Detection using mask adaptive transformers in unmanned aerial vehicle imagery

YE Huibiao[1]*, FAN Weiming[2], GUO Yuping[2], WANG Xuna[2], and ZHOU Dalin[3]

1. China Telecommunication Corporation Zhejiang Branch, Hangzhou 310020, China

2. Innovation Center for Smart Medical Technologies & Devices, Binjiang Institute of Zhejiang University, Hangzhou 100059, China

3. School of Computing, University of Portsmouth, Portsmouth, PO1 3HE, UK

(Received 27 July 2024; Revised 16 October 2024)

Drone photography is an essential building block of intelligent transportation, enabling wide-ranging monitoring, precise positioning, and rapid transmission. However, the high computational cost of transformer-based methods in object detection tasks hinders real-time result transmission in drone target detection applications. Therefore, we propose mask adaptive transformer (MAT) tailored for such scenarios. Specifically, we introduce a structure that supports collaborative token sparsification in support windows, enhancing fault tolerance and reducing computational overhead. This structure comprises two modules: a binary mask strategy and adaptive window self-attention (A-WSA). The binary mask strategy focuses on significant objects in various complex scenes. The A-WSA mechanism is employed to self-attend for balance performance and computational cost to select objects and isolate all contextual leakage. Extensive experiments on the challenging CarPK and VisDrone datasets demonstrate the effectiveness and superiority of the proposed method. Specifically, it achieves a mean average precision (mAP@0.5) improvement of 1.25% over car detector based on you only look once version 5 (CD-YOLOv5) on the CarPK dataset and a 3.75% average precision (AP@0.5) improvement over cascaded zoom-in detector (CZ Det) on the VisDrone dataset.

Document code: A **Article ID:** 1673-1905(2025)02-0113-8

DOI https://doi.org/10.1007/s11801-025-4185-7

1. Introduction

Unmanned aerial vehicles (UAVs) are capable of performing diverse tasks at high altitudes or in locations difficult for humans to reach. With significant advancements in UAV technology, they have demonstrated outstanding performance in various application scenarios. For instance, UAVs have shown extensive potential in fields such as military reconnaissance, agricultural monitoring, disaster assessment, and urban planning. Target detection is a key technology enabling UAVs to perform these tasks. By identifying and locating target objects, UAVs can achieve precise monitoring and response in complex environments. Recent breakthroughs in computer vision and deep learning have led to the development of models such as convolutional neural networks (CNNs[1]), single-stage detectors (e.g., you only look once (YOLO)[2]), and transformer-based detectors (e.g., detection transformer (DETR)[3]). These technologies have made it possible to accurately identify and locate targets from aerial images captured by UAVs. However, complex terrains and variable weather conditions introduce significant noise and interference in UAV aerial images, such as shadows and glare. Additionally, UAVs may experience vibrations and wind speed changes during flight, leading to image jitter and blur. Real-time requirements and computational resource constraints further increase the difficulty of target detection. This motivates researchers to develop more advanced algorithms and technologies to improve the accuracy and efficiency of target detection and recognition in complex environments.

In image classification and detection tasks, CNNs have demonstrated excellent performance. However, the convolution operations of CNNs typically focus only on local information, making it difficult to capture global contextual relationships[4]. Furthermore, training CNN models relies on a large amount of labeled data, resulting in high data requirements. To address these issues, YOLO emerged as a mainstream framework for object detection. Due to its simple structure, YOLO can maintain high detection speed and accuracy even with limited computational resources[4]. However, the effectiveness and efficiency of using YOLO or any object detection algorithm for human detection can be influenced by various factors, such as the implementation method, the

* YE Huibiao is an engineer at the China Telecommunication Corporation Zhejiang Branch. He received his master degree. His research interests are mainly in 5G communication technology application. E-mail: yhb.zj@chinatelecom.cn.

available computational resources, and general environmental conditions. With the development of attention mechanisms, computer vision researchers have gradually focused on transformer-based object detection methods[5], which have achieved competitive performance for space human-robot interaction. Transformer models introduce a self-attention mechanism that can better capture global contextual information, overcoming the limitations of traditional CNNs in this regard. However, these advantages come at the cost of increased computation, as evidenced by vision transformer (ViT)[6] and DETR[3].

In recent years, researchers have been dedicated to optimizing object detection models. ViT is an architecture that uses transformer models to process images by segmenting them into fixed-size patches and capturing the global relationships between patches through a self-attention mechanism to achieve efficient feature extraction. However, due to the lack of inductive bias inherent in CNNs, ViT relies heavily on large-scale data and may perform poorly in handling pixel-level details and spatial information. Small target (ST)-YOLOX-S further improves the efficiency of transformers, making it particularly suitable for efficient object detection on edge devices[7]. Previous research has mainly focused on improving convergence speed and detection performance, but the high computational cost and slow inference speed of end-to-end networks based on transformers need further refinement.

In this work, we propose a UAV target detection algorithm based on the mask adaptive transformer (MAT). MAT employs a window-based transformer architecture to achieve window token collaborative sparsification, reducing computational costs while improving performance. By utilizing the binary mask strategy, it adaptively selects the optimal sparsification method according to the complexity of different scenarios, achieving scene-specific sparsity optimization. To isolate all context leakage, we propose adaptive window self-attention (A-WSA), which effectively performs self-attention on tokens of varying window sizes to achieve optimal performance.

2. Related work

2.1 UAV aerial photography target detection

Current research on UAV aerial target detection focuses on enhancing detection accuracy and real-time performance, particularly in recognizing targets in complex backgrounds and detecting small objects. For example, lightweight network architectures designed using sparse convolution techniques including query detection (Querydet)[8] and context-enhanced adaptive sparse convolutional network (CEASC)[9]. Querydet achieves fast and accurate small object detection by utilizing a sparse detection head and introducing a new query mechanism to accelerate the inference process of feature pyramid network (FPN)-based detectors. CEASC employs a plug-and-play detection head optimization method using CEASC and an adaptive multi-layer mask scheme. Additionally, universal ViT (UViT)[10] combines multi-level feature fusion and multi-scale feature interaction mechanisms, optimizing the performance of image classification and target detection tasks. However, UViT may perform poorly in tasks requiring global contextual information, as its design focuses on local information interaction. To address this, we propose a new method. By introducing a screening module, we can effectively enhance feature information. This screening module reduces redundant information while retaining key features, making our method particularly effective in complex environments and small object detection.

2.2 Transformer target detection

Swin transformer[10] is a visual processing model built upon the transformer architecture. Introduce hierarchical local-global focus mechanism and window-based visual structure. In contrast to traditional ViT, swin transformer reduces the computational complexity of global attention through non-overlapping window grouping mechanisms. However, its window grouping approach may pose challenges in capturing comprehensive global details and precise spatial relationships in unstructured and high-dynamic range visual scenes.

Lite DETR[11] achieves reductions in model complexity and computational costs by minimizing the number of transformer encoder layers. Specifically, lite DETR implements a strategy of less frequent updates to low-level features and introduces an interleaved updating mechanism, which effectively reduces computational expenses. However, its lightweight design may limit the comprehensive capture of detailed features and global contextual information, potentially reducing accuracy in managing complex scenes or tasks demanding high-resolution features.

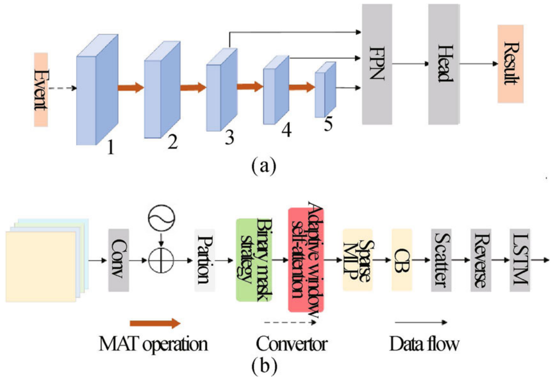
Cswin transformer[12] extends the capabilities of swin transformer by enhancing computational efficiency through a cross-shaped sliding window attention mechanism. Specifically, Cswin transformer alternates sliding windows horizontally and vertically with its cross-shaped window attention, covering more regions in images to achieve comprehensive and refined feature extraction. However, the reliance on a specific cross-shaped window pattern may limit Cswin transformer's effectiveness in capturing global features of targets with irregular shapes or significant scale variations.

Pyramid vision transformer v2 (PVTv2)[13] enhances upon the original PVT[14] by introducing multi-scale feature extraction and more efficient attention mechanisms. PVTv2 utilizes attention mechanisms with linear complexity and introduces a finer-grained feature pyramid structure, facilitating enhanced processing of visual information across diverse scales and complexities. However, the attention mechanisms and feature pyramid design of PVTv2 are primarily tailored to specific scales and feature patterns, which may limit its ability to capture all critical global features when handling highly complex and diverse large-scale urban landscape images.

3. Method

3.1 MAT

It illustrates the overall architecture of our method in Fig.1(a). Initially, it transforms the events occurring at each pixel into a collective representation of event volume, and fed into the MAT operation to extract multi-scale features. Then, a convolutional layer transforms the event volume into tokens, which applies the sinusoidal positional encoding. Four consecutive modules extract spatial features from these tokens. Specifically, adaptive sparsification is achieved by token and patch selection on event data. It improves the issue of handling blank areas caused by global self-attention calculations in traditional transformer frameworks. Additionally, conventional sparse transformers using a fixed proportion of tokens in scenarios with uneven complexity may result in losing important information. We address this by using a binary masking strategy to dynamically select the optimal level of sparsification based on the complexity of the scene and the density of the temporal data. Subsequently, it uses a long short term memory (LSTM) layer to propagate temporal information to subsequent layers. Due to the multi-scale design, the subsequent tokens are processed through a convolutional layer to reduce spatial resolution, followed by the previously described process. Features extracted from the third, fourth, and fifth layers are relayed to the FPN, which transmits the processed features to the detection head to generate the final detection results.



(a)

(b)

Fig.1 (a) MAT network architecture; (b) Architecture of MAT operation

3.2 MAT operation

As shown in Fig.1(b), patches are obtained in the MAT operation by dividing tokens. It then uses a binary masking strategy to select patches and determine their importance. It sequentially selects important windows and patches within the windows based on their scores. Next, it applies A-WSA to the selected windows and patches. The attention-enhanced patches are subsequently relayed to the sparse multilayer perceptron (MLP) layer, followed by the optional context broadcast (CB) operation. Finally, the processed patches are scattered back and restored to their original shape. The MAT operation realizes the sparsification of windows and patches, thereby avoiding the loss of important information and ensuring efficiency. We will now detail the two main components of the MAT operation: the binary masking strategy and A-WSA.

3.3 The binary masking strategy

As illustrated in Fig.2, the binary masking strategy aims to determine the importance of each patch. Unlike other strategies that derive scores directly from token values or attention maps, this approach is a learnable and adjustable module designed to assess importance effectively.

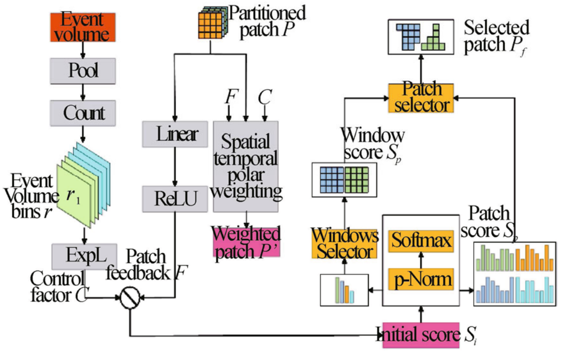


Fig.2 Architecture of binary masking strategy

Initially, the patch P divided from the token is processed through a linear layer and rectified linear unit (ReLU) to obtain the patch feedback F. The event volume is sent to the pooling layer to facilitate multi-scale feature extraction in a parallel branch. This down-sampling matches the receptive field of the patch. The event sparsity e is determined by calculating and concatenating the B nonzero ratio (r_1, r_2,…, r_B) of different event volume bins. Each ratio displays the sparsity of a specific event subset with unique time and polarity. The event sparsity e is then sent to the linear layer (ExpL) to be projected to the same dimension as the patch. The control factor C is obtained through this operation. The initial score S_i is defined using the patch feedback F and the control factor C:

$$S_i = \alpha \cdot \frac{F}{C} = \alpha \cdot \frac{\mathrm{ReLU}(w_F) \cdot P + b_F}{\exp(w_c) \cdot e}, \qquad (1)$$

where w_F and b_F represent the weights and biases of the linear layer, respectively, while w_c denotes the weights of the ExpL. To ensure that the control factor C is positive, an exponential operation $\exp()$ is performed on w_c, and α is a hyperparameter that controls absolute sparsity. Observational experiments have shown that C has a higher initial score when representing symbols of the most important objects.

Weighted patches P are obtained from tokens through the space-time-polarity (STP) mechanism. The STP calculation enables the feedback component and ExpL to be learnable. To obtain the spatial weight w_s and the time-polarity weight w_{tp}, a Sigmoid layer is applied to the patch feedback F and the control factor C. The weighted patch P' can be defined as follows based on the product of these weights:

$$P' = w_s \cdot w_{tp} \cdot P = \mathrm{Sigmoid}(C) \cdot \mathrm{Sigmoid}(F) \cdot T. \qquad (2)$$

The STP weighting process enables the model to highlight patches based on spatial and temporal context, aligning the network's sparse processing with the most salient features in both domains.

To select important windows and patches, we utilize the initial score S_i and the weighted patch P'. However, normalizing the initial scores among tokens shows insufficient discrimination. Therefore, to enhance effective discrimination, we first calculate the p-norm of S_i to obtain the normalized scores for each patch and window, respectively. By utilizing the selected P, the exponential relationship between the normalized scores and event sparsity can be determined. Then, a SoftMax operation is applied to amplify the differences between normalized scores, which better highlights certain values. The enhanced patch score S_p and enhanced window score S_w are described as follows:

$$S_p = \mathrm{SoftMax}(\|S_i\|_p^c), \qquad (3)$$

$$S_w = \mathrm{SoftMax}(\frac{\|S_i\|_p^{c,w}}{N_p}), \qquad (4)$$

where $\|\cdot\|_p^c$ and $\|\cdot\|_p^{c,w}$ denote the p-norm computed along the channel dimension and along the channel and window dimensions, respectively, dividing by N_p scales, and the normalized window scores by the number of patches within the window.

In scenes with dense events, the score distribution is uniform, retaining more windows and patches. However, in sparse scenes, lower score values predominate, making it easier to select less important windows and patches. To avoid missing critical object information, we specify two thresholds, θ_w and θ_p, in the process of selecting windows and patches, defined as follows:

$$\theta_p = \frac{P_w}{N_p}, \ \theta_w = \frac{P_w}{N_w}, \qquad (5)$$

where N_p and N denote the number of patches within a window and the total number of windows, respectively. P_w controls the strictness of selection, with parameter values ranging from 0.9 to 1. To capture important information, the values of θ_w and θ_p should be slightly below the average scores of patches and windows, ensuring consistency with the selected proportion and scene complexity.

Using a binary masking strategy on the input of the weighted patch P, patches are retained for windows where the score S_w exceeds θ_w. Further refinement retains patches where S_w exceeds θ_p. The retained patches are denoted as P_f.

3.4 A-WSA

Traditional window self-attention cannot be applied to selected tokens with uneven window sizes, as it is designed for equally sized windows and relies on parallel matrix multiplication. We have developed A-WSA, which enables self-attention on selected patches. This method effectively handles correlations between patches and prevents context leakage.

WSA begins with padding. Padded patches P_f are created by padding selected patches from each window to a uniform length, determined by the smallest number of patches selected in any window. Subsequently, multi-head self-attention (MHSA) is applied concurrently to the selected windows containing these padded patches. Additionally, both patches used for padding and initially unselected patches contribute to the attention map computation. The amount of padding is determined by the largest window size within a batch. A cover operation is implemented during attention map computation in A-WSA to prevent context leakage caused by uncertainty operations between patches and across batches. The entire process of A-WSA can be mathematically expressed as follows:

$$Q = P_p W^Q = \mathrm{Padding}(P_f) W^Q, \qquad (6)$$

$$K = P_p W^K = \mathrm{Padding}(P_f) W^K, \qquad (7)$$

$$V = P_p W^V = \mathrm{Padding}(P_f) W^V, \qquad (8)$$

$$P_a = \mathrm{UnPading}(\mathrm{SoftMax}(Mask + QK^{\mathrm{T}})V), \qquad (9)$$

where W^Q, W^K, W^V are the linear weights of the padding patch transformed into query Q, key K and value V, respectively. Padding represents the padding operation. Cover matches the attention map in size when other values are set to zero. In addition, it also contains large negative values in the columns corresponding to the patches used for padding. The cover operation isolates the influence of the unselected patches in the SoftMax process. Note that the enhanced patch P_a is obtained by first performing a matrix multiplication between the attention map after SoftMax and V, and then removing the patch at the padding position by the unpadding operation.

After A-WSA processing, attention-enhanced patches P_a feed into a sparse MLP layer to reduce computational complexity. The CB operations then disseminate information among selected patches to enhance feature representation and extraction. The process is formulated as:

$$\mathrm{CB}(\hat{P}(n)) = \frac{1}{2}\hat{P}(n) + \frac{1}{2N}\sum_{n=1}^{N}\hat{P}(n), \qquad (10)$$

where $\hat{P}(n)$ denotes the nth output patch from the sparse MLP layer, and N represents the total number of selected patches. CB operations do not introduce additional computations but promote more lenient sparsity preferences. Finally, the processed patch returns to the original patch P and is restored from the window to its original shape.

4. Experiments

4.1 Datasets

In our study, we utilized the CarPK[15] and VisDrone[16] datasets as benchmarks for evaluating UAV target detection algorithm performance. The CarPK dataset consists of 1 448 images collected from multiple parking lots, each annotated with precise bounding boxes delineating over 89 000 vehicles. The VisDrone dataset, designed specifically for UAV-based object detection and tracking, comprises 10 209 images and 179 264 video frames. It includes annotations for more than 2 600 000 objects across 10 predefined categories, such as pedestrians, vehicles, and bicycles, each labeled with precise bounding boxes. Our experiments strictly followed standard data generation procedures, object class definitions, and training/validation dataset partitions to ensure fair comparisons and reliable results. We selected the CarPK dataset for conducting ablation studies to comprehensively analyze and evaluate various factors influencing MAT performance.

4.2 Experimental setup

Our experiments were conducted using an algorithmic platform based on Pytorch and CUDA. We cropped the images in the CarPK and VisDrone datasets to 640×360 pixels. Scaling, resizing, and horizontal flipping were employed as data augmentation techniques during training. The experiments were executed on an NVIDIA GeForce RTX 4090. MaxViT served as the backbone network. The model was optimized using the SGF optimizer with a learning rate of 0.001, momentum of 0.9, and weight decay of 0.000 1. A linear warm-up strategy with a learning rate of 0.001 was applied for the initial 50 iterations, with a batch size of 4. For the comparative algorithms, we adhered to the settings specified in their respective papers. The experiment employed 10-fold cross-validation to enhance statistical significance. The dataset is divided into 10 subsets, or "folds" in the method. Each fold is used once for validation, while the remaining 9 folds are utilized for training. The final performance metrics are derived by averaging the results across all 10 cross-validation iterations. Average precision (*AP*@0.5, *AP*@0.75, *AP*@0.5: 0.95), precision, and recall were the primary evaluation metrics used to compare our method with existing methods. *AP*@0.5: 0.95 represents the average over all 10 intersection over union (IoU) thresholds within the range [0.5: 0.95] with a step size of 0.05. *AP*@0.5 and *AP*@0.75 correspond to IoU thresholds set at 0.5 and 0.75, respectively, for all detection categories. For attention-related experiments, we calculated the average floating-point operations per second (*A-FLOPS*), excluding computations attributed to convolutional layers. Inference time was also used to assess the model's speed.

4.3 Comparison to state-of-the-art

In our study, we benchmarked our approach against several state-of-the-art methods using the CarPK and VisDrone datasets. Tab.1 illustrates the performance of target detection algorithms on the CarPK dataset. Most current UAV target detection algorithms rely on CNN architectures. Tab.1 lists representative algorithms such as Faster-RCNN[17], single shot MultiBox detector (SSD)[18], fully convolutional network 8 sampling (FCN8s)[19], SegNet[20], and U-Net[21], based on CNN structures, evaluated for precision and recall. Our method achieves significantly higher performance than CNN-based networks, with precision and recall scores of 94.17% and 91.84%, respectively. Furthermore, we compared our method against various typical target detection algorithms in terms of average precision (*mAP*@0.5 and *mAP*@0.5: 0.95), including YOLO9000[22], YOLOv3[22], YOLOv4[22], SF-SSD[22], QueryDet[23], cascaded zoom-in detector (CZ Det)[24], and car detector based on you only look once version 5 (CD-YOLOv5)[25]. According to Tab.1, our proposed algorithm shows improvements of 1.25% in *mAP*@0.5 and 2.62% in *mAP*@0.5: 0.95 compared to the highest average precision achieved by CD-YOLOv5. This enhancement is attributed to our utilization of a transformer structure, which exhibits superior capabilities in learning image context features, extracting target features more effectively, and distinguishing between target and non-target objects. Consequently, this reduces instances of false positives and false negatives, thereby enhancing overall detection accuracy. Fig.3 illustrates predicted bounding boxes for the car. Due to scene complexity and object shape similarity, methods such as QueryDet encounter issues with false and missed detections (see columns 1, 2, and 3 in Fig.3).

Tab.1 Comparison of average precision, precision and recall with typical object detection algorithms on the CarPK dataset

Method	Precision (%)	Recall (%)	*mAP*@0.5 (%)	*mAP*@0.5: 0.95 (%)
Faster-RCNN	84.85	81.27	-	-
SSD	89.09	87.52	-	-
FCN8s	89.02	86.03	-	-
SegNet	87.77	86.89	-	-
U-Net	91.43	89.61	-	-
YOLO9000	-	-	20.90	-
YOLOv3	-	-	85.30	-
YOLOv4	-	-	87.81	-
SF-SSD	-		90.10	-
QueryDet	-	-	93.96	-
CZ Det	-	-	92.18	-
CD-YOLOv5	-	-	95.80	63.10
Our	94.17	91.84	97.05	65.72

In Tab.2, we evaluated various object detection algorithms based on CNNs and transformers using the VisDrone dataset, including feature selective anchor-free module (FSAF)[26], varifocal network (VFNet)[27], task-aligned one-stage object detection (TOOD)[28], disentangle your dense object detector (DDOD)[29], YOLOv8[30], drone-view object detection (DroneNet)[31], adaptive mixture regression network (AMRNet)[32], CZ

Det, as well as the transformer-based algorithms of DETR with improved denoising anchor boxes for end-to-end object detection (DINO)[33] and QueryDet. The experimental results demonstrate that our proposed algorithm achieves higher *AP* compared to the benchmark algorithms. Specifically, compared to YOLOv8, our algorithm shows improvements in *AP* of 9.82%, 13.59%, and 8.85%, respectively. Compared to the transformer-based network DINO, our algorithm increases *AP* by 5.24%, 4.63%, and 5.06%, respectively. These findings highlight the superiority and effectiveness of our approach in object detection tasks, particularly in managing complex datasets such as VisDrone. Fig.4 shows the output of the proposed model on VisDrone dataset. Despite employing a sparse querying approach to enhance computational efficiency, QueryDet inaccurately extracts information on small targets, resulting in reduced detection accuracy.

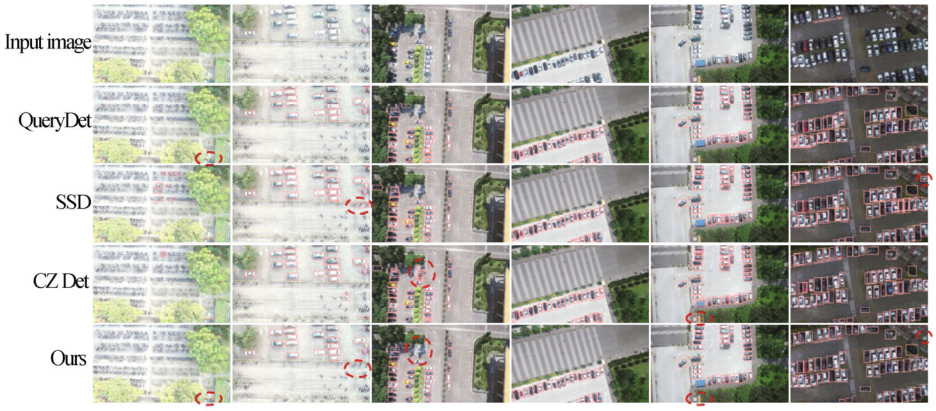


Fig.3 Visualizations of the comparison with other models on the CarPK dataset

Tab.2 Average precision comparison with typical object detection algorithms on the VisDrone dataset

Method	*AP*@0.5: 0.95 (%)	*AP*@0.5 (%)	*AP*@0.75 (%)
FSAF	20.8	36.4	20.5
VFNet	23.1	37.3	24.1
TOOD	24.4	39.8	25.3
DDOD	23.3	38.2	24.2
YOLOv8	25.9	42.9	26.4
DroneNet	29.6	50.4	29.6
AMRNet	31.7	52.7	33.1
DINO	30.48	51.86	30.29
QueryDet	21.58	38.51	21.29
CZ Det	32.2	52.74	30.91
Our	35.72	56.49	35.35

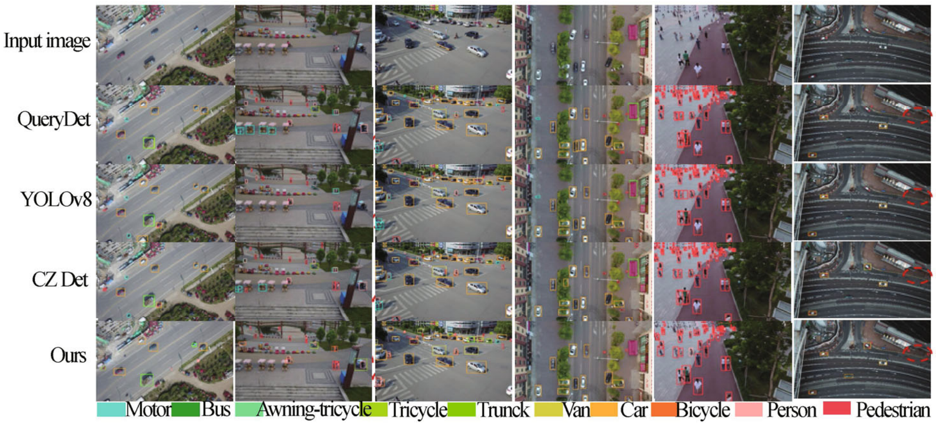


Fig.4 Visualizations of the comparison with other models on the VisDrone dataset

4.4 Ablation study

Several ablation experiments were conducted on the CarPK dataset to analyze the impact of key components of the proposed MAT. To ensure a fair comparison, the baseline and various model variants were configured with the same input size and parameter settings.

4.4.1 Binary mask method

We replaced the scoring component with established methods from other sparse transformers. During training, we also compared different selection methods by choosing only windows and patches, while maintaining consistency in other structures. As shown in Tab.3, the binary mask module outperforms other scoring methods, achieving an *AP*@0.5 of 97.05%. The binary mask strategy learns weights in both spatial and temporal polar domains, providing a more effective and context-aware approach to evaluate patch importance. Applying selection to both windows and patches results in a minimum runtime of 19.7 ms. This approach increases the model's use of critical information, forcing it to focus on the most significant features. Consequently, the model can more effectively represent and learn from densely compressed patch information.

Tab.3 Ablation study of binary mask strategy on the CarPK dataset

Method	Window	Patch	Window & patch	*AP*@0.5 (%)	Runtime (ms)
L2 Activation			√	94.29	20.4
Attention mask			√	94.61	20.4
Head scores			√	95.28	20.3
Our	√			94.37	20.1
Our		√		95.82	21.7
Our			√	97.05	19.7

4.4.2 Self-attention method

We compared various self-attention methods, as shown in Tab.4. Standard self-attention (SA) can cause inter-batch context leakage. Sparse self-attention (S-SA) uses padding to isolate different batches, but still results in context leakage between tokens. Masked sparse self-attention (MSSA) prevents both types of leakage through masking operations, but has higher *A-FLOPS*. Sparse window self-attention (S-WSA) introduces padding to adapt to different window sizes, but still leads to context leakage between tokens. A-WSA achieves optimal performance with the lowest computational complexity by being fully parallel and isolating all context leakage.

Tab.4 Ablation study of A-WSA on the CarPK dataset

Method	AP@0.5 (%)	*A-FLOPS*
SA	90.24	16.5
S-SA	93.57	20.5
MS-WSA	95.46	21.2
S-WSA	95.11	2.0
A-WSA	97.05	1.8

5. Conclusion

In this study, we propose a novel event-based drone target detection transformer, termed the MAT. MAT significantly reduces computational costs through an adaptive sparsification mechanism that supports collaborative window token sparsification. Specifically, we design a binary mask strategy and A-WSA to enable scene-aware adaptation. This approach dynamically optimizes sparsity across different scenes to enhance performance. Our method outperforms state-of-the-art alternatives on the Carpk and VisDrone datasets. While MAT enables real-time transmission in drone target detection tasks, it may experience missed detections in challenging scenarios such as nighttime or adverse weather conditions. This limitation motivates further exploration and utilization of object features in drone imagery.

Ethics declarations

Conflicts of interest

The authors declare no conflict of interest.

References

[1] LE C Y, BENGIO Y, HINTON G. Deep learning[J]. Nature, 2015, 521(7553): 436-444.

[2] REDMON J. YOLOv3: an incremental improvement[EB/OL]. (2018-04-08) [2024-03-12]. https://arxiv.org/abs/1804.02767.

[3] CARION N, MASSA F, SYNNAEVE G, et al. End-to-end object detection with transformers[C]//European Conference on Computer Vision, August 23-28, 2020, Glasgow, UK. Berlin: Springer, 2020: 213-229.

[4] HUANG G, LIU Z, VAN DER MAATEN L, et al. Densely connected convolutional networks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, July 21-26, 2017, Hawaii, USA. New York: IEEE, 2017: 4700-4708.

[5] YU J, GAO H, CHEN Y, et al. Deep object detector with attentional spatiotemporal LSTM for space human-robot interaction[J]. IEEE transactions on human-machine systems, 2022, 52(4): 784-793.

[6] CHEN X, FAN H, GIRSHICK R, et al. Improved baselines with momentum contrastive learning[EB/OL]. (2020-03-09) [2024-03-12]. https://arxiv.org/abs/2003.04297.

[7] ZHANG H, LU C, CHEN E. Obstacle detection: improved YOLOX-S based on swin transformer-tiny[J]. Optoelectronics letters, 2023, 19(11): 698-704.

[8] DU B, HUANG Y, CHEN J, et al. Adaptive sparse convolutional networks with global context enhancement for faster object detection on drone images[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 18-22, 2023, Vancouver, Canada. New York: IEEE, 2023: 13435-13444.

[9] BAO F, NIE S, XUE K, et al. All are worth words: a VIT backbone for diffusion models[C]// Proceedings of

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 18-22, 2023, Vancouver, Canada. New York: IEEE, 2023: 22669-22679.

[10] LIU Z, LIN Y, CAO Y, et al. Swin transformer: hierarchical vision transformer using shifted windows[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision, October 11-17, 2021, Montreal, Canada. New York: IEEE, 2021: 10012-10022.

[11] LI T, WANG J, ZHANG T. L-DETR: a light-weight detector for end-to-end object detection with transformers[J]. IEEE access, 2022, 10: 105685-105692.

[12] DONG X, BAO J, CHEN D, et al. Cswin transformer: a general vision transformer backbone with cross-shaped windows[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 19-24, 2022, Louisiana, USA. New York: IEEE, 2022: 12124-12134.

[13] WANG W, XIE E, LI X, et al. Pvt v2: improved baselines with pyramid vision transformer[J]. Computational visual media, 2022, 8(3): 415-424.

[14] WANG W, XIE E, LI X, et al. Pyramid vision transformer: a versatile backbone for dense prediction without convolutions[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision, October 11-17, 2021, Montreal, Canada. New York: IEEE, 2021: 568-578.

[15] HSIEH M R, LIN Y L, HSU W H. Drone-based object counting by spatially regularized regional proposal network[C]//Proceedings of the IEEE International Conference on Computer Vision, October 22-29, 2017, Venice, Italy. New York: IEEE, 2017: 4145-4153.

[16] DU D, ZHU P, WEN L, et al. VisDrone-DET2019: the vision meets drone object detection in image challenge results[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, October 27-November 2, 2019, Seoul, South Korea. New York: IEEE, 2019.

[17] MO N, YAN L. Oriented vehicle detection in high-resolution remote sensing images based on feature amplification and category balance by oversampling data augmentation[J]. The international archives of the photogrammetry, remote sensing and spatial information sciences, 2020, 43: 153-159.

[18] TANG T, ZHOU S, DENG Z, et al. Arbitrary-oriented vehicle detection in aerial imagery with single convolutional neural networks[J]. Remote sensing, 2017, 9(11): 1170.

[19] LONG J, SHELHAMER E, DARRELL T. Fully convolutional networks for semantic segmentation[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 7-12, 2015, Boston, USA. New York: IEEE, 2015: 3431-3440.

[20] BADRINARAYANAN V, KENDALL A, CIPOLLA R. Segnet: a deep convolutional encoder-decoder architecture for image segmentation[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 39(12): 2481-2495.

[21] RONNEBERGER O, FISCHER P, BROX T. U-net: convolutional networks for biomedical image segmentation[C]//Medical Image Computing and Computer-Assisted Intervention-MICCAI 2015, October 5-9, 2015, Munich, Germany. Berlin: Springer International Publishing, 2015: 234-241.

[22] YU J, GAO H, SUN J, et al. Spatial cognition-driven deep learning for car detection in unmanned aerial vehicle imagery[J]. IEEE transactions on cognitive and developmental systems, 2021, 14(4): 1574-1583.

[23] YANG C, HUANG Z, WANG N. QueryDet: cascaded sparse query for accelerating high-resolution small object detection[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 19-24, 2022, Louisiana, USA. New York: IEEE, 2022: 13668-13677.

[24] MEETHAL A, GRANGER E, PEDERSOLI M. Cascaded zoom-in detector for high resolution aerial images[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 18-22, 2023, Vancouver, Canada. New York: IEEE, 2023: 2046-2055.

[25] NGUYEN D L, VO X T, PRIADANA A, et al. Car Detector Based on YOLOv5 for Parking Management[C]//Conference on Information Technology and Its Applications, July 28-29, 2023, Da Nang, Vietnam. Cham: Springer Nature Switzerland, 2023: 102-113.

[26] ZHU C, HE Y, SAVVIDES M. Feature selective anchor-free module for single-shot object detection[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, October 27-November 2, 2019, Seoul, South Korea. New York: IEEE, 2019: 840-849.

[27] ZHANG H, WANG Y, DAYOUB F, et al. Varifocalnet: an IoU-aware dense object detector[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 19-25, 2021, Nashville, TN, USA. New York: IEEE, 2021: 8514-8523.

[28] FENG C, ZHONG Y, GAO Y, et al. TOOD: task-aligned one-stage object detection[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision, October 11-17, 2021, Montreal, Canada. New York: IEEE, 2021: 3490-3499.

[29] CHEN Z, YANG C, LI Q, et al. Disentangle your dense object detector[C]//Proceedings of the 29th ACM International Conference on Multimedia, October 21-25, 2021, Chengdu, China. New York: ACM, 2021: 4939-4948.

[30] JOCHER G, CHAURASIA A, QIU J. YOLO by Ultralytics[EB/OL]. (2023-01-01) [2024-03-12]. https://github.com/ultralytics/ultralytics/blob/main/CITATION.cff.

[31] WANG X, YAO F, LI A, et al. DroneNet: rescue drone-view object detection[J]. Drones, 2023, 7(7): 441.

[32] WEI Z, DUAN C, SONG X, et al. AMRNet: chips augmentation in aerial images object detection[EB/OL]. (2020-09-15) [2024-03-12]. https://arxiv.org/abs/2009.07168.

[33] ZHANG H, LI F, LIU S, et al. DINO: DETR with improved denoising anchor boxes for end-to-end object detection[EB/OL]. (2022-03-07) [2024-03-12]. https://arxiv.org/abs/2203.03605.