

Drone Referring Localization: An Efficient Heterogeneous Spatial Feature Interaction Method For UAV Self-Localization

Ming Dai, Enhui Zheng, Jiahao Chen, Lei Qi, Zhenhua Feng, *Senior Member, IEEE* and Wankou Yang, *Member, IEEE*

Abstract—Image retrieval (IR) has emerged as a promising approach for self-localization in unmanned aerial vehicles (UAVs). However, IR-based methods face several challenges: 1) Pre- and post-processing incur significant computational and storage overhead; 2) The lack of interaction between dual-source features impairs precise spatial perception. In this paper, we propose an efficient heterogeneous spatial feature interaction method, termed Drone Referring Localization (DRL), which aims to localize UAV-view images within satellite imagery. Unlike conventional methods that treat different data sources in isolation, followed by cosine similarity computations, DRL facilitates the learnable interaction of heterogeneous features. To implement the proposed DRL, we design two transformer-based frameworks, Post-Fusion and Mix-Fusion, enabling end-to-end training and inference. Furthermore, we introduce random scale cropping and weight balance loss techniques to augment paired data and optimize the balance between positive and negative sample weights. Additionally, we construct a new dataset, UL14, and establish a benchmark tailored to the DRL framework. Compared to traditional IR methods, DRL achieves superior localization accuracy (MA@20 +9.4%) while significantly reducing computational time (1/7) and storage overhead (1/3). The dataset and code will be made publicly available. The dataset and code are available at <https://github.com/Dmmm1997/DRL>.

Index Terms—unmanned aerial vehicle, geo-localization, benchmark, transformer.

I. INTRODUCTION

IN recent years, UAVs have seen extensive application in vision-related tasks across various domains [1], [2], [3], such as facility inspection, agricultural operations, ground reconnaissance, and civilian aerial photography. UAVs typically rely on global positioning system (GPS) signals for self-localization. However, the stability of these signals varies across different environments, and UAVs often operate in scenarios where GPS signals are weak or completely unavailable. In such situations, UAVs may experience operational failures or even complete loss of control. This paper explores an alternative self-localization approach for UAVs that relies

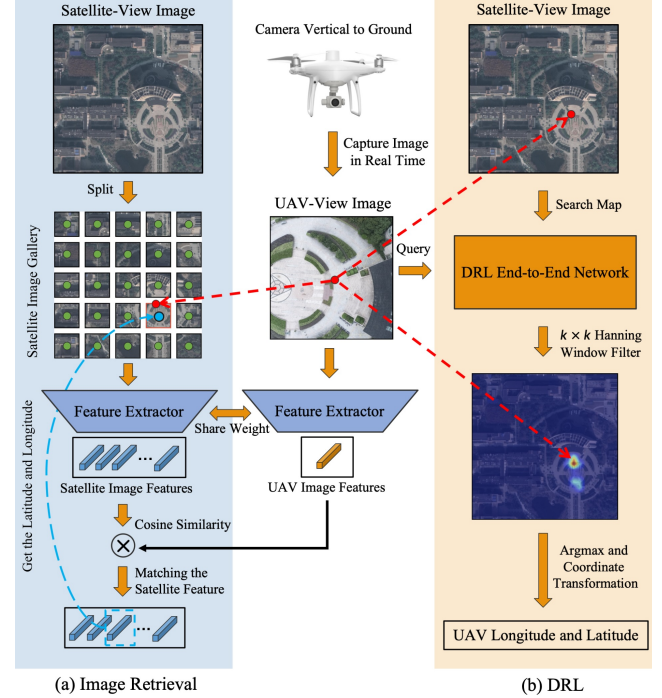


Fig. 1. A comparison between (a) image retrieval and (b) the proposed DRL framework for UAV self-localization. In IR-based methods, the features of different satellite images are spatially isolated and do not interact with UAV image features. Additionally, this approach requires complex pre- and post-processing operations. The green dot represents the sampling position in the gallery, so the IR-based method inevitably introduces inherent errors caused by sampling. In contrast, the proposed DRL method adopts an end-to-end heterogeneous spatial feature fusion architecture, which simplifies the entire localization process and overcomes the inherent errors.

solely on visual information, providing a viable solution in GPS-denied environments.

For conventional positioning tasks, feature point matching methods such as scale-invariant feature transform (SIFT) [4] and speeded-up robust features (SURF) [5] are commonly used. These methods rely on hand-crafted feature descriptors that exhibit advantageous properties, including scale, rotation, and invariance. They have been widely and successfully applied in tasks like simultaneous localization and mapping (SLAM) [6] and image stitching [7]. However, in scenarios where image quality varies significantly across domains, corner-point matching methods can only extract a limited number of distinctive features, leading to a substantial decline in performance. To address these limitations, deep learning has

M. Dai, W. Yang are with the School of Automation, Southeast University, Nanjing 210096, China. E-mails: (mingdai, wkyang)@seu.edu.cn.

E. Zheng, J. Chen are with the Unmanned System Application Technology Research Institute, China Jiliang University, Hangzhou, 310018, China. Email: (ehzheng, P21010854010)@cjlu.edu.cn.

Z. Feng is with the School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi 214122, China. E-mail: fengzhenhua@jiangnan.edu.cn

L. Qi is with the School of Computer Science, Southeast University, Nanjing 210096, China E-mails: qilei@seu.edu.cn

emerged as a powerful alternative, capable of capturing richer feature relationships and delivering superior performance in challenging tasks.

The UAV self-localization task presents several challenges, including drastic domain differences, uncertain scale, inconsistent viewing angles, and misaligned spatial information due to time offsets. Thus, some deep learning methods have been developed to improve content understanding and feature integration. One such method is DenseUAV [8], which adopted an image retrieval scheme and used the UAV-view image as a query to retrieve the most similar satellite-view image from a satellite gallery. Subsequently, GPS information is obtained to determine the UAV's position. At present, fine-grained positioning methods have emerged for matching ground panoramas with satellite images [9], [10], [11], [12], with VIGOR-type [13] methods being prominent examples. These methods improve positioning accuracy by adding offset prediction to image retrieval. This paper focuses on the UAV self-localization task, aiming to achieve end-to-end UAV positioning through an efficient method distinct from image retrieval. In this context, the process of searching for a rough position can be bypassed. Due to the spatial continuity of object movement, once the initial position is determined, a corresponding rough satellite image can be cropped. If the UAV's location can be identified within this rough satellite image, its trajectory can be continuously tracked as it moves by iteratively applying this process.

Although IR-based methods significantly improve positioning accuracy, they have limitations in two key aspects. (1) These methods do not integrate features from different sources, limiting their ability to evaluate image similarity based solely on feature distance from respective sources. As a result, IR-based methods are unable to learn spatial relative relationships and offer an isolated solution for UAV self-localization. Furthermore, because the gallery in IR-based methods is sampled at fixed spatial intervals, data-level errors are inevitable. As illustrated by the red dotted line in Fig. 1, the green points represent the positions of the cropped satellite images. If the sampling interval is large, meaning the green points are sparse, positioning errors will increase. However, denser sampling leads to higher computational and storage costs. (2) IR-based methods cannot achieve end-to-end processing during application, as they involve multiple steps with high time and space complexity, as shown in Fig. 1 (a). First, in the pre-processing step, large-scale satellite images need to be cropped and a feature gallery needs to be constructed through model inference. The calculation and storage consumption of this part depend on the sampling interval and the size of the real space area to be retrieved. Then, in the post-processing part, the UAV image features acquired in actual scenes need to be compared against the satellite gallery by calculating the cosine distance between features. The computational burden here is directly related to the size of the gallery. Although optimization methods like KNN [14] can reduce computational load, they also introduce a certain loss of accuracy, which is suboptimal for tasks requiring high precision.

Given the limitations discussed above, this paper proposes an end-to-end heterogeneous spatial feature interaction method

for UAV self-localization, termed Drone Referring Localization (DRL). For clarity, we refer to the UAV image as the *query* and the satellite image as the *search map*. The process is illustrated in Fig. 1 (b). DRL simultaneously inputs both drone and satellite images into the model and outputs a heatmap whose spatial distribution aligns with the satellite image. Each pixel in the heatmap represents the probability of the drone being at the corresponding location. Unlike IR-based methods, DRL allows for full interaction between heterogeneous source features and avoids errors related to sampling intervals, thereby enhancing positioning accuracy. Additionally, DRL streamlines the entire localization process by eliminating the need for complex pre- and post-processing operations, reducing computational and storage overhead, and improving inference speed.

The primary focus of this paper is to investigate an end-to-end heterogeneous feature interaction method for UAV self-localization. Additionally, we conduct experiments from both data and model perspectives to validate the proposed approach. To support the training requirements of the DRL architecture, we reconstruct the pairwise attribute dataset, UL14, derived from the DenseUAV classification dataset, with a more challenging test set configuration. Furthermore, we propose two model architectures, Post-Fusion and Mix-Fusion, which integrate heterogeneous features at the end and intermediate stages of the backbone network, respectively. To comprehensively evaluate performance, we design two additional metrics: Meter-level Accuracy (MA), which intuitively measures spatial distance, and Relative Distance Score (RDS), which assesses positioning performance at the model level. Furthermore, we apply data augmentation techniques, including random scaling and random offset, to diversify the training sample pairs. Additionally, a negative sample weight factor is introduced within the balanced loss framework, significantly improving optimization during training.

The main contributions can be listed as follows:

- 1) We propose an efficient heterogeneous feature interaction method for UAV self-localization task called Drone Referring Localization (DRL), which deeply interacts with heterogeneous features in a learnable manner and circumvents complex pre- and post-processing steps.
- 2) We present two model architectures, Post-Fusion and Mix-Fusion, and conclude that the Mix-Fusion is more effective than the Post-Fusion. Moreover, we introduce a data augmentation technique involving random scale and offset to enrich the diversity of training data. Additionally, we improve balance loss and analyze the impact of the number and weight of positive and negative samples.
- 3) We constructed a new benchmark, which includes a new dataset UL14 constructed from paired samples, and evaluation indicators Meter-level Accuracy (MA) and Relative Distance Score (RDS).
- 4) The proposed DRL has higher positioning accuracy (MA@20 +9.4%) while significantly reducing time (1/7) and storage (1/3) overhead than the IR-based solution.

The remainder of this paper is structured as follows. First, we introduce the related work in Section II. Then, the model archi-

texture of DRL is designed in Section III, and the proposed dataset and evaluation metrics are presented in Section IV. Next, we analyze the experiments from data- and model-level in Section V and Section VI. And the visualization and limitations are drawn in Section VII and Section VIII. Last, we conclude the context of this paper in Section IX

II. RELATED WORK

A. Geo-Localization Dataset

1) *Ground-to-Aerial Matching*: Geo-localization originally addressed the task of matching ground and aerial imagery. Some seminal investigations conducted by [15], [16], [17] laid the foundation for leveraging publicly available resources to establish image pairs encompassing ground and aerial views. Subsequently, the CVUSA [18] project devised image pairings derived from ground-based panoramic images and satellite imagery, whereas CVACT [19] further enhanced CVUSA by incorporating spatial elements, such as orientation maps. Recently, VIGOR [13] redefined the problem by adopting a more realistic assumption: that the query image can be of arbitrary nature within the specified area of interest.

2) *Drone-to-Satellite Matching*: Univeristy-1652 [1] introduced drone-view into cross-view geo-localization and proposed two drone-based subtasks: *drone-view target localization* and *drone navigation*, and regarded them as image retrieval tasks. SUE-200 [20] improved the model's adaptability to drone flight altitudes by collecting drone images at 4 different altitudes. Inspired by Univeristy-1652, DenseUAV [8] achieved high precision localization of UAVs by dense sampling, which is the first time to solve the vision-based UAV self-localization problem by the scheme of image retrieval.

B. Deeply-Learned Geo-Localization

Due to the potential applications, Cross-View Geo-Localization (CVGL) has gotten increased attention in recent years. Some pioneering methods [21], [22], [23], [24], concentrated on extracting hand-crafted features. Inspired by the powerful feature extraction capability of deep Convolutional Neural Networks (CNNs), most of the later works are carried out on the basis of deep learning [25], [26], [27], [28], [29]. Since drone-view was introduced for the CVGL task, several methods have been proposed specifically for addressing the challenges in drone-view [30], [31]. DSM [32] took into account a limited field of view and employed a dynamic similarity matching module to align the orientation of cross-view images. PLCD [33] took advantage of drone-view information as a bridge between ground-view and satellite-view domains. LPN [34] proposed the square-ring partition strategy to allow the network to pay attention to more fine-grained information at the edge and achieved a huge improvement. FSRA [35] introduced a simple and efficient transformer-based structure to enhance the ability of the model to understand contextual information as well as to understand the distribution of instances. MCCG [36] introduced a ConvNeXt-based method to captures rich discriminative information by cross-dimension interaction and acquires multiple feature representations.

C. Transformer in Vision

In recent years, Transformer [37] has gradually become the mainstream basic model architecture in the fields of natural language processing (NLP) [38], [39] and computer vision (CV) [40], [41]. Some CVGL methods have also been further improved with the help of Transformer. Currently, the backbone based on the vision Transformer can be mainly divided into two categories. One is the unstructured Transformer structure, the most typical models such as ViT [40], and DeiT [41]. The other is the hierarchical Transformer structure, which typically includes PvT [42], CvT [43], and Swin-Transformer [44], which are widely used in fine-grained tasks such as object detection [45], [46] and image segmentation [47], [48].

Considering the requirements of this task for spatial context understanding and fine-grained mining capabilities. Through the experience of previous work [8] and the verification of our experiments, Transformer can perform better. More details will be analyzed in Section VI-D.

III. METHODOLOGY

A. Localization Schemes

The proposed framework, Drone Referring Localization (DRL), introduces a novel approach to UAV self-localization. To the best of our knowledge, it is the first architecture to employ an end-to-end solution for this task. To clearly convey the innovative aspects of DRL, we provide a brief overview of the IR-based scheme in Section III-A1, followed by an in-depth exploration of the unique features of DRL in Section III-A2.

1) *Image Retrieval*: This approach estimates image similarity by comparing features across domains, then indirectly derives GPS information for localization. The detailed process is illustrated in Fig. 1(a). However, IR-based methods have several inherent limitations: (1) *Isolated heterogeneous features*: UAV and satellite images lack interaction. Similarity is only assessed by the distance between isolated final features, without integrating information from different sources. (2) *Inherent positioning error*: As shown by the green sampling points in Fig. 1(a), the sampling interval introduces inevitable error, even for the optimal retrieval result, which deviates from the ground truth. (3) *Complicated pre-processing*: The interval for satellite image cropping directly affects positioning accuracy and speed; finer cropping improves accuracy but reduces inference speed. (4) *Intricate post-processing*: Calculating feature similarity incurs substantial computational costs, which increase linearly with the size of the satellite gallery. (5) *Inflexible model iterability*: When the model is updated, the feature gallery must be regenerated. These limitations hinder IR-based methods from achieving real-time inference performance.

2) *Drone Referring Localization (DRL)*: The feature similarity calculation process in image retrieval can be analogized to convolution operations, where feature correlation is computed using a sliding window. Inspired by this approach and drawing on techniques from the SOT field, which typically handles inputs from the same domain, we propose a novel localization framework named Drone Referring Localization

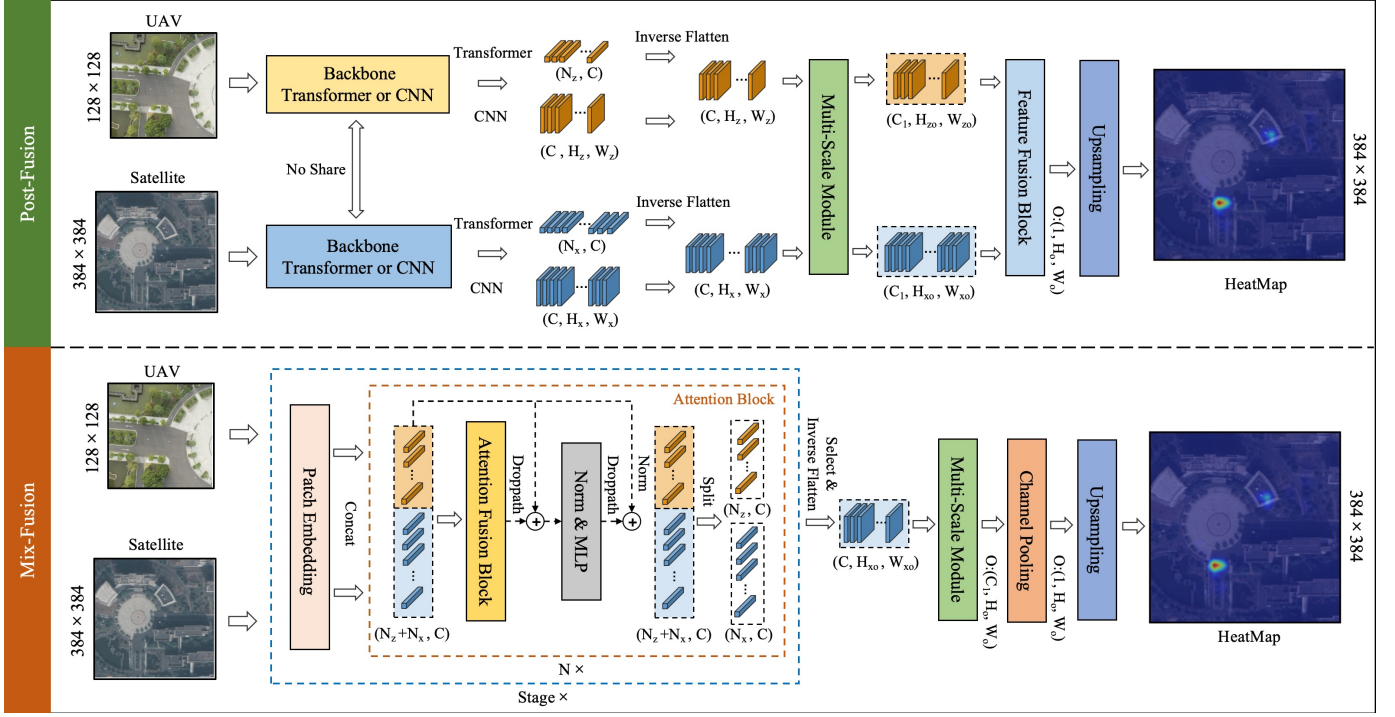


Fig. 2. Post-Fusion is a dual-stream architecture that interacts with UAV- and satellite-view features through a Feature Fusion Block. Mix-Fusion is a single-stream network that interacts with feature information in the feature extraction part.

(DRL) to address the UAV self-localization task. The DRL architecture not only facilitates effective interaction between heterogeneous features, resulting in significantly improved positioning accuracy, but also distinguishes itself from IR-based methods by eliminating the need for complex pre- and post-processing steps. This allows UAV self-localization to be accomplished through end-to-end inference. The steps for achieving localization with the DRL framework are illustrated in Fig. 1(b). DRL performs feature interaction, fuses information from the two sources, and ultimately outputs a heatmap representing the spatial probability distribution. Following simple post-processing steps, such as applying a Hanning window filter and coordinate transformation, the UAV's position can be determined. DRL not only simplifies the UAV self-localization process but also significantly improves inference speed and reduces storage resource consumption, which will be further analyzed in Section V-B.

B. Model Structure of DRL

This section offers an elaborate explanation of two model structures designed for DRL architecture. The first structure employs a dual-stream network, named *Post-Fusion* (refer to Section III-B1). The second structure utilizes a single-stream network, named *Mix-Fusion* (refer to Section III-B2).

1) *Post-Fusion*: As illustrated in Fig. 2, the Post-Fusion entails a dual-stream structure, where images from two sources undergo feature extraction and subsequent fusion using feature fusion blocks (FFB). The feature extraction phase commonly employs backbones such as CNNs or Transformers, which will be discussed in Section VI-D.

This paper introduces three feature fusion approaches, namely group convolution (GC), cross-attention fusion (CAF),

and self-attention fusion (SAF). Their structures are shown in Fig. 3(c). The experimental analysis of FFB will be discussed in Section VI-A.

Group Convolution (GC): The features of the UAV-view are treated as the kernels, while the features of the satellite-view serve as the sources. Interaction between the features of the two domains is achieved through group convolution [49].

Cross-Attention Fusion (CAF): The spatial dimensions of UAV and satellite features are flattened to $N_z = H_{z0} \times W_{z0}$ and $N_x = H_{x0} \times W_{x0}$ respectively. The features from UAV-view $F_z \in \mathbb{R}^{(C, N_z)}$ serve as key and value, while the features from satellite-view $F_x \in \mathbb{R}^{(C, N_x)}$ serve as query. The multi-head self-attention (MSA) mechanism is employed to fuse the features, producing the corresponding features $F_o \in \mathbb{R}^{(C, N_x)}$ of the same size as query. Then, dropout and layernorm operations are applied to F_o and added to F_x to resemble a residual structure. The FFN module is used for feature integration, involving linear, activation, normalization, and dropout operations. Last, the C-dimension is compressed through average pooling and restored to the width and height shape $(1, H_o, W_o)$ via an inverse flatten operation.

Self-Attention Fusion (SAF): SAF constructs query, key, and value by concatenating the features from the UAV- and satellite-view. Specifically, the input dimension of the multi-head attention mechanism is $(C, N_z + N_x)$. Last, the part originally belonging to the satellite feature is divided from the spliced features through the split operation.

2) *Mix-Fusion*: Mix-Fusion is essentially a single-stream network structure that allows image features from two domains to fuse within the transformer backbone, without additional parameters. Therefore, the pre-trained weights can still be loaded. The structure is shown in the lower half of Fig. 2. First,

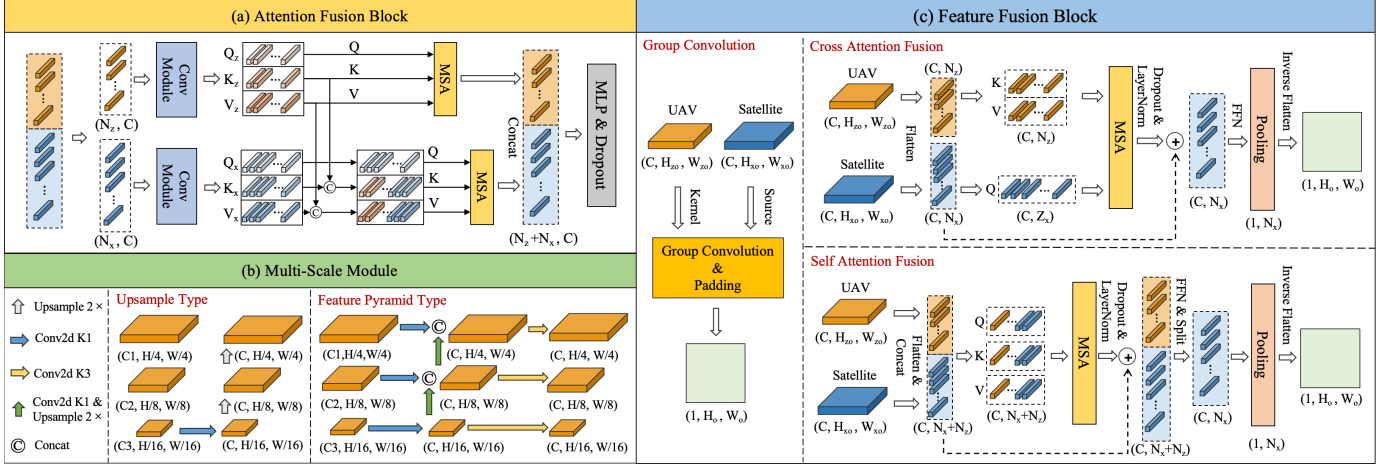


Fig. 3. This diagram illustrates some detailed modules in the DRL architecture (Fig. 2). (a) is the Attention Fusion Block part of the Mix-Fusion structure. The Multi-Scale Module represented by (b) can be applied to both architectures, and two types of structures are used here to implement it. (c) is the Feature Fusion Block part of the Post-Fusion structure, which contains 3 different types to fuse heterogeneous features.

images from two sources are simultaneously fed to the model, and compressed by patch embedding to $1/4$ of the original size. Next, flatten the spatial dimension as $F_z \in \mathbb{R}^{(N_z, C)}$ and $F_x \in \mathbb{R}^{(N_x, C)}$. Then, concatenate F_x and F_z to obtain $F_m \in \mathbb{R}^{(N_x+N_z, C)}$ and use the attention fusion block (AFB) for feature fusion. Here, a residual structure is added to protect the original features and prevent the attention mechanism from degrading the results. Then, normalize the features through layernorm and enrich feature diversity through MLP. The original input, after normalization, is added to the output to obtain the new fused features F_m . Last, complete one stage by splitting F_m into F_z and F_x . Excluding the patch embedding operation, the remaining part constitutes a complete attention block process.

Next, we introduce the structure of the AFB, as depicted in Fig. 3(a). First, the input features are divided into F_z and F_x , which are then mapped to query, key, and value using a convolutional (conv) module. Specifically, the conv module performs inverse flatten, convolution, and flatten operations. The attention mechanism in the AFB consists of two parts: (1) a self-attention mechanism for UAV features, and (2) a cross-attention mechanism where the satellite features provide the query, while the concatenated features from both domains serve as the key and value. Finally, the outputs representing the UAV and satellite features are concatenated, followed by MLP and dropout operations.

After the feature extraction process, the satellite-view features are restored to their original dimensions (C, H_{so}, W_{so}) using an inverse flatten operation. A multi-scale module (MSM) is optionally applied to capture both shallow and deep features. Additionally, channel pooling is utilized to compress the channel dimension, and upsampling is employed to restore the spatial scale.

C. Feature Scales

1) *Multi-Scale Modules (MSM)*: We adopt two resolution expansion strategies to evaluate the impact of multi-scale processing on UAV self-localization tasks. First, the upsampling strategy, as depicted in Fig. 3(b), restores the scale

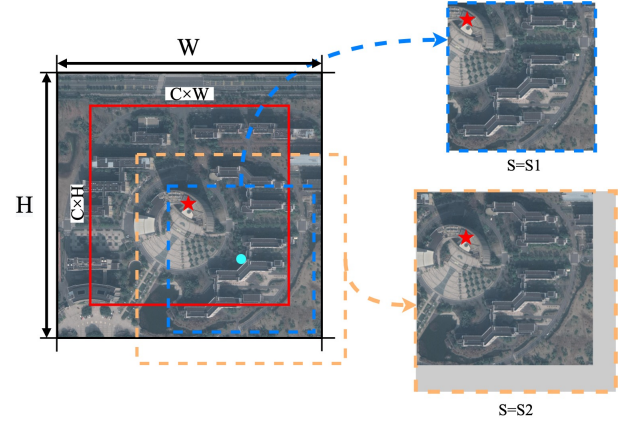


Fig. 4. The process of the RSC augmentation method. The red pentagon is the position of the UAV. The light blue dot is the randomly generated center position of the image. The orange and blue dotted boxes correspond to 2 cropped images of different scales.

by applying bilinear interpolation to the final layer feature, which has dimensions $(C3, H/16, W/16)$. Second, the feature pyramid approach [50], also shown in Fig. 3(b), leverages the original structure of the feature pyramid, merging features from different layers through a combination of convolution, upsampling, and concatenation operations. The experimental results of these strategies will be discussed in Section VI-C1.

2) *Output Resolution*: For the UAV self-localization task, the resolution of the output feature map significantly impacts the localization accuracy. A higher resolution leads to a smaller localization error. In fact, experimental results demonstrate that adding an upsampling module at the end of the model effectively improves positioning performance. Further analysis is provided in Section VI-C2.

D. Random Scale Crop (RSC)

There are two primary data construction schemes for training DRL: pre-building fixed UAV-satellite pairs and generating pairs randomly through data augmentation. We adopt the latter approach for greater flexibility and introduce a random scale

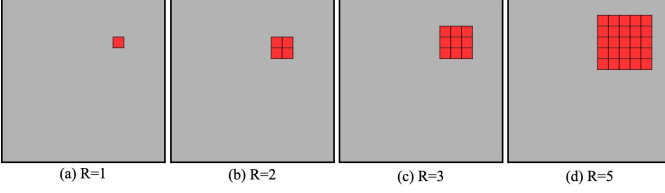


Fig. 5. Schematic diagram of selecting positive samples according to R .

and offset augmentation method, termed Random Scale Crop (RSC).

The RSC method generates sample pairs with both random target distributions and random scales. Figure 4 illustrates the implementation of RSC, which uses two hyperparameters to dynamically crop satellite-view images during training: the centroid coverage range C (default 0.85) and the scale range S (default 512→1000). The hyperparameter C controls the distribution interval of the query within the search map. A smaller C results in queries being concentrated closer to the center of the search map. The hyperparameter S determines the spatial scale of the search map, with a larger S increasing the spatial distance covered by each pixel after resizing. The RSC method aims to enhance the model's robustness to scale and offset variations. In simpler terms, C defines the size of the red area where queries can be distributed, while different values of S (e.g., S_1 and S_2) are used to generate search maps at varying scales.

Algorithm 1 Weighted Balance Loss.

Input: $map \in \mathbb{R}^{H \times W}$, $label \in \mathbb{R}^{H \times W}$, $w_{neg} \in \mathbb{R}$, $R \in \mathbb{N}^+$

Output: loss

```

1: # Generate the 0,1 matrix as shown in Fig. 5
2:  $(map, label, R) \Rightarrow t$ 
3: # Num of the positive and negative samples
4:  $w \leftarrow t$ 
5:  $N_{pos} = R^2$ 
6:  $N_{neg} = H \times W - R^2$ 
7: # Weight of the positive and negative samples
8:  $W_{pos} = 1/N_{pos}$ 
9:  $W_{neg} = (1/N_{neg}) \times w_{neg}$ 
10: # Weight normalization
11:  $w \leftarrow w / \sum w_i$ 
12: # Map normalization
13:  $p \leftarrow \text{sigmoid}(map)$ 
14: # Weighted balance loss
15:  $loss = -\sum (\log(p_{ij})t_{ij}w_{ij} + \log(1 - p_{ij})(1 - t_{ij})w_{ij})$ 
16: return loss

```

E. Weighted Balance Loss (WBL)

The number of positive and negative samples, as well as their associated weights, are crucial variables in model optimization. An imbalance in these samples can lead to the model converging towards an extreme mode that minimizes the loss function, which may result in misleading the model or slowing down convergence. balance loss [51] is an effective method for balancing positive and negative samples, with its core principle being to maintain a 1:1 weight ratio between them. However, balance loss does not sufficiently address the issue of sample imbalance, and a strict 1:1 balance is not optimal for localization tasks. To address these issues, we

propose an improved method called weighted balance loss (WBL), which is detailed in Alg. 1. WBL classifies samples within a specified range R around the ground truth as positive samples and introduces a weight factor, w_{neg} , to adjust the influence of negative samples.

Specifically, the parameter R is chosen based on the number of positive samples. As illustrated in Fig. 5, $R=1$ implies that only the position closest to the ground truth is considered positive, with all other positions classified as negative. When $R=2$, the four neighboring points around the ground truth are also considered positive samples. As R increases, the number of positive samples increases accordingly, but the weight assigned to each positive sample decreases proportionally.

In terms of sample weights, firstly, the number of positive and negative samples should be counted. A weight matrix whose size is the same as the heatmap needs to be prepared to store the weight of each sample. Since the number of negative samples is much larger than that of positive, the weight of each positive is much larger than that of the negative after balancing. Therefore, w_{neg} is introduced to dynamically adjust the weights for negative samples. Last, the weight matrix is multiplied by the cross-entropy loss value for all samples.

IV. DATASET AND EVALUATION

A. Dataset Description

Table I lists the amount of training data, sampling platforms, data distribution, localization targets, and evaluation metrics for some typical geo-localization datasets. Explicitly, the UL14 dataset possesses the following characteristics. (1) Image from two perspectives: UAV and satellite views. (2) Paired training data: the UAV images and satellite images in the training data are constructed in pairs, and the data augmentation method will be used in the actual training process to expand the diversity of paired data. (3) Multiple flight altitudes: UAV images are sampled in three flight altitudes (80, 90, and 100 meters). (4) Multi-scale test data: The test satellite data is constructed of 12 different scales to enhance the difficulty of positioning.

Next, we will provide a detailed introduction to the construction of the train and test set, and the quantity statistics are presented in Table II.

1) *Train Set*: The training set comprises images collected by 10 universities. To enhance the utility of the training set, the resolution of UAV-view images is uniformly set to (512, 512), while the satellite images are set to (1280, 1280). The decision to retain larger satellite images is made to allow for greater flexibility in subsequent data augmentation, which is be discussed in Section III-D. Last, some paired images of the train set are shown in Fig. 6(a).

2) *Test Set*: The test set comprises images collected by 4 universities, with no overlap with the train set. The spatial scale of the satellite images is defined by setting the pixels in the range of 700-1800 (0.294 meter/pixel) at intervals of 100 pixels. This implies that each UAV image in the test set generated 12 satellite images of varying scales. Also, the position of the UAV in the satellite imagery is randomly distributed, either in the center or at the edge. For a clear view, one set of images in the test set is shown in Fig. 6(b).

TABLE I
GEO-LOCALIZATION DATASET SUMMARY. THE "TRAINING" COLUMN INDICATES THE NUMBER OF SAMPLES IN EACH LOCATION. THE UL14 TRAINING SET IS CONSTRUCTED USING 1-TO-1 PAIRED DATA.

Datasets	UL14 (ours)	DenseUAV [8]	SUES-200 [20]	University-1652 [1]	VIGOR [13]	CVUSA [18]
Training	$6.8k \times 2$	$10 \times 225.6 \times 9$	120×51	701×71.64	$91k + 53k$	$35.5k \times 2$
Platform	Drone, Satellite	Drone, Satellite	Drone, Satellite	Drone, Ground, Satellite	Ground, Satellite	Ground, Satellite
Imgs./Platform	1 + 1	3 + 6	50 + 1	54 + 16.64 + 1	/	1 + 1
Target	UAV	UAV	Diverse	Building	User	User
Method	DRL	IR	IR	IR	IR + Reg	IR
Evaluation	RDS & MA	R@K & SDM	R@K & AP & RB & PF	R@K & AP	MA	R@K

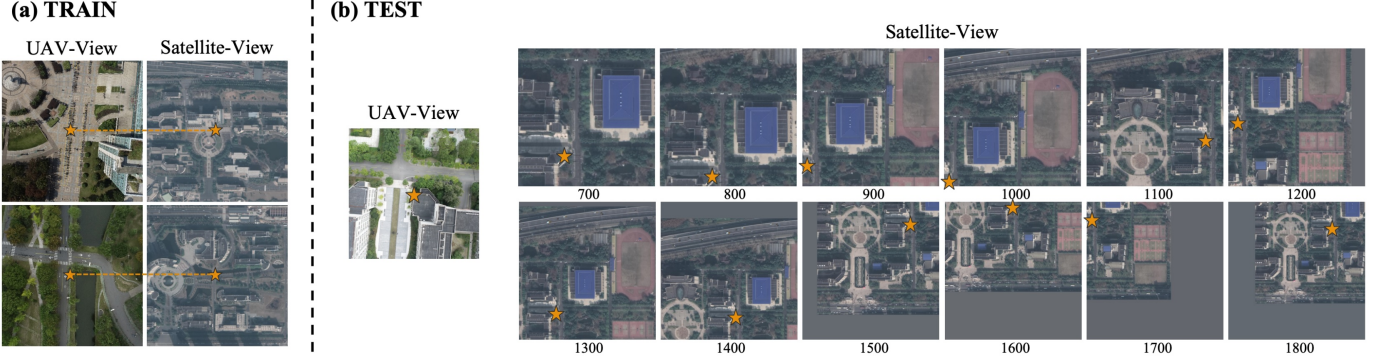


Fig. 6. Visualization of some samples in the train and test set. The numbers in the test set represent the original pixel scale (0.294 meters/pixel).

TABLE II
STATISTICAL TABLE FOR THE UL14 DATASET, WHERE #UNIVERSITIES REFERS TO THE NUMBER OF UNIVERSITIES INCLUDED.

#Subset	#Imgs		#Universities
	UAV-view	Satellite-view	
Train	6768	6768	10
Test	2331	27972	4

B. Evaluation Indicators

Recall@K and AP[52], [53] are commonly used evaluation metrics, which only consider whether the sample is positive or negative. This is a kind of discrete metric, which defeats the real purpose of the localization task. Unlike the indirect positioning of the image retrieval scheme, DRL needs a more intuitive way to evaluate the accuracy of localization. Therefore, we propose two evaluation metrics, Meter-level Accuracy (MA) and Relative Distance Score (RDS), to evaluate the localization accuracy from the space- and model-level respectively.

1) *Meter-level Accuracy (MA)*: One intuitive way to measure localization accuracy is to analyze it from the level of spatial distance, that is, to measure the Meter-level Accuracy (MA). The expression of the MA evaluation index proposed in this paper is as follows:

$$MA@K = \frac{\sum_{i=1}^N 1_{SD < Km}}{N} \quad (1)$$

$$1_{SD < Km} = \begin{cases} 1 & SD < Km \\ 0 & SD \geq Km \end{cases} \quad (2)$$

where SD refers to the real spatial distance in meters. K is an adjustable parameter. MA@K refers to the number of samples

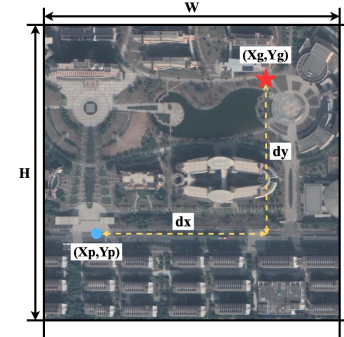


Fig. 7. The pentagram (X_g, Y_g) indicating ground-truth and the blue circle (X_p, Y_p) indicating the position predicted by the model.

with positioning errors within Km as a proportion of the total samples. In conclusion, MA@K can be represented as the accuracy of positioning error less than Km. The expression of SD is as follows:

$$SD = \sqrt{(\Delta x)^2 + (\Delta y)^2} \quad (3)$$

Δx denotes the meter-level error between the model prediction and the ground-truth in the longitude direction, and Δy denotes the meter-level error in the latitude direction.

2) *Relative Distance Score (RDS)*: Although MA can express the accuracy of positioning intuitively, it also has some shortcomings. 1) The setting of K has different effects on search maps of different scales. For large-scale search maps, after resizing, the spatial distance represented by a unit pixel will also become larger. Therefore, there will be a small deviation in the search map, but a large deviation in the spatial distance. 2) MA often needs to be expressed by a series of K, which is not convenient for directly expressing the performance of the model.

Based on the above discussion, we propose the Relative Distance Score (RDS) to evaluate from the model level. As shown in Fig. 7, (X_p, Y_p) represents the predicted coordinates and (X_g, Y_g) represents the ground-truth coordinates. The expression for Relative Distance (RD) is as follows:

$$RD = \sqrt{\frac{(\frac{dx}{w})^2 + (\frac{dy}{h})^2}{2}} \quad (4)$$

where $dx = |X_p - X_g|$ and $dy = |Y_p - Y_g|$. To make the RD perform in the same direction as the accuracy and distributed between 0 and 1, we convert the distance into a score, and the expression of the converted RDS is as follows:

$$RDS = e^{-k \times \sqrt{\frac{(\frac{dx}{w})^2 + (\frac{dy}{h})^2}{2}}} \quad (5)$$

where k is the scaling factor, which is set to 10 by default.

Why propose an RDS? First of all, RDS measures the performance of the model by the relative distance at the pixel level in the image, which can resist the scale transformation of the search map. The distance in RDS is no longer the real space distance, but the pixel distance in the image, which is beneficial to evaluate the performance from the model level. Second, the RDS is an overall score, which is distributed between 0 and 1, and is positively correlated with the positioning accuracy, that is, the higher the RDS, the higher the positioning accuracy. Finally, RDS uses an exponential method to expand the relative distance. When the distance increases, the RDS will return to 0 in an exponential form, which is in line with the expectation that the large distance deviation will be regarded as a wrong positioning.

V. EXPERIMENT

A. Implementation Details

In the experiments, all initial parameters of backbones are pre-trained on ImageNet [54]. The AdamW [55] optimizer is adopted with a weight decay of $5e-4$. Moreover, we incorporate a cosine annealing learning rate decay schedule, and the minimum learning rate is 1/100 of the initial learning rate. During the training phase, the UAV-view images are set at a resolution of 128×128 pixels, while the satellite-view images are formatted at 384×384 pixels in default. For the inference phase, the hanning window size is the same as the training setting of R in WBL.

B. DRL v.s. Image Retrieval

1) *Localization Performance*: To make the comparison as fair as possible, we uniformly use the real space meter-level accuracy (MA@K) to compare these two schemes. Since the test set of DenseUAV is constructed at intervals of 20m and the gallery contains images that perfectly match each query. We have additionally constructed a gallery of satellite images with a fixed sampling interval and not perfectly center-corresponding, which is consistent with the setting of the UL14. We have used some previous SOTA models in image retrieval for comparative experiments. The experimental results are shown in Table III. The Mix-Fusion structure of the

TABLE III
THE POSITIONING ACCURACY OF THE DRL METHOD BASED ON THE MIX-FUSION ARCHITECTURE IS SIGNIFICANTLY IMPROVED COMPARED TO THE IR-BASED METHOD. THIS IS MAINLY ATTRIBUTED TO THE ARCHITECTURAL ADVANTAGES OF DRL, WHICH DEEPLY INTERACTS WITH HETEROGENEOUS FEATURES AND OVERCOMES THE INHERENT ERROR PROBLEM.

Type	Method	Backbone	MA@3	MA@10	MA@20
Image Retrieval	Uni-1652 [1]	RN50	0.005	0.047	0.145
	Triplet Loss [56]	RN50	0.006	0.046	0.140
	LCM [57]	RN50	0.014	0.112	0.250
	LPN [34]	RN50	0.015	0.158	0.273
	RKNet [58]	RN50	0.021	0.177	0.317
	Uni-1652 [1]	ViT-S	0.069	0.541	0.723
	GeM [59]	ViT-S	0.076	0.526	0.707
	MCCG [36]	ConvNext-T	0.078	0.574	0.752
	FSRA [35]	ViT-S	0.079	0.580	0.743
	FSRA [35]	CvT13	0.029	0.265	0.423
DRL	Post-Fusion	CvT13	0.084	0.436	0.643
	Mix-Fusion	CvT13	0.131	0.625	0.837

TABLE IV
TIME AND STORAGE CONSUMPTION COMPARISON OF FSRA AND DRL. WITHOUT CONSIDERING PREPROCESSING, DRL IS STILL 7 TIMES FASTER THAN IR-BASED METHODS AND CONSUMES 3 TIMES LESS STORAGE.

Method	Time Consumption(s)				Storage Consumption(MB)			
	BSG	GFP	IAL	Total	SI	GI	GF	Total
FSRA(ViT-S)	4.2	6.5	43.2	43.2	4.8	44.6	10.8	15.6
Post-Fusion	0	0	6.2	~6.2	4.8	0	0	4.8
Mix-Fusion	0	0	6.0	~6.0	4.8	0	0	4.8

proposed DRL architecture has an absolute improvement of 9.4 points over the IR-based FSRA model on the MA@20 indicator. In high-precision positioning indicators such as MA@3, there is also an absolute improvement of 5.2 points.

2) *Efficiency*: The detailed time consumptions of DRL and image retrieval are counted in Table IV, which contains 600 samples. In order to ensure the fairness of the experimental data as much as possible, all operations are performed without using multi-process technology. The main steps are as follows:

- Building Satellite Gallery (BSG): Intensive split of satellite images to construct a gallery database just like the process in Fig. 1(a). The IR-based method takes about 4.2s, while DRL only requires a simple crop operation, which only takes microseconds.
- Gallery Forward Propagation (GFP): Extract features for all images in the gallery. The consumption of the IR-based scheme is determined by both the sampling interval and the gallery size, while DRL does not require this step.
- Inference And Localization (IAL): Real-time inference and localization. For IR-based methods, Part of the overhead is UAV image inference, which accounts for a relatively small proportion. The main overhead comes from the calculation of feature similarity between UAV and satellite gallery. For DRL, only model inference is needed.

the DRL architecture has achieved significant advantages in inference through the end-to-end architecture. Even without considering the time-consuming preprocessing of BSG and GFP, the DRL method is still more than 7 times faster than the IR-based method.

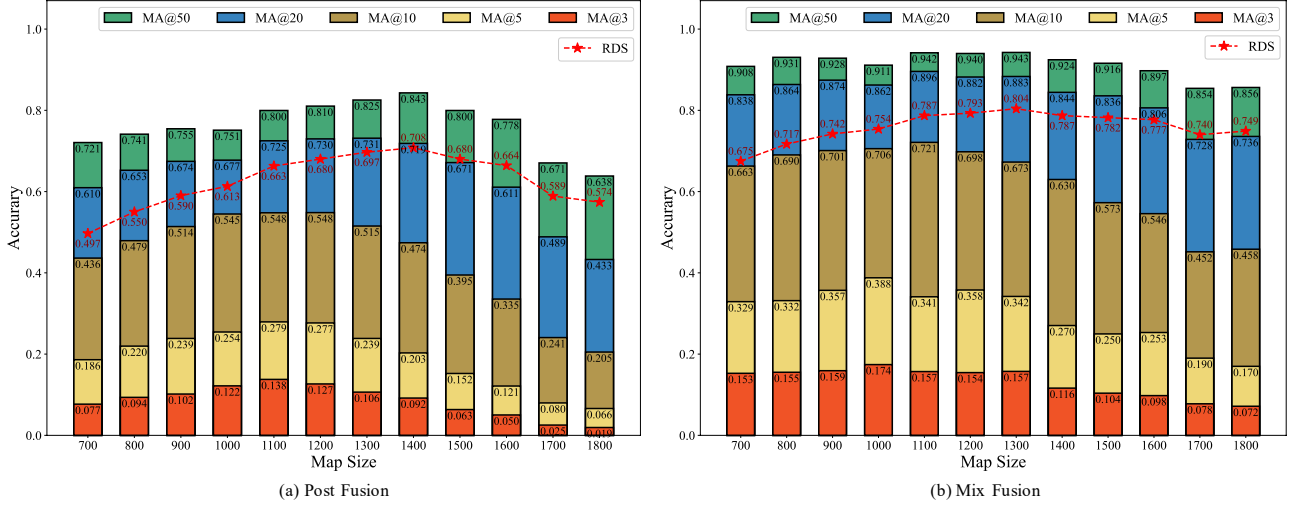


Fig. 8. The impact of search mapsize on the positioning of two DRL structures, including RDS and MA@K indicators.

3) *Storage Consumption*: Next, the DRL and IR-based schemes will be compared from the perspective of storage. The results are shown in Table IV. The main steps are as follows:

- **Satellite Images (SI)**: Pre-downloaded satellite maps.
- **Gallery Images (GI)**: The IR-based method requires the additional generation of an image gallery (split from satellite images). The architecture of DRL only requires a satellite image to be loaded into memory.
- **Gallery Features (GF)**: The IR-based methods need to construct gallery features through model inference for gallery images, while DRL does not require this step.

For the IR-based method, the image generated by GI is an intermediate product. To ensure fairness, after removing this part of the consumption, a total of 15.6MB of memory is occupied. Due to its end-to-end characteristics, DRL also greatly reduces storage consumption, requiring only 1/3 of the IR-based method.

C. Impact of Satellite Spatial Scale

The spatial scale of the search map directly determines the fine-grained information of the input satellite image. Given that the scale of the model input is typically fixed, the model's resistance to scale shifts is important. To investigate the influence of different scales of search maps, we conduct a series of comparison experiments with satellite images ranging from 700 to 1800 pixels (0.294 meters/pixel).

On the one hand, the results of the Post-Fusion structure are presented in Fig. 8(a). We can observe a marked decline in the MA@K index as the scale of the satellite images increases, particularly evident in higher precision indicators like MA@3, MA@5, and MA@10. This trend can be easily rationalized. Given that the input size for all models is consistent, larger satellite image scales would mean that each pixel in the input image represents a larger actual spatial distance. For instance, when the mapsize is 700, the corresponding actual spatial distance is 205.9m. If the output feature map size from the model is 384, then the actual spatial distance between each pixel in the feature map is 0.536m. However, when the

mapsize is 1800, it becomes 1.206m. Although larger mapsize yield poorer localization performance, it is still necessary to use them. The reason lies in the fact that large mapsize can simplify the actual process of UAV self-localization. For instance, if one large image can resolve the localization of all positions, it eliminates the need for any post-processing operations on the application side. Further observing the change rule of RDS, we can find that the overall curve is relatively smooth, but it is worth noting that it does not perform well at small scales. This is mainly due to the fact that small-scale images cover less spatial information and the loss of edge information is exacerbated.

On the other hand, the results of the Mix-Fusion structure are shown in Fig. 8(b). It is evident that, compared to the Post-Fusion structure, Mix-Fusion exhibits better robustness to scale. Although it also follows the aforementioned trend, but smoother. This is primarily because deeper information interaction and fusion can assist the model in better learning spatial distribution information.

D. Impact of Query Distribution

During the application, the relative position of the query in the search map is uncertain due to factors such as time delays and flight speed. Therefore, it is important to explore the impact of query distribution. As shown in Fig. 9(a), we divide different ranges according to the relative spatial distance between the query and the center point of the search map. The MA curve is shown in Fig. 9(b). We can clearly see that there is a significant decrease in localization accuracy when queries are located in the region >1.0 . This is consistent with our expectations, to take an extreme example, when the query is located on 4 corners of the search map, only 1/4 of the information will be available, and the remaining 3/4 of the image information will not find the corresponding information on the search map, which undoubtedly increases the challenge of localization. Furthermore, Fig. 9(c) illustrates the number of samples, as well as the RDS for each region. The quantity distribution is approximately proportional to the area of the

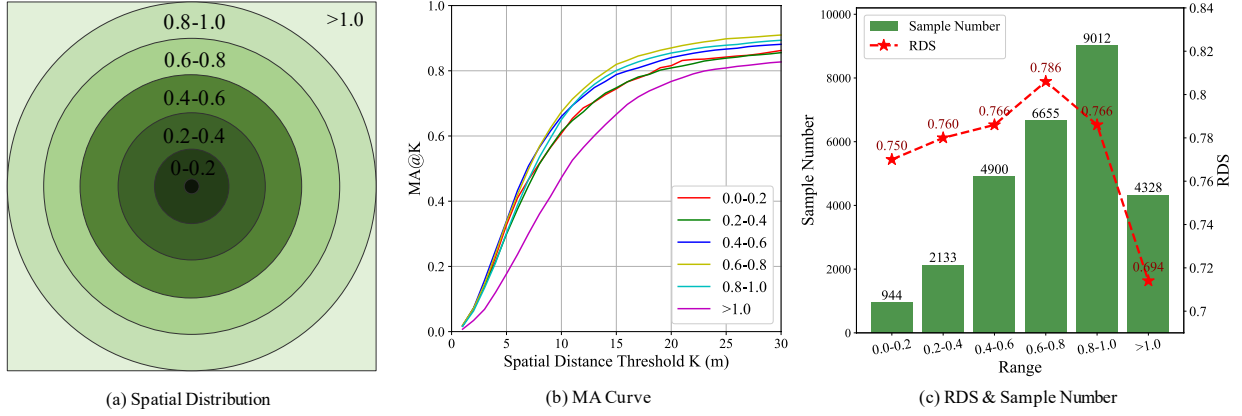


Fig. 9. (a) Query distributions. (b) MA curves under different spatial distributions. (c) RDS and sample number under different spatial distributions.

region, which results in fewer samples in the 0-0.2 range and more in the 0.8-1.0 range. From the RDS, we can similarly observe a cliff-like decline in the results for the >1.0 region. This decline is attributable to the loss of edge information, which greatly affects the discriminative ability of the model. In conclusion, queries located on the edge of the search map significantly reduce localization accuracy.

E. Impact of Flight Altitude

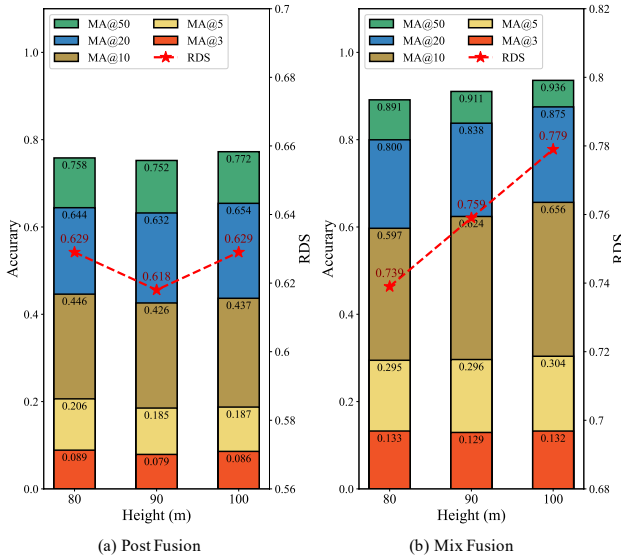


Fig. 10. MA@K and RDS of UAV at different flight altitudes.

In the application scenario, the altitude of the UAV flight is an uncertain factor, which directly affects the spatial scale of the query. Taking this into account, UL14 collected UAV images from 80, 90, and 100m altitudes. In the testing phase, data of different heights are divided for separate testing. The experimental results of the two architecture models are shown in Fig. 10. It can be seen that both MA@K and RDS show relatively stable performance at these three different heights. However, due to factors such as the height limit of the UAV, higher data have not been collected to fully verify the robustness of the model to flight heights.

VI. MODEL ANALYSIS

In this section, we will conduct experiments and analysis at the model level. Firstly, in Section VI-A, we will systematically analyze the two proposed model architectures. And RDS experiments were introduced in Section VI-B. Then, in Section VI-C, we will introduce two multi-scale approaches and analyze their experimental results. Subsequently, we will specifically analyze the impact of the backbone network and the loss function in Sections VI-D and VI-E. Last, we will analyze some model-related hyperparameters, including input scale, weight sharing, padding, and present the relevant content in Sections VI-F, VI-G, and VI-H, respectively.

A. Model Structure

1) *Post-Fusion*: We conducted experiments on the three fusion methods, *GC*, *SAF*, and *CAF*, as shown in Table V. In terms of computation and parameter quantities, the computation of *SAF* is 0.2GMacs higher than that of *GC*, and its parameter quantity is 0.3M larger than that of *GC*, with *CAF* being slightly lower than *SAF*. Regarding inference speed, *SAF* is significantly slower than *GC* and slightly slower than *CAF*. However, in terms of model accuracy, the fusion method of *GC* significantly outperforms the *CAF* and *SAF* methods across all indicators. This is a perplexing phenomenon. In the field of SOT, the fusion methods based on attention can interact more fully with features and help alleviate scale inconsistency issues. We attribute this phenomenon to the data level. Given that the input data is from different sources and that there are challenges like rotation uncertainty and spatial information inconsistency due to time, which increases the complexity of learning relationships between features. For attention methods, it means that more training data and iterations are required to learn general knowledge. Conversely, the *GC*-based method interacts with features using a sliding window, which restricts the spatial scale and makes it easier for the model to learn certain patterns.

2) *Mix-Fusion*: Unlike *Post-Fusion*, *Mix-Fusion* can be regarded as a single-stream structure that adds an attention module for feature fusion in blocks of the backbone. As shown in Table V, compared to the *Post-Fusion*, *Mix-Fusion* possesses significant advantages in terms of parameters, speed, and localization accuracy. Specifically, RDS improved from

TABLE V

THE IMPACT OF FUSION METHODS ON THE TWO ARCHITECTURES. FOR POST-FUSION, THREE FUSION METHODS WERE USED FOR ABLATION EXPERIMENTS CORRESPONDING TO THOSE DESCRIBED IN SECTION III-B1. THE EXPERIMENTAL CONTENT OF MIX-FUSION CORRESPONDS TO THE CHANNEL POOLING PART IN FIG. 2.

Structure	Fusion Method	Macs	Params	InferTime	RDS	MA@3	MA@5	MA@10	MA@20	MA@50
Post-Fusion	GC	13.3G	39.3M	24.0s	0.576	0.043	0.111	0.329	0.581	0.727
	SAF	13.5G	39.6M	27.5s	0.293	0.001	0.004	0.026	0.091	0.374
	CAF	13.4G	39.5M	27.0s	0.287	0.001	0.003	0.025	0.086	0.358
Mix-Fusion	Mix+AvgPool	13.1G	19.6M	22.4s	0.680	0.043	0.132	0.418	0.725	0.875
	Mix+Linear	13.2G	19.7M	21.3s	0.681	0.045	0.137	0.430	0.721	0.873
	Mix+GeM	13.1G	19.7M	21.6s	0.680	0.042	0.134	0.423	0.727	0.871
	Mix+GC	13.2G	19.6M	21.5s	0.670	0.056	0.146	0.427	0.711	0.850

TABLE VI

THE IMPACT OF USING DIFFERENT SCALING STRATEGIES IN RSC DATA ENHANCEMENT ON MODEL PERFORMANCE. AMONG THEM, [512, 768] REPRESENTS THE USE OF TWO SCALES, 512 AND 768, AND 512→718 REPRESENTS THE USE OF RANDOM SCALES FROM 512 TO 768.

Scale Type	Scale	RDS	MA@5	MA@20
Single Scale	512	0.529	0.109	0.524
	768	0.533	0.081	0.489
	1000	0.395	0.024	0.238
Multiple Scale	[512, 768]	0.607	0.127	0.628
	[768, 1000]	0.639	0.112	0.648
	[512, 768, 1000]	0.671	0.146	0.703
Random Scale	512→768	0.606	0.126	0.631
	768→1000	0.592	0.087	0.574
	512→1000	0.680	0.137	0.721

0.576 to 0.681, a boost of 10.5 points, and MA@20 increased by 14.6%. This is primarily due to Mix-Fusion enabling deep interaction of context information from the two domains and effectively overcoming the scale inconsistency problem introduced by the GC-based method. Furthermore, within the framework of Mix-Fusion, we adopted four different methods to compress the number of channels for training: AvgPool, Linear, GeM [59], and GC. As a result, the Linear method achieved the best result on RDS, reaching 0.681, and reached the fastest inference speed of 21.3s/1000samples. It is worth mentioning that the GC method effectively improves the short-distance positioning accuracy, which is the advantage of introducing additional prior inductive bias.

B. RSC Data Augmentation

In the experiments, three scale augmentation schemes, namely single scale, multiple scale, and random scale are implemented. As shown in Table VI, [512, 768] denotes scales of 512 and 768, meanwhile, 512→768 indicates the random integers down to 512 and up to 768. It can be observed that models trained using single scale underperform in localization tasks. In contrast, both multiple scale and random scale methods result in substantial improvements. This observation aligns with the expectations, as having the model learn universal rules of localization across different scales during training can enhance the model's robustness to scale during testing. Ultimately, our baseline model employs the random scale (512→1000) augmentation scheme.

C. Feature Scales

1) *Multi-Scale Structure*: Experiments for two multi-scale schemes are formulated as shown in Table VII. Overall, it can be seen that the feature pyramid has achieved noticeable improvements over the brutal upsampling operation. For instance, the RDS of feature pyramid on the (H/1, W/1) scale has increased from 0.730 to 0.758, which is a nearly 2.8 points rise, and an increase of 6.4% on MA@10. Similarly, there are noticeable improvements on other scales as well. As we know, deep features always contain high-level semantic information, while shallow features carry information such as outline and color. The fusion of shallow features can provide more information for discrimination. However, the significant improvement also demonstrates the necessity of pyramid features in this task.

2) *Output Resolution*: The feature resolution of the final layer based on the CvT13 (*Baseline* in Table VII) is 1/16 of the original image. In the experimental process, we enlarged the feature scale by a factor of 2 until it matched the original input. However, for the feature pyramid structure, to restore the original scale, we additionally employed bilinear interpolation for up-sampling. As shown in Table VII, it is evident that with the increase of the output scale, most of the metrics show an upward trend, particularly on the small distance error metrics such as MA@3 and MA@5, which is expected. Due to the enlargement of the output scale, the real spatial distance corresponding to a unit pixel in the output heatmap will become smaller. Therefore, the actual distance deviation caused by wrongly predicting one pixel is multiplicatively reduced. This experiment also verifies a hypothesis that the output scale of the model significantly affects the performance.

D. Backbones

Existing mainstream backbone networks can be divided into CNN-based and Transformer-based. In experiments, ResNet50, EfficientNet-B5, and ConvNext-T are adopted in the CNN-based category. Also, Transformer-based models are divided into two types: Unstructured and Hierarchical. The Unstructured denotes the type of native ViT architecture including ViT-S, ViT-B, DeiT-S, whereas Hierarchical refers to models possessing hierarchical architecture such as PViT-S and CvT13. To ensure fairness, all weights of backbones are pretrained from the timm [63] framework, and the Post-Fusion structure with GC is adopted.

TABLE VII
THE IMPACT OF DIFFERENT FEATURE SCALES AND FEATURE FUSION METHODS. $\rightarrow$ REPRESENTS AN UP-SAMPLING OPERATION.

Multi-Scale Structure	Output Scale	RDS	MA@3	MA@5	MA@10	MA@20	MA@50	MA@100
Baseline	(H/16, W/16)	0.680	0.045	0.137	0.430	0.721	0.873	0.939
Upsample	(H/8, W/8)	0.717	0.089	0.220	0.518	0.773	0.893	0.948
	(H/4, W/4)	0.731	0.100	0.244	0.557	0.795	0.899	0.949
	(H/2, W/2)	0.728	0.105	0.243	0.547	0.788	0.897	0.948
	(H/1, W/1)	0.730	0.109	0.251	0.561	0.790	0.893	0.946
Feature Pyramid	(H/8, W/8)	0.733	0.114	0.270	0.584	0.796	0.844	0.940
	(H/4, W/4)	0.732	0.117	0.267	0.573	0.802	0.888	0.936
	(H/4, W/4) $\rightarrow$ (H/2, W/2)	0.753	0.126	0.287	0.614	0.832	0.908	0.949
	(H/4, W/4) $\rightarrow$ (H/1, W/1)	0.758	0.131	0.298	0.625	0.837	0.912	0.953

TABLE VIII
THE IMPACT OF DIFFERENT TYPES OF NETWORK ARCHITECTURES INCLUDING CNN AND TRANSFORMER ON THE SELF-LOCALIZATION PERFORMANCE OF UAVS.

Type	Structure	Backbone	Macs	Params	InferTime	RDS	MA@3	MA@5	MA@10	MA@20
CNN	Hierarchical	ResNet50 [60]	13.7G	47.2M	9.9s	0.208	0.001	0.004	0.016	0.059
		EfficientNet-B5 [61]	7.44G	54.3M	33.1s	0.402	0.007	0.020	0.078	0.273
		ConvNext-T [62]	14.6G	55.7M	8.1s	0.465	0.022	0.058	0.198	0.412
Transformer	Unstructured	ViT-S [40]	13.9G	43.3M	9.9s	0.577	0.038	0.099	0.313	0.564
		ViT-B [40]	55.0G	171.5M	10.3s	0.595	0.050	0.126	0.363	0.597
		DeiT-S [41]	13.9G	43.3M	9.7s	0.597	0.048	0.123	0.358	0.604
	Hierarchical	PvT-S [42]	12.0G	47.3M	18.3s	0.524	0.020	0.054	0.169	0.470
		CvT13 [43]	13.3G	39.3M	25.2s	0.576	0.043	0.111	0.329	0.581

TABLE IX
THE IMPACT OF DIFFERENT LOSS FUNCTIONS ON PERFORMANCE.

Loss	RDS	MA@5	MA@10	MA@20
Focal Loss [64]	0.730	0.261	0.578	0.794
CrossEntropy Loss [65]	0.750	0.276	0.606	0.828
Balance Loss [66]	0.654	0.197	0.455	0.676
Weighted Balance Loss	0.758	0.298	0.625	0.837

The experimental results are presented in Table VIII, leading to several key conclusions. Firstly, Transformer-based backbones significantly outperform CNN-based ones. This superiority is largely attributed to the UAV self-localization task's inherent challenges, such as domain differences, rotational uncertainties, viewpoint offsets, and temporal variations in spatial information. These challenges necessitate networks that can globally model and extract contextual semantic information which is a strength of Transformer architectures. Secondly, even unstructured Transformer models exhibit relatively strong performance, likely due to the limited utilization of shallow features in this context. Among the hierarchical models, the CvT13 backbone network stands out, not only for its hierarchical feature output but also for its overall performance. Consequently, CvT13 is adopted as the backbone network for our baseline, as its hierarchical structure facilitates multi-scale feature fusion and demonstrates excellent performance.

E. Loss Functions

1) *Loss Type*: In the experiments, we trained with four different loss functions: cross-entropy loss, focal loss, balance loss, and the proposed weighted balance loss (WBL). The experimental results are shown in Table IX. The performance of using balance loss alone is far worse than using cross-entropy loss or focal loss. As a result, a complete balance

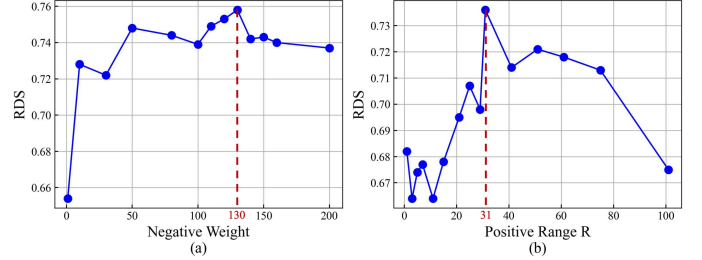


Fig. 11. The impact of negative sample weight and positive sample sampling range on RDS index.

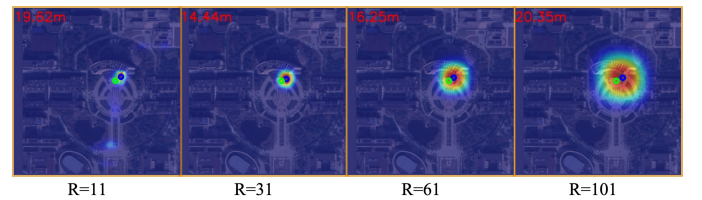


Fig. 12. The influence of R on the thermal distribution of the final output heatmap. As R increases, the thermal distribution takes the form of diffusion.

of positive and negative samples is not conducive to model convergence. Since the number of positive samples is much smaller than that of negative samples, the weight assigned to a single negative sample will be much smaller than that of positive samples after balancing. However, the results based on WBL lead to a 10-points RDS increase, and all MA@K indicators also achieve significant improvements.

2) *Negative Weight*: To find the most suitable value of w_{neg} , we made a lot of experiments with w_{neg} as the variable, and the experimental results are shown in the Fig. 11(a). When $w_{neg} = 1$, the proposed WBL is equivalent to balance loss. When w_{neg}

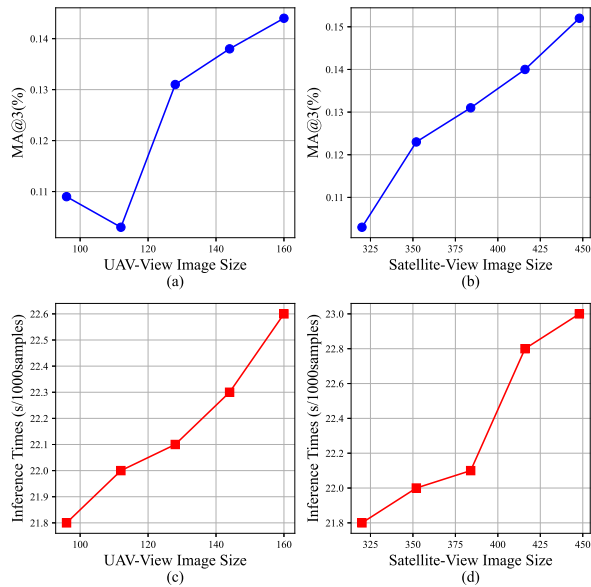


Fig. 13. The impact of varied input scales on the performance.

reaches 130, all indicators are reaching the optimal state, as w_{neg} keeps increasing, and all indexes gradually decrease. Last, $w_{neg} = 130$ is adopted in our baseline.

3) *Positive Range*: Positive range refers to the range occupied by positive samples in the feature map. Its setting method is shown in Fig. 5. This experiment uses the R parameter in both training and testing. In training, it represents the range of positive samples. In testing, the hanning window size is consistent with R . In fact, as shown in Fig. 12, the value of R directly determines the degree of dispersion of the output heat map. The larger the R value, the more divergent the heat distribution. From the visualization, it can be found that overly focused and overly divergent thermal distribution are unfavorable factors, so a moderate parameter R needs to be selected to tune performance. Finally, we conducted experiments on different values of R , as shown in Fig. 11(b). The optimal performance was achieved when $R = 31$.

F. Input Scale

High-resolution input can provide more fine-grained information. We conduct two sets of comparison experiments on input images. One group is to control the UAV image scale as 128 pixels and change the satellite image scale (320→448 pixels; stride=32). The other group is to control the satellite image scale as 384 pixels and change the UAV image scale (96→160 pixels; stride=16). The results are shown in 13(a)(b), we can conclude that as the input scale increases, the positioning performance generally shows a positive correlation trend. Additionally, the inference time is counted in Fig. 13(c)(d). It can be seen that the effect of the image scale variation of UAV on the inference time is smaller, mainly because the image scale of UAV is much smaller compared to satellite. Last, the resolutions of satellite and UAV images used for our baseline are 384 and 128 pixels, respectively.

TABLE X
ABLATION EXPERIMENTAL RESULTS OF PADDING OPERATION AND SHARED WEIGHTS. IT CAN BE OBSERVED THAT PADDING CAN PLAY A POSITIVE ROLE IN BOTH ARCHITECTURES.

Structure	Padding	Sharing	RDS	MA@3	MA@20
Post-Fusion	×	×	0.407	0.016	0.226
	✓	×	0.576	0.111	0.581
	✓	✓	0.276	0.003	0.093
Mix-Fusion	×	×	0.571	0.049	0.493
	✓	×	0.680	0.137	0.721

G. Sharing Weight in Post-Fusion

Weight sharing can help reduce the parameters. As shown in Table X, we find that the training process converges very slowly and performance is extremely poor when weights are shared. That is because the two-source data inputs have a massive domain difference, with uncertainties like viewpoint bias and rotation. Additionally, unlike the CVGL tasks, this task needs to fuse features and learn the spatial consistency relationship between two domains. Non-sharing allows for more differences in the learning process of the two domains.

H. Padding in GC

Both structures of Post-Fusion and Mix-Fusion can use GC for feature fusion in their output stages. However, we find that padding in GC plays a crucial role in maintaining spatial position information. The padding setting is to make the kernel center slide over each pixel of the source, ensuring that the spatial information of the output feature map is consistent with the original source. As shown in Table X, the introduction of padding in GC has resulted in significant improvements (16.9-points boost in RDS for Post-Fusion and 10.9-points boost in RDS for Mix-Fusion).

VII. VISUALIZATION

According to the characteristics of the output feature map supervision, the heatmap shows the spatial distribution of response strength after completing the sigmoid. That is, the higher the value in the heatmap, the greater the probability. Based on this setting, we normalize the output feature map and directly visualize it to analyze the response results. The visualization results are shown in Fig. 14, which can be divided into 3 types of cases. From groups 1 and 2, it can be observed that the structure based on Mix-Fusion presents more positive results in thermal response, without ambiguous thermal distribution. We believe that this is mainly due to the deep feature interaction method of Mix-Fusion, which allows the model to better understand spatial semantic information and thus generate more confident results. From groups 3 and 4, it can be seen that the Mix-Fusion structure is better at small distance deviations because Mix-Fusion allows the backbone to learn relative spatial distribution information in shallow features, thereby more precise positioning. Last, groups 5 and 6 are cases where Mix-Fusion performed poorly. However, when we compared the original input UAV images, there was indeed a great deal of similarity in the locations of the mismatches. For such scenes with very similar images in space, it is indeed a

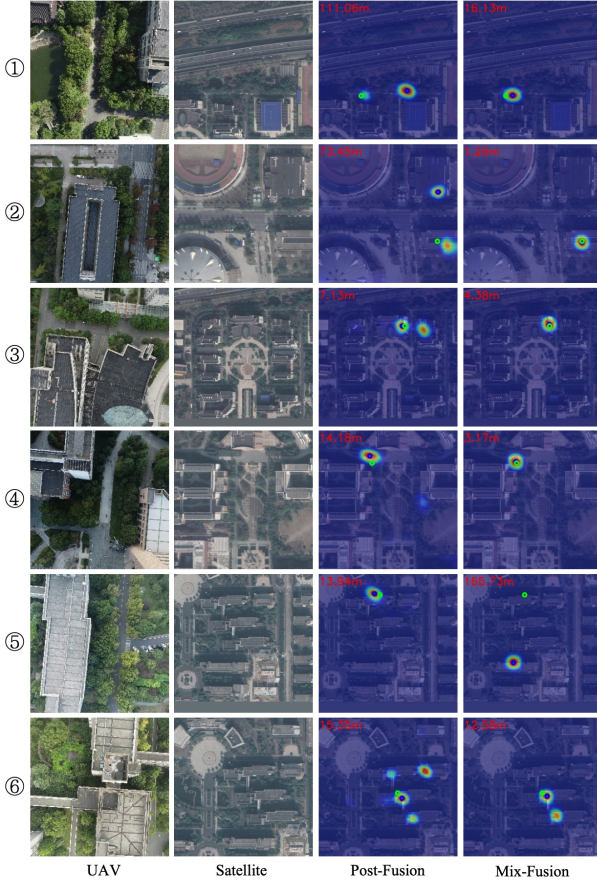


Fig. 14. The first and second columns are the UAV and satellite images, and the third and fourth columns are the heatmaps of the Post-Fusion and Mix-Fusion, respectively. The green circle is groundtruth, and the blue is the predicted position. The red number in the upper left corner is the positioning error, and the unit is meter.

huge challenge for the model to have stronger fine-grained mining and spatial information identification capabilities.

VIII. LIMITATIONS AND FUTURE WORK

There are still many aspects of DRL worthy of improvement. 1) Poor high-precision positioning: This is mainly limited by the large spatial distance between unit pixels of the output heat map. Adopting certain strategies to further expand the output heat map size, or learning an additional offset to further improve high-precision positioning performance is a direction worth exploring. 2) Overcoming the negative effects of scale and spatial distribution: As can be seen from Fig. 8 and Fig. 9(a)(b), changes in scale and spatial distribution currently have a negative impact on positioning, and design methods to mitigate it are also worth exploring.

IX. CONCLUSION

This paper introduces an efficient multi-source spatial feature interaction method, namely Drone Referring Localization (DRL), which not only interacts with heterogeneous features in a learnable manner and overcomes the inherent errors but also simplifies the entire localization process in an end-to-end manner. Meanwhile, two model structures named Post-Fusion

and Mix-Fusion are proposed, which are implemented as dual-stream and single-stream networks, respectively. Experiments show that the deep fusion method (Mix-Fusion) can achieve better performance. Furthermore, random scale crop (RSC) augmentation method is proposed to expand paired data with random scales and offsets. Also, weighted balanced loss (WBL) is designed to emphasize the influence of negative samples in the training process. To train the DRL model, a paired dataset (UL14) containing UAV and satellite images is constructed. Additionally, two evaluation metrics are presented to assess positioning accuracy: meter-level accuracy (MA) and relative distance score (RDS). In addition, this paper carries out a large number of experimental analyses from the data and model levels to provide some references. Finally, our proposed DRL architecture achieves better positioning performance (MA@20 +9.4%) compared to IR-based methods, while exponentially reducing time (1/7) and storage (1/3) consumption.

REFERENCES

- [1] Z. Zheng, Y. Wei, and Y. Yang, "University-1652: A multi-view multi-source benchmark for drone-based geo-localization," in *ACMMM*, 2020, pp. 1395–1403.
- [2] J. McCoy, A. Rawal, D. B. Rawat, and B. M. Sadler, "Ensemble deep learning for sustainable multimodal uav classification," *TITS*, vol. 24, no. 12, pp. 15 425–15 434, 2023.
- [3] A. V. Savkin and H. Huang, "Multi-uav navigation for optimized video surveillance of ground vehicles on uneven terrains," *TITS*, vol. 24, no. 9, pp. 10 238–10 242, 2023.
- [4] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, 2004.
- [5] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (surf)," *CVIU*, 2008.
- [6] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "Orb-slam: A versatile and accurate monocular slam system," *TRO*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [7] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," in *IJCV*, vol. 74, no. 1, 2007, pp. 59–73.
- [8] M. Dai, E. Zheng, Z. Feng, L. Qi, J. Zhuang, and W. Yang, "Vision-based uav self-positioning in low-altitude urban environments," *TIP*, pp. 1–1, 2023.
- [9] Z. Xia, O. Booi, M. Manfredi, and J. F. Kooij, "Visual cross-view metric localization with dense uncertainty estimates," in *ECCV*, Springer, 2022, pp. 90–106.
- [10] Z. Xia, O. Booi, and J. F. Kooij, "Convolutional cross-view pose estimation," *TPAMI*, 2023.
- [11] T. Lentsch, Z. Xia, H. Caesar, and J. F. Kooij, "Slicematch: Geometry-guided aggregation for cross-view pose estimation," in *CVPR*, 2023, pp. 17 225–17 234.
- [12] F. Fervers, S. Bullinger, C. Bodensteiner, M. Arens, and R. Stiefelhagen, "Uncertainty-aware vision-based metric cross-view geolocalization," in *CVPR*, 2023, pp. 21 621–21 631.
- [13] S. Zhu, T. Yang, and C. Chen, "VIGOR: Cross-view image geo-localization beyond one-to-one retrieval," in *CVPR*, 2021.
- [14] T. Cover and P. Hart, "Nearest neighbor pattern classification," *TIT*, vol. 13, no. 1, pp. 21–27, 1967.
- [15] T.-Y. Lin, Y. Cui, S. Belongie, and J. Hays, "Learning deep representations for ground-to-aerial geolocalization," in *CVPR*, 2015.
- [16] Y. Tian, C. Chen, and M. Shah, "Cross-view image matching for geo-localization in urban environments," in *CVPR*, 2017.
- [17] N. N. Vo and J. Hays, "Localizing and orienting street views using overhead imagery," in *ECCV*, 2016, pp. 494–509.
- [18] M. Zhai, Z. Bessinger, S. Workman, and N. Jacobs, "Predicting ground-level scene layout from aerial imagery," in *CVPR*, 2017.
- [19] L. Liu and H. Li, "Lending orientation to neural networks for cross-view geo-localization," in *CVPR*, 2019.
- [20] 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," *TCSVT*, pp. 1–1, 2023.

- [21] F. Castaldo, A. Zamir, R. Angst, F. Palmieri, and S. Savarese, "Semantic cross-view matching," in *ICCVW*, 2015.
- [22] T.-Y. Lin, S. Belongie, and J. Hays, "Cross-view image geolocalization," in *CVPR*, 2013.
- [23] T. Senlet and A. Elgammal, "A framework for global vehicle localization using stereo images and satellite and road maps," in *ICCVW*, 2011.
- [24] M. Bansal, H. S. Sawhney, H. Cheng, and K. Daniilidis, "Geolocalization of street views with aerial image databases," in *ACMMM*, 2011.
- [25] S. Workman and N. Jacobs, "On the location dependence of convolutional neural network features," in *CVPRW*, 2015.
- [26] S. Workman, R. Souvenir, and N. Jacobs, "Wide-area image geolocalization with aerial reference imagery," in *ICCV*, 2015.
- [27] S. Hu, M. Feng, R. M. H. Nguyen, and G. H. Lee, "CVM-net: Cross-view matching network for image-based ground-to-aerial geolocalization," in *CVPR*, 2018.
- [28] Y. Shi, L. Liu, X. Yu, and H. Li, "Spatial-aware feature aggregation for image based cross-view geo-localization," *NeurIPS*, vol. 32, 2019.
- [29] Y. Shi, X. Yu, L. Liu, T. Zhang, and H. Li, "Optimal feature transport for cross-view image geo-localization," *AAAI*, vol. 34, no. 07, pp. 11990–11997, 2020.
- [30] R. Zhu, M. Yang, K. Zhang, F. Wu, L. Yin, and Y. Zhang, "Modern backbone for efficient geo-localization," in *ACMMM*, 2023, pp. 31–37.
- [31] F. Deuser, K. Habel, M. Werner, and N. Oswald, "Orientation-guided contrastive learning for uav-view geo-localisation," in *ACMMM*, 2023, pp. 7–11.
- [32] Y. Shi, X. Yu, D. Campbell, and H. Li, "Where am i looking at? joint location and orientation estimation by cross-view matching," in *CVPR*.
- [33] Z. Zeng, Z. Wang, F. Yang, and S. Satoh, "Geo-localization via ground-to-satellite cross-view image retrieval," *TMM*, pp. 1–1, 2022.
- [34] T. Wang, Z. Zheng, C. Yan, J. Zhang, Y. Sun, B. Zhenga, and Y. Yang, "Each part matters: Local patterns facilitate cross-view geo-localization," *TCSVT*, 2021.
- [35] M. Dai, J. Hu, J. Zhuang, and E. Zheng, "A transformer-based feature segmentation and region alignment method for uav-view geo-localization," *TCSVT*, vol. 32, no. 7, pp. 4376–4389, 2022.
- [36] T. Shen, Y. Wei, L. Kang, S. Wan, and Y.-H. Yang, "Mccg: A convnext-based multiple-classifier method for cross-view geo-localization," *TCSVT*, 2023.
- [37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017, pp. 5998–6008.
- [38] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL*, Jun. 2019, pp. 4171–4186.
- [39] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," in *ICLR*, 2020.
- [40] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly et al., "An image is worth 16x16 words: Transformers for image recognition at scale," *ICLR*, 2020.
- [41] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jegou, "Training data-efficient image transformers & distillation through attention," in *ICML*, vol. 139, 2021, pp. 10347–10357.
- [42] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *ICCV*, 2021.
- [43] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "Cvt: Introducing convolutions to vision transformers," in *ICCV*, 2021, pp. 22–31.
- [44] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *ICCV*, 2021, pp. 10012–10022.
- [45] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," *ECCV*, 2020.
- [46] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *ICLR*, 2020.
- [47] R. Strudel, R. Garcia, I. Laptev, and C. Schmid, "Segmnet: Transformer for semantic segmentation," in *ICCV*, 2021, pp. 7262–7272.
- [48] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr, and L. Zhang, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *CVPR*, 2021.
- [49] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *CVPR*, 2017, pp. 1492–1500.
- [50] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *CVPR*, 2017, pp. 2117–2125.
- [51] M. Cen and C. Jung, "Fully convolutional siamese fusion networks for object tracking," in *ICIP*, 2018.
- [52] R. Zhao, W. Ouyang, and X. Wang, "Person re-identification by saliency matching," in *ICCV*, 2013, pp. 2528–2535.
- [53] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *ICCV*, 2015, pp. 1116–1124.
- [54] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009, pp. 248–255.
- [55] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *ICLR*, 2017.
- [56] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *CVPR*, 2015, pp. 815–823.
- [57] L. Ding, J. Zhou, L. Meng, and Z. Long, "A practical cross-view image matching method between uav and satellite for uav-based geolocalization," *Remote Sens.*, vol. 13, no. 1, p. 47, 2020.
- [58] J. Lin, Z. Zheng, Z. Zhong, Z. Luo, S. Li, Y. Yang, and N. Sebe, "Joint representation learning and keypoint detection for cross-view geolocalization," *TIP*, vol. 31, pp. 3780–3792, 2022.
- [59] Y. Deng, X. Lin, R. Li, and R. Ji, "Multi-scale gem pooling with n-pair center loss for fine-grained image search," in *ICME*, 2019, pp. 1000–1005.
- [60] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.
- [61] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *ICML*, pp. 6105–6114, 2019.
- [62] Y. Tang, Y. Li, F. Sun, C. Li, and X. Li, "Convnext: A generic convolutional neural network architecture for text classification," in *SIGIR*, 2020, pp. 1785–1788.
- [63] R. Wightman, "Pytorch image models," <https://github.com/rwightman/pytorch-image-models>, 2019.
- [64] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *ICCV*, 2017.
- [65] X. Wu, G. Shi, L. Lin, and X. Zhang, "Deep color image demosaicking with per-pixel cross-entropy minimization," *TIP*, vol. 28, no. 2, pp. 965–978, 2019.
- [66] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *CVPR*, 2019, pp. 9268–9277.