.

C. Implementation Details

Our method is implemented on Ubuntu 22.04, PyTorch Framework, and experiments are conducted on an Nvidia RTX 4090 GPU with 24-GB VRAM.

TABLE I
PERFORMANCE COMPARISON ON THE UNIVERSITY-1652 DATASET. THE BEST RESULTS ARE IN BOLD

Method	Publication	Test Image size	Drone-Satellite		Satellite-Drone	
			R@1	AP	R@1	AP
Instance Loss [8]	MM' 20	256×256	58.49	63.13	71.18	58.74
LPN [13]	TCSVT' 22	512×512	75.93	79.14	86.45	74.79
RK-Net [9]	TIP' 22	256×256	77.60	80.55	86.59	75.96
FSRA [16]	TCSVT' 22	256×256	82.25	84.82	87.87	81.53
TransFG [45]	TGRS' 24	256×256	84.01	86.31	90.16	84.61
MJRLIFS [20]	TGRS' 24	256×256	86.06	88.08	91.44	85.73
GeoFormer [19]	JSTARS' 24	224×224	89.08	90.83	92.30	88.54
SeGCN [18]	JSTARS' 24	256×256	89.18	90.89	94.29	89.65
MCCG [41]	TCSVT' 23	384×384	89.64	91.32	94.30	89.39
SDPL [14]	TCSVT' 24	256×256	90.16	91.64	93.58	89.45
CCR [15]	TCSVT' 24	384×384	92.54	93.78	95.15	91.80
Sample4Geo [42]	ICCV' 23	384×384	92.65	93.81	96.43	93.79
SRLN [46]	TGRS' 24	384×384	92.70	93.77	95.14	91.97
CAMP [21]	TGRS' 24	384×384	94.46	95.38	96.15	92.72
DAC [44]	TCSVT' 24	384×384	94.67	95.50	96.43	93.79
Ours (QDFL)	-	280×280	**95.00**	**95.83**	**97.15**	**94.57**

1) Model Details: For the AQEU module, the number of queries is set to 8. In the FFU module, the kernel sizes for height and width fusion were set to $(1, 3)$ and $(3, 1)$, respectively. The number of the hidden layer neuron was configured as dim/3, where dim represents the input feature map dimension.

2) Training Details: We trained our models using the SGD optimizer with a momentum of 0.9, a weight decay of 0.0005, and a total of 160 epochs. The batch size was set to 24. The initial learning rate (lr) was set to 0.03. We set the initial learning rate for the adapters to 0.3 × lr for training stability. The learning rate was warmed up for the first 175 steps, after which a cosine scheduler with $T_{\max} = 160$ was applied. We use 224 × 224 images for training. Data augmentations including padding, flipping, color jittering, and cropping were applied.

3) Evaluating Details: We evaluated our method using an image resolution of 280 × 280 on both the University-1652 dataset and the SUES-200 dataset. We opted for this resolution instead of common resolutions such as 256 × 256 or 512 × 512 since the DINOv2 model partitions the input image into 14 × 14 patches. Furthermore, Euclidean distance was used to compute the similarity between query and gallery images.

D. Comparison to the SOTA Methods

1) Results on University-1652: Table I presents the performances of QDFL on the University-1652 dataset, achieving R@1 of 95.00% and AP of 95.83% for the drone–satellite task, and R@1 of 97.15% and AP of 94.57% for the satellite–drone task. For the both satellite–drone and drone–satellite tasks, QDFL surpasses all the listed methods in overall performance. It is worth noting that QDFL, which uses images with resolution of 280 × 280, shows better performance compared with SOTA models which operate images with a higher resolution of 384 × 384. In this experiment, QDFL shows its advantages in handling images with lower resolutions.

2) Results on SUES-200: As shown in Table II, our method demonstrates outstanding performance on the SUES-200 dataset. For the drone–satellite task, QDFL achieves R@1 scores of 93.97%, 98.25%, 99.30%, and 99.31% and AP scores of 95.42%, 98.67%, 99.48%, and 99.48% at heights of 150, 200, 250, and 300 m, respectively. In the satellite–drone task, QDFL reaches R@1 scores of 98.75%, 98.75%, 100%, and 100% and AP scores of 95.10%, 97.92%, 99.07%, and 99.07% at the same respective heights. These results highlight the robustness and versatility of the QDFL across varying heights. Despite using a relatively low resolution, our method not only maintains competitive performance but also surpasses SOTA in several metrics.

E. Generalization Performance on Cross-Dataset Task

In real-world applications, the CVGL model is trained before being deployed on onboard computers. During inference, it operates in regions beyond the training dataset in most cases. Thus, evaluating the generalizability of QDFL across different datasets is vital to ensure robustness and reliability in diverse scenarios. We followed the experimental setup in [21] and [44], using the University-1652 dataset for training and the SUES-200 test set for evaluation. We selected SOTA methods for comparison, including MCCG, Sample4Geo, CAMP, DAC, and FSRA. For the first three methods, we used reported results from [21] and [44]. In addition, we reproduced the FSRA following the settings in [16]. The results in Table III indicate that QDFL can achieve good localization performance without prior fine-tuning of the scene, and its generalizability exceeds that of the compared SOTA methods. Concretely, in the 150-m task, our method achieves R@1 of 85.15% and AP of 88.16% in the drone–satellite task, and 95.00% in R@1 and 83.06% in AP, in the satellite–drone task. Compared with DAC [44], our method achieves remarkable improvement in generalization performances, with an average increase of 5.57% in R@1 and 4.81% in AP for the drone–satellite task and 2.19% in R@1 and 2.80% in AP for the satellite–drone task. Similarly, QDFL outperforms CAMP [21] with an average improvement of 4.87% in R@1 and 4.12% in AP for the drone–satellite task and 2.50% in R@1 and 4.10% in AP

TABLE II

PERFORMANCE COMPARISON BETWEEN DIFFERENT METHODS ON DRONE-TO-SATELLITE AND SATELLITE-TO-DRONE RETRIEVAL TASKS

Drone-Satellite

Method	Publication	Image Size	150m		200m		250m		300m	
			R@1	AP	R@1	AP	R@1	AP	R@1	AP
SUES-200 [40]	TCSVT' 23	384×384	59.32	64.93	62.30	67.24	71.35	75.49	77.17	67.80
LPN [13]	TCSVT' 22	512×512	61.58	67.23	70.85	75.96	80.38	83.80	81.47	84.53
FSRA [16]	TCSVT' 22	256×256	68.25	73.45	83.00	85.99	90.68	92.27	91.95	91.95
MJRLIFS [20]	TGRS' 24	256×256	77.57	81.30	89.50	91.40	92.58	94.21	97.40	97.92
SeGCN [18]	JSTARS' 24	256×256	90.80	92.32	91.93	93.41	92.53	93.90	93.33	94.61
MCCG [41]	TCSVT' 23	384×384	82.22	85.47	89.38	91.41	93.82	95.04	95.07	96.20
SDPL [14]	TCSVT' 24	256×256	82.95	85.82	92.73	94.07	96.05	96.69	97.83	98.05
CCR [15]	TCSVT' 24	384×384	87.08	89.55	93.57	94.90	95.42	96.28	96.82	97.39
Sample4Geo [42]	ICCV' 23	384×384	92.60	96.38	97.38	97.81	98.28	98.64	99.18	99.36
SRLN [46]	TGRS' 24	384×384	89.90	91.90	94.32	95.65	95.92	96.79	96.37	97.21
CAMP [21]	TGRS' 24	384×384	95.40	96.38	97.63	98.16	98.05	98.45	**99.33**	99.46
DAC [44]	TCSVT' 24	384×384	**96.80**	**97.54**	97.48	97.97	98.20	98.62	97.58	98.14
Ours (QDFL)	-	280×280	93.97	95.42	**98.25**	**98.67**	**99.30**	**99.48**	99.31	**99.48**

Satellite-Drone

Method	Publication	Image Size	150m		200m		250m		300m	
			R@1	AP	R@1	AP	R@1	AP	R@1	AP
SUES-200	TCSVT' 23	384×384	82.50	58.95	85.00	62.56	88.75	69.96	96.25	84.16
LPN	TCSVT' 22	512×512	83.75	66.78	88.75	75.01	92.50	81.34	92.50	85.72
FSRA	TCSVT' 22	256×256	83.75	76.67	90.00	85.34	93.75	90.17	95.00	92.03
MJRLIFS	TGRS' 24	256×256	93.75	79.49	97.50	90.52	97.50	96.03	**100.00**	97.66
SeGCN	JSTARS' 24	256×256	93.75	92.45	95.00	93.65	96.25	94.39	97.50	94.55
MCCG	TCSVT' 23	384×384	93.75	89.72	93.75	92.21	96.25	96.14	98.75	96.64
SDPL	TCSVT' 24	256×256	93.75	83.75	96.25	92.42	97.50	95.65	96.25	96.17
CCR	TCSVT' 24	384×384	92.50	88.54	97.50	95.22	97.50	97.10	97.50	97.49
Sample4Geo	ICCV' 23	384×384	97.50	93.63	**98.75**	96.70	98.75	98.28	98.75	98.05
SRLN	TGRS' 24	384×384	93.75	93.01	97.50	95.08	97.50	96.52	97.50	96.71
CAMP	TGRS' 24	384×384	96.25	93.69	97.50	96.76	98.75	98.10	**100.00**	98.85
DAC	TCSVT' 24	384×384	97.50	94.06	**98.75**	96.66	98.75	98.09	98.75	97.87
Ours (QDFL)	-	280×280	**98.75**	**95.10**	**98.75**	**97.92**	**100.00**	**99.07**	**100.00**	**99.07**

TABLE III

COMPARISON TO SOTA RESULTS IN CROSS-DATASET TRANSFERABILITY

Drone-Satellite

Method	Image Size	150m		200m		250m		300m		Average	
		R@1	AP	R@1	AP	R@1	AP	R@1	AP	R@1$_{\text{offset}}$	AP$_{\text{offset}}$
FSRA [16]	256×256	46.17	53.22	59.10	64.92	68.90	73.69	72.08	76.53	$61.56_{-31.65}$	$67.09_{-27.51}$
MCCG [41]	256×256	57.62	62.80	66.83	71.60	74.25	78.35	82.55	85.27	$70.31_{-22.90}$	$74.50_{-20.10}$
Sample4Geo [42]	384×384	70.05	74.93	80.68	83.90	87.35	89.72	90.03	91.91	$82.02_{-11.18}$	$85.12_{-9.49}$
CAMP [21]	384×384	78.90	82.38	86.83	89.28	91.95	93.63	95.68	96.65	$88.34_{-4.87}$	$90.49_{-4.12}$
DAC [44]	384×384	76.65	80.56	86.45	89.00	92.95	94.18	94.53	95.45	$87.65_{-5.57}$	$89.80_{-4.81}$
Ours (QDFL)	280×280	**85.15**	**88.16**	**93.05**	**94.51**	**96.62**	**97.29**	**98.02**	**98.45**	$\mathbf{93.21_{-0}}$	$\mathbf{94.60_{-0}}$

Satellite-Drone

Method	Image Size	150m		200m		250m		300m		Average	
		R@1	AP	R@1	AP	R@1	AP	R@1	AP	R@1$_{\text{offset}}$	AP$_{\text{offset}}$
FSRA	256×256	52.50	43.36	61.25	53.22	70.00	58.61	71.25	62.61	$63.75_{-32.19}$	$54.45_{-37.27}$
MCCG	256×256	61.25	53.51	82.50	67.06	81.25	74.99	87.50	80.20	$78.13_{-17.81}$	$68.94_{-22.78}$
Sample4Geo	384×384	83.75	73.83	91.25	83.42	93.75	89.07	93.75	90.66	$90.63_{-5.31}$	$84.25_{-7.48}$
CAMP	384×384	87.50	78.98	95.00	87.05	95.00	91.05	96.25	93.44	$93.44_{-2.50}$	$87.63_{-4.10}$
DAC	384×384	87.50	79.87	96.25	88.98	95.00	92.81	96.25	94.00	$93.75_{-2.19}$	$88.92_{-2.80}$
Ours (QDFL)	280×280	**95.00**	**83.06**	**96.25**	**91.60**	**96.25**	**95.72**	**96.25**	**96.50**	$\mathbf{95.94_{-0}}$	$\mathbf{91.72_{-0}}$

for the satellite–drone task. In general, our method delivers an average improvement of 3.87% in R@1 and 3.80% in AP over DAC 3.69% in R@1, and 4.10% in AP over CAMP on both the tasks. This evaluation showcases that our method surpasses all the current SOTA methods in most altitude tasks and exhibits excellent generalizability across datasets.

F. Comparison of Trainable Parameters and Performance

We further compare the trainable parameters, computational efficiency, and performance of QDFL with several SOTA methods. The QDFL achieves an optimal balance between model capacity and training efficiency. While the complete architecture contains 108.87 M parameters in total, only

TABLE IV

Comparison of Parameters, Computational Complexities, and Performances on University-1652

Method	Backbone	Parameters		Image Size	Test FLOPs	Drone-Satellite		Satellite-Drone	
		Total	Trainable			R@1	AP	R@1	AP
FSRA [16]	ViT-S	51.80M	51.80M	256 × 256	12.29G	82.25	84.82	87.87	81.53
SDPL [14]	Resnet-50	40.51M	40.51M	512 × 512	34.85G	85.19	87.43	89.30	82.75
GeoFormer [19]	E-Swin-L	157.00M	157.00M	224 × 224	37.70G	89.08	90.83	92.30	88.54
CCR [15]	ConvNeXt-B	158.06M	158.06M	384 × 384	45.20G	92.54	93.78	95.15	91.80
DAC [44]	ConvNeXt-B	96.50M	96.50M	384 × 384	45.12G	94.67	95.50	96.43	93.79
Ours (QDFL)	Adapted-DINOv2-B	108.87M	**22.30M**	280 × 280	40.12G	**95.00**	**95.83**	**97.15**	**94.57**

TABLE V

Ablation Experimental Results of the Proposed QDFL Method

Method				Drone-Satellite		Satellite-Drone	
PETL	AQEU	FFU	CLS	R@1	AP	R@1	AP
				87.25	89.31	92.15	85.97
✓				90.28	91.86	94.44	90.30
✓	✓			94.23	95.19	95.72	93.70
✓	✓	✓		94.42	95.38	96.72	93.68
✓	✓	✓	✓	**95.00**	**95.83**	**97.15**	**94.57**

TABLE VI

Effect of Learnable Queries' Number

Queries Number	Drone-Satellite		Satellite-Drone	
	R@1	AP	R@1	AP
2	93.96	94.96	95.58	93.58
4	94.81	95.63	96.15	94.36
8	**95.00**	**95.83**	**97.15**	**94.57**
16	94.15	95.12	96.01	93.78
32	94.64	95.52	96.43	94.37
64	94.53	95.37	97.00	94.26

22.30 M parameters (20.76% of total capacity) require gradient updates during training. Table IV shows our method demonstrates superior performance while requiring fewer trainable parameters compared with other methods. Although QDFL has a relatively higher number of parameters compared to smaller backbone methods such as FSRA, it maintains the lowest number of trainable parameters during the training phase. For larger models such as CCR, the efficiency of QDFL originates from its adapter-based PETL, which adapts the DINOv2-B foundation model to CVGL with only 14.19 M trainable parameters. In addition, compared with the postbackbone component in CCR, QDFL reduces trainable parameters by 88.32% (8.11 versus 69.47 M) through a unified 512D feature representation and the elimination of CCR's MCB classifiers, which account for 67.12 M parameters. Another large model, GeoFormer, leverages the E-Swin-L foundation model as its backbone, resulting in a significantly larger overall parameter count. In contrast, despite the use of adapted foundation model in QDFL, the computational burden remains acceptable.

G. Ablation Studies

1) Effect of Each Component: First, to verify the effectiveness of the PETL method that we introduced, we train the vanilla DINOv2-B model by freezing all the parameters except the last two layers. We adopt both the serial and parallel adapters into the backbone. All the postbackbone branches are disabled except a GeM layer [62] is adopted to extract a 512D descriptor. As shown in Table V, using adapters substantially improves the baseline performance of the R@1 and AP metrics in two tasks. Compared with directly fine-tuning the backbone, adapter-based PETL method could be more capable of adapting prior knowledge to downstream tasks. The introduced AQEU and FFU prominently boost the performance. Compared with the baseline model, incorporating AQEU led to a 3.95% improvement in R@1 and a 3.33% increase in AP for the drone–satellite task, a 1.28% improvement in R@1 and 3.40% rise in AP for the satellite–drone task, respectively. On top of these improvements, FFU further enhances performance, especially in R@1 for the satellite–drone task. Finally, the [CLS] token further enhances the overall performance.

2) Effect of Learnable Queries Number: We vary the number of queries initialized in AQEU, and the results are shown in Table VI. When using a small number of learnable queries, the model struggles to capture sufficient viewpoint-invariant features. As the number of learnable queries increases, performance improves accordingly. However, we observe that beyond a certain point (eight queries), further increasing the number of learnable queries leads to a slight degradation in performance. The reason for degradation is likely due to the model becoming overly complex and leading to overfitting, resulting in the extraction of redundant or less informative features, which dilute the effectiveness of the key features that drive performance improvements.

3) Effect of the Backbone: To evaluate performance variation across different backbones and verify the robustness of the proposed method, we conducted experiments on the University-1652 dataset using different backbones. We selected three representative CNN-based backbones, ResNet-50, ResNet-101, and ConvNeXt-T, as well as three ViT-based backbones: ViT-S, DINOv2-S, and DINOv2-B. It is worth noting that all the ViT-based models are equipped with adapters, and we used vanilla pretrained models for the CNNs. Empirical results in Table VII show the effectiveness of our proposed QDFL method. All the experiments are conducted under the same hyperparameter setting mentioned in Section IV-C2. When the DINOv2-B backbone of the QDFL is replaced with its smaller counterpart, DINOv2-S, the performance of QDFL experiences a decline. The performance degradation can be attributed to two primary factors:

TABLE VII

ABLATION EXPERIMENTAL RESULTS OF DIFFERENT BACKBONES. † WE CONDUCTED TRAINING USING THE DEFAULT HYPERPARAMETERS, WITH THE EXCEPTION OF ADJUSTING THE WARMUP STEPS TO 350 (ORIGINALLY 175 STEPS)

Backbone	Image Size	Drone-Satellite		Satellite-Drone	
		R@1	AP	R@1	AP
Resnet-50	512×512	79.55	82.56	89.44	79.79
Resnet-101	512×512	83.83	86.39	90.44	82.86
ConvNeXt-T	384×384	92.03	93.30	95.01	91.36
ViT-S(Adapted)	256×256	86.46	88.54	93.01	85.80
DINOv2-S(Adapted)	280×280	90.37	91.92	93.72	89.55
DINOv2-S(Adapted)†	280×280	92.24	93.42	95.44	91.96
DINOv2-B(Adapted)	280×280	**95.00**	**95.83**	**97.15**	**94.57**

TABLE VIII

PERFORMANCE COMPARISON OF QDFL TO DIFFERENT INPUT SIZES ON UNIVERSITY-1652

Test Image size	Drone-Satellite		Satellite-Drone	
	R@1	AP	R@1	AP
224×224	94.70	95.56	96.15	94.14
252×252	94.91	95.75	96.43	94.51
280×280	95.00	95.83	**97.15**	**94.57**
322×322	**95.16**	**95.95**	96.72	94.51
378×378	94.90	95.76	96.72	94.42

1) limited model capacity. Specifically, the smaller architecture of DINOv2-S, with fewer parameters (22.05 M) and narrower channels (384D) compared with DINOv2-B, inherently restricts its representational power and 2) suboptimal hyperparameter tuning. The initial use of default hyperparameters, such as a 175-step warmup, was not tailored for the smaller model like DINOv2-S. By extending the warmup steps to 350, the performance gap between DINOv2-S and DINOv2-B was reduced.

4) *Effect of the Image Size:* We conducted additional experiments to explore the impact of different input image sizes on model performance. As image resolution increases, so do the computational complexity and memory requirements. Our model demonstrates outstanding accuracy even at lower resolutions. The results in Table VIII show that as resolution increases from 224 × 224 to 378 × 378, model performance consistently improves. However, further increasing the resolution not only fails to enhance performance but also slightly degrades it.

5) *Effect of Fine-Tuning Methods:* We investigated the effects of different fine-tuning strategies and evaluated their impact on model performance, and the results are shown in Table IX. Initially, we froze all the parameters in the DINOv2-B backbone, training only the QDFL, which yielded solid performance on both the satellite–drone and drone–satellite tasks. Fine-tuning only a few layers resulted in improved performances. However, when we fine-tuned the entire backbone, the performance declined, as fully fine-tuning breaks the prior knowledge established in the pretrained model. We also tested the results of using adapters in the backbone. Only adding serial or parallel adapters could enhance the performance, and it can be seen that parallel adapters have a greater performance improvement than serial adapters. When two types of adapters are used together, their effectiveness is further enhanced.

TABLE IX

PERFORMANCE COMPARISON USING DIFFERENT FINE-TUNING METHODS

Method		Train Param.	Drone-Satellite		Satellite-Drone	
			R@1	AP	R@1	AP
Frozen Backbone		8.1M	84.99	87.44	91.58	82.82
Fine-tune	All	93.2M	79.39	82.03	86.73	79.13
	9-12 Layers	36.5M	93.23	94.35	96.29	93.60
	11-12 Layers	22.3M	91.65	93.07	94.15	91.26
Adapters	Only Serial	14.0M	93.87	94.89	95.01	93.12
	Only Parallel	15.2M	94.57	95.48	96.43	94.26
	Both	22.3M	**95.00**	**95.83**	**97.15**	**94.57**

TABLE X

ABLATION STUDY TO VERIFY THE EFFECTS OF THE LOSS

Method				Drone-Satellite		Satellite-Drone	
Triplet	MS	S-KL	JS	R@1	AP	R@1	AP
✓				93.34	94.39	95.86	92.52
	✓			94.79	95.66	96.72	94.38
	✓	✓		94.81	95.70	96.43	**94.58**
	✓		✓	**95.00**	**95.83**	**97.15**	94.57

TABLE XI

EFFECT OF THE FEU OUTPUT DIMENSION

Feature Dim.	Drone-Satellite		Satellite-Drone	
	R@1	AP	R@1	AP
128	92.83	93.96	95.01	92.36
256	94.47	95.33	96.29	93.99
512	95.00	95.83	**97.15**	94.57
1024	94.73	95.61	96.72	94.63
2048	**95.12**	**95.93**	97.00	**94.90**

6) *Effect of MS-Loss and JS-Loss:* We investigated the impact of various loss functions on model retrieval performance, using MS-Loss and JS-Loss, and compared them with triplet loss and symmetric-KL loss (S-KL Loss). For consistency, the margins for both triplet loss and MS-Loss were set at 0.4. As shown in Table X, MS-Loss outperformed triplet loss, demonstrating improvements in retrieval metrics. Specifically, in the drone–satellite task, MS-Loss resulted in a 1.45% increase in R@1 and a 1.27% increase in AP. In the satellite–drone task, the gains were more pronounced, with a 0.86% increase in R@1 and a 1.86% increase in AP. Further analysis reveals that JS-Loss aligns feature distributions from different perspectives more effectively than S-KL Loss. JS-Loss reduces biased estimates and performance discrepancies between tasks, resulting in improved AP metrics.

H. Effect of FEU Output Dimension

To further examine the impact of feature vector dimension on performance, we select extraction dimensions of FEU ranging from 128 to 2048. The results are presented in Table XI. It is evident that the QDFL achieves optimal performance at 512 dimensions. While increasing the dimensionality slightly enhances the overall metric, it also adds to inference burden.

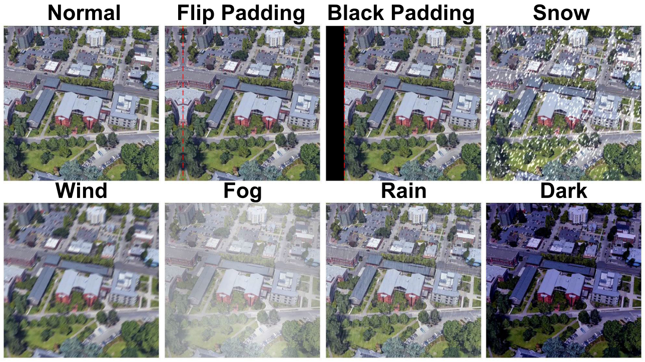


Fig. 7. Examples of black, flip padding, and synthesized results under various weather conditions.

TABLE XII

AP Variances of the Proposed QDFL Under Different Padding Sizes

Pad Pixel	Black Padding(AP)		Flip Padding(AP)	
	Ours (QDFL)	FSRA [16]	Ours (QDFL)	FSRA
0	**95.83**$_{-0}$	84.77$_{-0}$	**95.83**$_{-0}$	84.77$_{-0}$
10	**95.66**$_{-0.17}$	84.13$_{-0.64}$	**95.46**$_{-0.37}$	84.19$_{-0.58}$
20	**95.13**$_{-0.70}$	82.70$_{-2.07}$	**94.36**$_{-1.47}$	82.26$_{-2.51}$
30	**93.93**$_{-1.90}$	80.03$_{-4.74}$	**92.66**$_{-3.17}$	78.46$_{-6.31}$
40	**91.87**$_{-3.96}$	76.41$_{-8.36}$	**89.10**$_{-6.73}$	73.13$_{-11.64}$
50	**88.60**$_{-7.23}$	71.60$_{-13.17}$	**83.15**$_{-12.68}$	66.07$_{-18.70}$
60	**84.27**$_{-11.56}$	52.09$_{-29.08}$	**76.09**$_{-19.74}$	57.76$_{-26.81}$

I. Robustness of Proposed Method

To investigate the robustness of QDFL in handling positional shifting tasks, we implemented black padding and flip padding experiments, as depicted in Fig. 7. The results are demonstrated in Table XII, our method consistently outperforms FSRA [16] in terms of maintaining higher AP scores under increasing padding perturbations. At moderate 30-pixel padding sizes, QDFL demonstrates better robustness, with AP scores of 93.93% and 92.66% for black and flip padding, significantly surpassing FSRA. Even at severe padding sizes of 60 pixels, QDFL achieves an AP of 84.27% for black padding and 76.09% for flip padding.

To further examine the performance variations of the QDFL under different weather conditions, we synthesized weathers, including snow, wind, fog, rain, and low visibility, as depicted in Fig. 7. As shown in Table VIII, the results indicate that rainy and snowy days have a significant impact on the accuracy of the model, while foggy, dark, and windy days have a relatively small impact on the model performance. In general, our method is robust when facing multiple weather conditions.

J. Visualization of Qualitative Results

1) Qualitative Comparison of Recall Results: As illustrated in Fig. 8, we qualitatively compared the retrieved results on QDFL, LPN, FSRA, SDPL, and CCR using the University-1652 dataset. The descriptors learned by QDFL for recalling the ground truth are robust enough, even in scenarios with large variations in illumination, temporal, and viewpoint. The first four rows demonstrate cases where QDFL successfully recalls the correct location in two subtasks, while all other methods fail. The final row presents a challenging example where all the methods fail. In concrete, rows 1–3 of Fig. 8 demonstrate that even when the roof color varies due to different capture angles or seasons, our method retrieves the correct results, while other methods tend to focus on matching similar colors rather than identifying discriminative features

TABLE XIII

Effect of Proposed QDFL Under Different Weather Conditions

Weather Conditions	Drone-Satellite		Satellite-Drone	
	R@1	AP	R@1	AP
Normal	95.08	95.89	96.72	94.69
Fog	93.95	95.01	96.43	92.79
Rain	91.94	93.20	96.01	91.64
Snow	92.19	93.43	96.43	91.69
Dark	94.18	95.15	96.72	93.00
Wind	93.63	94.67	96.58	92.17

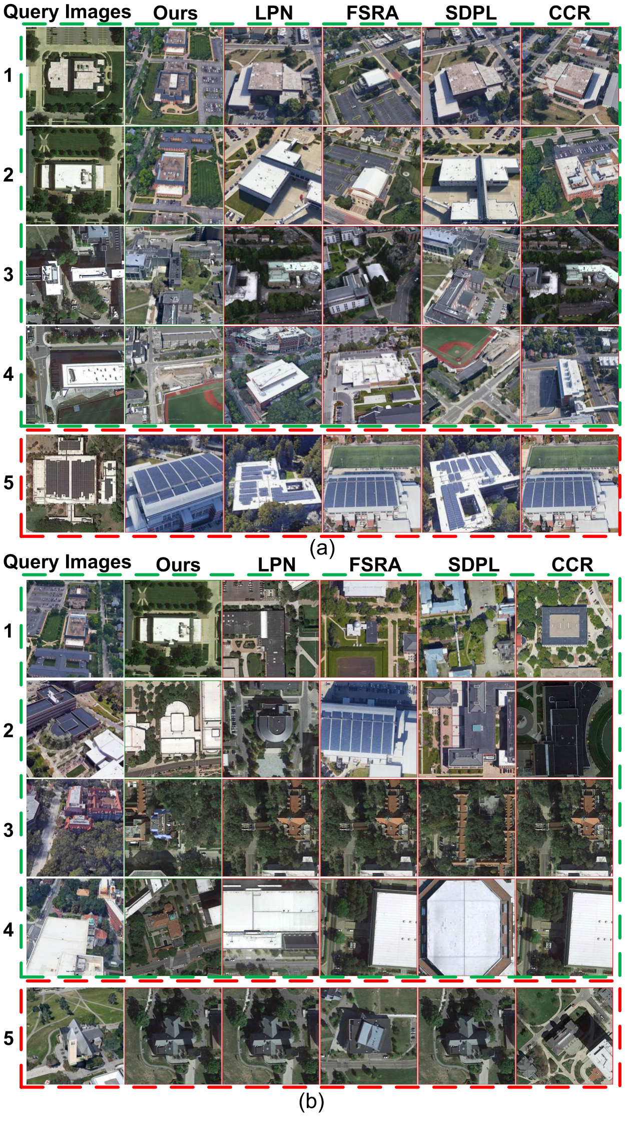


Fig. 8. Recall@1 results obtained with QDFL, LPN, FSRA, SDPL, and CCR. (a) Results of drone localization on University-1652. (b) Results of drone navigation on University-1652. The green dashed box highlights instances where QDFL successfully retrieved the correct cases, whereas other methods failed. The red dashed box indicates that all the methods, including QDFL, failed to retrieve the correct results.

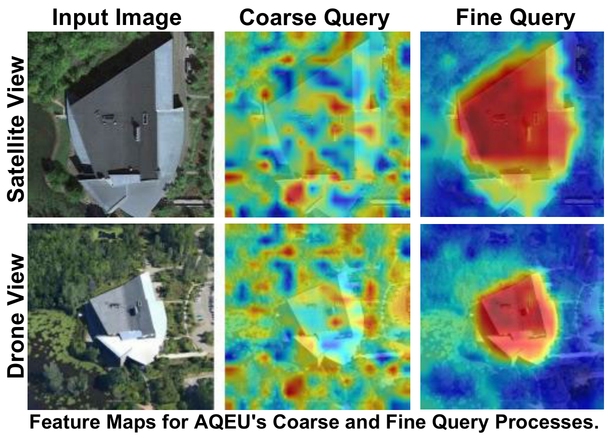


Fig. 9. Comparison of heatmaps processed by AQEU and FFU, showing input images, heatmaps before the coarse query process, and heatmaps processed by FFU from left to right.

within the image. Row 4 in Fig. 8(a) illustrates a scenario where the target building exists in the satellite image but has been dismantled in the drone view. Our method also retrieves the correct result, while other methods retrieve images with the wrong target building at the center. Row 4 in Fig. 8(b) presents a case where the drone captured an image of the target building from an angle where the surrounding buildings block it, only a small portion of the building is visible, but our method still retrieved the corresponding satellite image. Both the cases in row 5 illustrate that all the methods including our method fail to retrieve the correct result. These cases show perceptual aliasing phenomena in the dataset, where the images at different locations are very similar to each other, particularly in terms of prominent landmarks such as solar panels and pointed buildings.

2) Feature Maps for the Coarse and Fine Query: Fig. 9 presents a comparison of feature maps. For the coarse and fine query in AQEU, coarse feature maps are processed by the SA mechanism within AQEU, while FFU processes fine feature maps. During the coarse query process, AQEU emphasizes global contexts, distributing attention equally across all the regions of the image. In contrast, FFU refines this coarse query feature map, focusing more on the unique features within the image.

K. Limitations and Future Work

While our method achieves improvements in drone-view CVGL tasks and demonstrates strong generalization capabilities in cross-dataset experiments, there are still some limitations that deserve further exploration. First, QDFL relies solely on single-modal data (i.e., images) and follows an image retrieval-based paradigm. When the input image lacks distinctive clues or salient landmarks, the model is prone to recalling incorrect matches with visually similar sites. In future research, we will explore the fusion of multimodal information, such as geospatial text metadata [63], video streams [64], and 3D-scene [5], [65]. Textual attributes offer semantic constraints while successive video frames provide motion parallax cues for scene understanding. 3D-scene offers a rich source of spatial consistency data that can enhance localization accuracy by leveraging spatial geometric relationships and scene distance information. Second, the large parameter size of our QDFL (DINOv2-B) model imposes a high computational burden, resulting in difficult deployment on resource-constrained edge devices. To address this, future work could explore model compression techniques, such as knowledge distillation [66], to reduce computational overhead while maintaining excellent performance.

V. Conclusion

In this work, we propose a novel CVGL method named QDFL, which incorporates two innovative units: AQEU and FFU. AQEU learns a set of query vectors that contains the mutual feature patterns from cross-view images, querying robust viewpoint-invariant feature vectors in a coarse-to-fine manner. The FFU integrates features from both the spatial and channel dimensions, generating feature maps that emphasize salient landmarks, further enhancing the fine-grained features. With the collaborative interaction of AQEU and FFU, robust viewpoint-invariant feature vectors are queried from feature map. In addition, we use PETL paradigm by integrating tunable adapters into the frozen pretrained backbone, maintaining feature representation capabilities of foundation models while enabling seamless adaptation to CVGL task. Furthermore, to exploit the use of similarity relations within mini-batches, we adopt MS-Loss to CVGL task. To further enhance performance, we use JS-Loss to align feature distributions across different perspectives. Our method achieves SOTA performance on University-1652 and SUES-200. Moreover, our method demonstrates impressive generalizability in cross-dataset experiments. In the future, we will focus on multimodal data-enabled CVGL methods and the research of compression and lightweight deployment techniques for high-performance CVGL models.

Acknowledgment

The authors sincerely thank the contributors of the University-1652 and SUES-200 datasets for their support in this research.

References

[1] H. D. Yoo and S. M. Chankov, “Drone-delivery using autonomous mobility: An innovative approach to future last-mile delivery problems,” in *Proc. IEEE Int. Conf. Ind. Eng. Eng. Manage. (IEEM)*, Dec. 2018, pp. 1216–1220, doi: 10.1109/IEEM.2018.8607829.

[2] G. E. Jan, C.-C. Lin, C.-W. Hsu, C.-M. Kung, H.-C. Hsieh, and Y.-H. Wong, “Innovative search and rescue methods using drones: An algorithm combining 3D modeling and laser ranging scanning,” in *Proc. IEEE Int. Conf. e-Bus. Eng. (ICEBE)*, Nov. 2023, pp. 297–303, doi: 10.1109/icebe59045.2023.00054.

[3] G. De Masi and E. Ferrante, “Quality-dependent adaptation in a swarm of drones for environmental monitoring,” in *Proc. Adv. Sci. Eng. Technol. Int. Conf. (ASET)*, Feb. 2020, pp. 1–6, doi: 10.1109/ASET48392.2020.9118235.

[4] A. N. J. Kukunuri and D. Singh, “Efficient application of drone with satellite data for early-stage wheat detection: For precision agriculture monitoring,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2022, pp. 4388–4391, doi: 10.1109/IGARSS46834.2022.9883266.

[5] W. Zhang, Q. Li, Y. Yuan, and Q. Wang, “Visual consistency enhancement for multiview stereo reconstruction in remote sensing,” *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5646011, doi: 10.1109/TGRS.2024.3482697.

[6] S. Ji, S. Wei, and M. Lu, "Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 1, pp. 574–586, Jan. 2019, doi: 10.1109/TGRS.2018.2858817.

[7] X. Tian, J. Shao, D. Ouyang, and H. T. Shen, "UAV-satellite view synthesis for cross-view geo-localization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 7, pp. 4804–4815, Jul. 2022, doi: 10.1109/TCSVT.2021.3121987.

[8] Z. Zheng, Y. Wei, and Y. Yang, "University-1652: A multi-view multi-source benchmark for drone-based geo-localization," in *Proc. ACM Multimedia*, Oct. 2020, pp. 1395–1403, doi: 10.1145/3394171.3413896.

[9] J. Lin et al., "Joint representation learning and keypoint detection for cross-view geo-localization," *IEEE Trans. Image Process.*, vol. 31, pp. 3780–3792, 2022, doi: 10.1109/TIP.2022.3175601.

[10] Y. Zhu, B. Sun, X. Lu, and S. Jia, "Geographic semantic network for cross-view image geo-localization," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4704315, doi: 10.1109/TGRS.2021.3121337.

[11] T. Wang, Z. Zheng, Z. Zhu, Y. Sun, C. Yan, and Y. Yang, "Learning cross-view geo-localization embeddings via dynamic weighted decorrelation regularization," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5647112, doi: 10.1109/TGRS.2024.3491757.

[12] W.-J. Ahn, S.-Y. Park, D.-S. Pae, H.-D. Choi, and M.-T. Lim, "Bridging viewpoints in cross-view geo-localization with Siamese vision transformer," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 4707312, doi: 10.1109/TGRS.2024.3429570.

[13] T. Wang et al., "Each part matters: Local patterns facilitate cross-view geo-localization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 2, pp. 867–879, Feb. 2022, doi: 10.1109/TCSVT.2021.3061265.

[14] Q. Chen et al., "SDPL: Shifting-dense partition learning for UAV-view geo-localization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 11, pp. 11810–11824, Nov. 2024, doi: 10.1109/TCSVT.2024.3424196.

[15] H. Du, J. He, and Y. Zhao, "CCR: A counterfactual causal reasoning-based method for cross-view geo-localization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 11, pp. 11630–11643, Nov. 2024, doi: 10.1109/TCSVT.2024.3425509.

[16] M. Dai, J. Hu, J. Zhuang, and E. Zheng, "A transformer-based feature segmentation and region alignment method for UAV-view geo-localization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 7, pp. 4376–4389, Jul. 2022, doi: 10.1109/TCSVT.2021.3135013.

[17] P. Wang, Z. Yang, X. Chen, and H. Xu, "A transformer-based method for UAV-view geo-localization," in *Proc. 32nd Int. Conf. Artif. Neural Netw.*, Heraklion, Greece. Berlin, Germany: Springer, Jan. 2023, pp. 332–344, doi: 10.1007/978-3-031-44223-0_27.

[18] X. Liu, Z. Wang, Y. Wu, and Q. Miao, "SeGCN: A semantic-aware graph convolutional network for UAV geo-localization," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 6055–6066, 2024, doi: 10.1109/JSTARS.2024.3370612.

[19] Q. Li et al., "GeoFormer: An effective transformer-based Siamese network for UAV geolocalization," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 9470–9491, 2024, doi: 10.1109/JSTARS.2024.3392812.

[20] F. Ge et al., "Multibranch joint representation learning based on information fusion strategy for cross-view geo-localization," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5909516, doi: 10.1109/TGRS.2024.3378453.

[21] Q. Wu, Y. Wan, Z. Zheng, Y. Zhang, G. Wang, and Z. Zhao, "CAMP: A cross-view geo-localization method using contrastive attributes mining and position-aware partitioning," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5637614, doi: 10.1109/TGRS.2024.3448499.

[22] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. ICLR*, Jan. 2020, pp. 1–21.

[23] A. Radford et al., "Learning transferable visual models from natural language supervision," 2021, *arXiv:2103.00020*.

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

[25] K. James et al., "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 13, pp. 3521–3526, Mar. 2017, doi: 10.1073/pnas.1611835114.

[26] X. Wang, X. Han, W. Huang, D. Dong, and M. R. Scott, "Multi-similarity loss with general pair weighting for deep metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 5022–5030, doi: 10.1109/CVPR.2019.00516.

[27] N. Houlsby et al., "Parameter-efficient transfer learning for NLP," in *Proc. 36th Int. Conf. Mach. Learn.*, Jan. 2019, pp. 2790–2799.

[28] F. Lu, L. Zhang, X. Lan, S. Dong, Y. Wang, and C. Yuan, "Towards seamless adaptation of pre-trained models for visual place recognition," in *Proc. 12th Int. Conf. Learn. Represent.*, 2024, pp. 1–20.

[29] B. Li, L. Y. Wu, D. Liu, H. Chen, Y. Ye, and X. Xie, "Image template matching via dense and consistent contrastive learning," in *Proc. IEEE Int. Conf. Multimedia Expo. (ICME)*, Jul. 2023, pp. 1319–1324, doi: 10.1109/ICME55011.2023.00229.

[30] T.-Y. Lin, S. Belongie, and J. Hays, "Cross-view image geolocalization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 891–898, doi: 10.1109/CVPR.2013.120.

[31] M. Bansal, H. S. Sawhney, H. Cheng, and K. Daniilidis, "Geo-localization of street views with aerial image databases," in *Proc. 19th ACM Int. Conf. Multimedia*, Nov. 2011, pp. 1125–1128, doi: 10.1145/2072298.2071954.

[32] M. Zhai, Z. Bessinger, S. Workman, and N. Jacobs, "Predicting ground-level scene layout from aerial imagery," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4132–4140, doi: 10.1109/CVPR.2017.440.

[33] Y. Shi, L. Liu, X. Yu, and H. Li, *Spatial-Aware Feature Aggregation for Cross-View Image Based Geo-Localization*. Red Hook, NY, USA: Curran Associates Inc., 2019, doi: 10.5555/3454287.3455192.

[34] X. Zhang, X. Li, W. Sultani, C. Chen, and S. Wshah, "GeoDTR+: Toward generic cross-view geolocalization via geometric disentanglement," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 12, pp. 10419–10433, Dec. 2024, doi: 10.1109/TPAMI.2024.3443652.

[35] X. Zhao, L. Cui, X. Wei, C. Liu, and J. Yin, "Lunar rover cross-view localization through integration of rover and orbital images," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5642414, doi: 10.1109/TGRS.2024.3462487.

[36] N. N. Vo and J. Hays, "Localizing and orienting street views using overhead imagery," in *Computer Vision—ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., Cham, Switzerland: Springer, 2016, pp. 494–509.

[37] S. Workman, R. Souvenir, and N. Jacobs, "Wide-area image geolocalization with aerial reference imagery," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3961–3969.

[38] L. Liu and H. Li, "Lending orientation to neural networks for cross-view geo-localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5617–5626, doi: 10.1109/CVPR.2019.00577.

[39] S. Zhu, T. Yang, and C. Chen, "VIGOR: Cross-view image geo-localization beyond one-to-one retrieval," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 3640–3649, doi: 10.1109/CVPR46437.2021.00364.

[40] R. Zhu, L. Yin, M. Yang, F. Wu, Y. Yang, and W. Hu, "SUES-200: A multi-height multi-scene cross-view image benchmark across drone and satellite," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 9, pp. 4825–4839, Sep. 2023, doi: 10.1109/TCSVT.2023.3249204.

[41] T. Shen, Y. Wei, L. Kang, S. Wan, and Y.-H. Yang, "MCCG: A ConvNeXt-based multiple-classifier method for cross-view geo-localization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 3, pp. 1456–1468, Mar. 2024, doi: 10.1109/TCSVT.2023.3296074.

[42] F. Deuser, K. Habel, and N. Oswald, "Sample4Geo: Hard negative sampling for cross-view geo-localisation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 16801–16810, doi: 10.1109/iccv51070.2023.01545.

[43] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11966–11976, doi: 10.1109/CVPR52688.2022.01167.

[44] P. Xia, Y. Wan, Z. Zheng, Y. Zhang, and J. Deng, "Enhancing cross-view geo-localization with domain alignment and scene consistency," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 12, pp. 13271–13281, Dec. 2024, doi: 10.1109/TCSVT.2024.3443510.

[45] H. Zhao, K. Ren, T. Yue, C. Zhang, and S. Yuan, "TransFG: A cross-view geo-localization of satellite and UAVs imagery pipeline using transformer-based feature aggregation and gradient guidance," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 4700912, doi: 10.1109/TGRS.2024.3352418.

[46] H. Lv et al., "Direction-guided multiscale feature fusion network for geo-localization," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5622813, doi: 10.1109/TGRS.2024.3396912.

[47] Z. Liu et al., "Swin transformer v2: Scaling up capacity and resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11999–12009, doi: 10.1109/CVPR52688.2022.01170.

[48] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 248–255, doi: 10.1109/CVPR.2009.5206848.

[49] J. E. Hu et al., “LoRA: Low-rank adaptation of large language models,” in *Proc. Int. Conf. Learn. Represent.*, Jan. 2021, pp. 1–26.

[50] B. Lester, R. Al-Rfou, and N. Constant, “The power of scale for parameter-efficient prompt tuning,” in *Proc. Conf. Empirical Methods Natural Lang. Process.* Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021, pp. 3045–3059, doi: 10.18653/v1/2021.emnlp-main.243.

[51] S. Chen et al., “AdaptFormer: Adapting vision transformers for scalable visual recognition,” in *Proc. NIPS*, 2022, pp. 16664–16678.

[52] M. Jia et al., “Visual prompt tuning,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Jan. 2022, pp. 709–727, doi: 10.1007/978-3-031-19827-4_41.

[53] T. Chen et al., “SAM-adapter: Adapting segment anything in underperformed scenes,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2023, pp. 3359–3367, doi: 10.1109/iccvw60793.2023.00361.

[54] Y. Qiao, Z. Yu, and Q. Wu, “VLN-PETL: Parameter-efficient transfer learning for vision-and-language navigation,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 15397–15406, doi: 10.1109/iccv51070.2023.01416.

[55] F. Lu, X. Lan, L. Zhang, D. Jiang, Y. Wang, and C. Yuan, “CricaVPR: Cross-image correlation-aware representation learning for visual place recognition,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 16772–16782, doi: 10.1109/cvpr52733.2024.01587.

[56] R. Hadsell, S. Chopra, and Y. LeCun, “Dimensionality reduction by learning an invariant mapping,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2006, pp. 1735–1742, doi: 10.1109/CVPR.2006.100.

[57] E. Hoffer and N. Ailon, “Deep metric learning using triplet network,” in *Proc. Int. Workshop Similarity-Based Pattern Recognit. (SIMBAD)*. Cham, Switzerland: Springer, 2015, pp. 84–92, doi: 10.1007/978-3-319-24261-3_7.

[58] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” 2020, *arXiv:2005.12872*.

[59] A. Ali-Bey, B. Chaib-Draa, and P. Giguère, “BoQ: A place is worth a bag of learnable queries,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 17794–17803, doi: 10.1109/cvpr52733.2024.01685.

[60] A. Vaswani et al., “Attention is all you need,” in *Proc. NIPS*, 2017, pp. 6000–6010.

[61] S. Chen, E. Xie, C. Ge, R. Chen, D. Liang, and P. Luo, “CycleMLP: A MLP-like architecture for dense prediction,” in *Proc. Int. Conf. Learn. Represent.*, Jan. 2021, pp. 1–21, doi: 10.1109/TPAMI.2023.3303397.

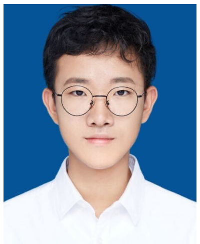
[62] F. Radenovic, G. Tolias, and O. Chum, “Fine-tuning CNN image retrieval with no human annotation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1655–1668, Jul. 2019, doi: 10.1109/TPAMI.2018.2846566.

[63] M. Chu, Z. Zheng, W. Ji, T. Wang, and T. Chua, “Towards natural language-guided drones: GeoText-1652 benchmark with spatial relation matching,” in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2024, pp. 213–231, doi: 10.1007/978-3-031-73247-8_13.

[64] H. Ju, S. Huang, S. Liu, and Z. Zheng, “Video2BEV: Transforming drone videos to BEVs for video-based geo-localization,” 2024, *arXiv:2411.13610*.

[65] G. Berton, L. Junglas, R. Zaccone, T. Pollok, B. Caputo, and C. Masone, “MeshVPR: Citywide visual place recognition using 3D meshes,” 2024, *arXiv:2406.02776*.

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



Shuyu Hu received the B.S. degree from China University of Petroleum, Beijing, China, in 2023. He is currently pursuing the M.S. degree with Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China.

His research interests include image-based geo-localization, cross-spectral geo-localization, and drone navigation.

Zelin Shi received the B.S. degree in mathematics and the M.S. degree in computer application from Xidian University, Xi’an, China, in 1987 and 1990, respectively, and the Ph.D. degree in mechanical and electronic engineering from the Graduate School, Chinese Academy of Sciences, Beijing, China, in 2004.

He is currently a Professor with Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China. His research interests include optoelectronic imaging, image processing, and pattern recognition.

Tong Jin received the B.S. degree in automation from Beihang University, Beijing, China, in 2023. He is currently pursuing the M.S. degree with Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China.

His research interests include computer vision, visual place recognition, and image-based geo-localization.

Yunpeng Liu (Member, IEEE) received the Ph.D. degree in pattern recognition and machine intelligence from Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China, in 2010.

He is currently a Professor with Shenyang Institute of Automation, Chinese Academy of Sciences. His research interests include cross-spectral image patch matching, image segmentation, infrared small target detection, small target tracking, and recognition based on the Riemannian manifold.