



OPEN YOLO-SAM an end-to-end framework for efficient real time object detection and segmentation

XiuMei Li✉, XiaHua Pu, WenChao Ling & XuDong Song

Although effective and practical YOLO methods have dominated the field of object detection, they rely on predefined and trained object categories, which limits their broad application. To overcome this limitation, YOLO-World enhances YOLO's open-vocabulary detection capabilities through modeling visual language and pretraining on large-scale datasets. Therefore, this manuscript proposes an improved object detection and segmentation model based on YOLO-World-S to improve detection efficiency and accuracy. The computational complexity and memory usage are reduced by introducing large-kernel separable convolutions into the RepVL-PAN of YOLO-World-S. The Neck incorporates a dynamic sparse attention mechanism (PSBRA module) to reduce the computational cost of traditional multi-head self-attention, while facilitating the integration of YOLO-World-S with EfficientSAM. In addition, the loss function is reconstructed to effectively resolve conflicts on shared features or optimization objectives between object detection and segmentation tasks. This method achieves mAPs (mean average precisions) of 58.8% and 308 FPSs (frames per second) on the COCO dataset, improving both accuracy and speed compared with those of the original baseline model.

Keywords YOLO-World, Self-attention mechanism, Feature fusion, Hybrid attention module

In recent years, YOLO has become one of the most influential real-time object detection algorithms in the field of computer vision because of its rapid iteration and performance. They have attracted widespread attention because of their high precision, efficiency and wide application in fields such as autonomous vehicles, security surveillance, and medical imaging. YOLO proposes a unified architecture with the ability to predict bounding boxes and class probabilities concurrently when it is contrasted with traditional two-stage object detection architectures^{1–6}. The image detection process is transformed into a one-time regression, and the categories and locations of multiple objects in a single forward stage can be predicted very quickly by the YOLO algorithm.

YOLOs have achieved fast inference speeds, but large datasets and longer training times are needed to obtain the prediction model. Moreover, it relies on predefined object categories, which means that it can detect only the object categories provided by the model during the training stage, so the model cannot flexibly recognize new categories. To solve this problem, a model that can detect new objects on the basis of text prompts provided by users through natural language processing capabilities is introduced into the YOLO-World method, which means that the prediction model does not need to be trained for each new category; thus, its adaptability and scalability are significantly enhanced. This architecture makes YOLO-World highly efficient at processing on zero-shot or unseen datasets⁷.

Designing the YOLO architecture requires careful trade-offs between accuracy and inference speed, making the task inherently challenging. To achieve a more efficient model architecture, researchers have explored various design strategies aimed primarily at reducing inference time and computational cost, while also enhancing the feature extraction capabilities of the backbone network, such as DarkNet⁸, CSPNet⁹, EfficientRep, and ELAN¹⁰. To improve multiscale feature fusion, neck structures, such as the PAN, BiC, GD, and RepGFPN models, have been investigated. However, considerable computational redundancy and low parameter utilization still exist within YOLO, leading to suboptimal efficiency. The performance of the model is relatively poor, leaving ample opportunity to improve its accuracy. YOLO-World is currently focused on object detection tasks, but its visual application tasks still need to be further explored.

To enhance the performance of YOLO-World-S, this work improves its feature extraction and fusion modules and incorporates the EfficientSAM¹¹ model through an end-to-end joint training framework for simultaneous object detection and segmentation.

YOLO-SAM seamlessly integrates EfficientSAM with YOLO-World-S within a unified joint-training pipeline, enabling simultaneous object detection and instance segmentation without stage-wise processing. In

Dalian Jiaotong University, Dalian, Liaoning, China. ✉email: 279417552@qq.com

this framework, EfficientSAM is not executed as a separate offline segmentation model; instead, it is directly embedded as a segmentation head in the YOLO backbone, sharing feature inputs with the detection branch and being jointly optimized in the same training process. Given a single input image, YOLO-SAM can produce detection boxes and corresponding instance segmentation masks in a single forward pass, without additional model invocation or secondary feature extraction.

The main technical contributions are summarized as follows:

- (1) Large Separable Kernel Attention (LSKA) in I-Pooling Attention – We replace the original multi-head convolutional attention (MHCA) in RepVL-PAN with LSKA to enhance the capture of spatial and channel dependencies while keeping computational and memory costs low. This design improves the ability to detect fine-grained details in complex scenes.
- (2) Partial Self-level Bi-level Routing Attention (PSBRA) for Efficient Global Modeling – We propose PSBRA to address the high computational complexity of multi-head self-attention (MHSA) in long-sequence scenarios. By integrating a bi-level routing mechanism, PSBRA effectively filters redundant information and enhances global feature representation, improving both efficiency and rare-category detection accuracy.
- (3) YS-CSPLayer for Cross-branch Feature Fusion – We design YS-CSPLayer, an MHSA-based fusion module that integrates YOLO-World’s semantic detection features with EfficientSAM’s fine-grained segmentation features. This enables precise cross-branch alignment and boosts both detection precision and mask quality.
- (4) Performance Gains with Balanced Efficiency – Extensive experiments on the COCO dataset show that YOLO-SAM achieves +2.1% AP over YOLO-World-S, while keeping FLOPs at 318G and FPS at 308, demonstrating a favorable trade-off between accuracy and efficiency.

The remainder of this paper is organized as follows: Sect. “Related work” reviews related work. Section “Proposed method” presents the proposed method. Section “Experiments” reports the experimental results. Section “Conclusion” concludes the paper.

Related work

YOLO-World is capable of zero-shot detection; even if the model has never seen a particular type of object, it can still recognize and detect through a corresponding textual description. This significantly enhances the flexibility and generalizability of the model in practical applications. YOLO-World is designed primarily for rapid object detection through natural language prompts, and it is particularly powerful. YOLO-World follows the standard YOLO architecture^{12–15} and uses a pretrained CLIP¹⁶ text encoder to encode the input text. Furthermore, RepVL-PAN (a reparameterizable vision-language path aggregation network) is proposed to connect text features with image features; therefore, visual-semantic representation is improved. During inference, the text encoder can be removed, and the text encoder can be embedded as reparameterized weights of the RepVL-PAN for efficient deployment. YOLO-World has strong open-vocabulary^{17–19} detection capabilities for pretraining many bounding boxes and text pairs. Additionally, YOLO-World explores a prompt-then-detect normal form to further increase the efficiency of open-vocabulary object detection in real-world scenarios. However, this method only detects the bounding boxes of objects and cannot perform precise pixel-level segmentation.

EfficientSAM is a lightweight model based on the Segment Anything Model (SAM). ViT-Tiny/Small replaces ViT-H in the image encoder of the SAM through knowledge distillation to reduce the computational load and parameter count. Thus, it is more suitable for resource-constrained application scenarios^{20,21}. EfficientSAM focuses on efficient image segmentation, and it can quickly generate high-quality object masks after detecting targets, so it is suitable for more refined image segmentation tasks. It can optimize the detection method via YOLO, which is based on grid partitioning because the localization accuracy of bounding boxes may not be as precise as that of algorithms based on sliding windows or region proposals in complex scenes. For example, when multiple objects overlap or occlude each other, the performance of YOLO may not match that of some more complex models. By introducing efficient SAM, the model can not only detect zero-shot scenarios but also accurately segment new category objects. This has transformed YOLO-World from a single-object detection model to a complete detection and segmentation model.

On the basis of the above, this paper introduces a model that combines YOLO-World and EfficientSAM to further increase the flexibility, accuracy, and efficiency of the model by integrating object detection with precise segmentation capabilities. This means that by combining the efficient image segmentation capabilities of the EfficientSAM with the zero-shot detection capabilities of YOLO-World, the model can achieve accurate detection and segmentation of new category objects. This not only expands the application capabilities of YOLO-World in open-vocabulary scenarios but also enables the model to handle more complex visual scenes. This paper improves the efficiency and accuracy of the model by optimizing feature extraction and lowering computational complexity. Additionally, the introduction of self-attention and multilayer routing mechanisms reduces the computational load while enhancing the processing of global information. The model demonstrates excellent performance in object detection and segmentation tasks through efficient feature fusion, especially in dealing with complex scenes. Similar strategies for improving feature representation and computational efficiency have also been explored in other multimedia processing tasks. For example, attention-based recursive transformers have been applied to image super-resolution²², large-strip convolution combined with Mamba networks has enhanced remote sensing object detection²³, and intelligent weighted object detectors have been proposed to enrich global image information²⁴. These approaches, while targeting different applications, share the common goal of strengthening feature extraction and global-local information integration, which is consistent with the design philosophy of YOLO-SAM.

Proposed method

To expand the application of YOLO-World in complex visual tasks, a series of optimizations have been carried out in this paper, and the improved model is named YOLO-SAM. The feature extraction backbone network of the YOLO-World model first extracts features of the objects from the input images. The PSBRA module subsequently efficiently processes and optimizes the feature maps by filtering and weighting information at each level, thereby reducing unnecessary computations and memory consumption. The feature fusion method of the YS-CSPLayer is then employed to integrate features from different depth levels. The effectiveness of feature fusion directly impacts the accuracy and robustness of the model in object detection and segmentation tasks. To enhance the feature extraction and fusion capabilities of the model, this study has optimized the feature extraction network and the feature fusion module separately, enabling the model to more effectively capture global information and local details, thereby improving the detection and segmentation performance. The structural diagram of the improved YOLO-SAM model is shown in Fig. 1.

Improved I-Pooling attention based on LSKA

In the field of feature extraction, traditional convolutional operations may fail to capture spatial and channel information fully when dealing with complex images, leading to insufficient feature extraction capabilities, thus affecting the performance of the model. To address this issue, this paper introduces the LSKA (large separable kernel attention) module to improve the I-Pooling Attention of the RepVL-PAN network. The capture of target details and the global information ability of the network are effectively enhanced by incorporating large separable kernel convolution operations while maintaining low computational complexity. LSKA applies weighted processing to multi-scale feature maps by first computing spatial and channel-wise weights for each feature map, then adjusting the features accordingly. This enables the model to focus more on target-relevant regions and suppress background noise, thereby improving detection accuracy while keeping the computation efficient. This improvement enables the network to more accurately process image features in complex scenes and provide more accurate feature representations for subsequent object detection and segmentation tasks.

LSKA decomposes a 2D convolutional kernel into horizontal and vertical 1D convolutional kernels; thus, the computational complexity and memory consumption are reduced. The $(2d-1) \times (2d-1)$ convolutional kernels of depthwise convolution are replaced with cascaded $1 \times (2d-1)$ and $(2d-1) \times 1$ convolutional kernels, and LSKA can

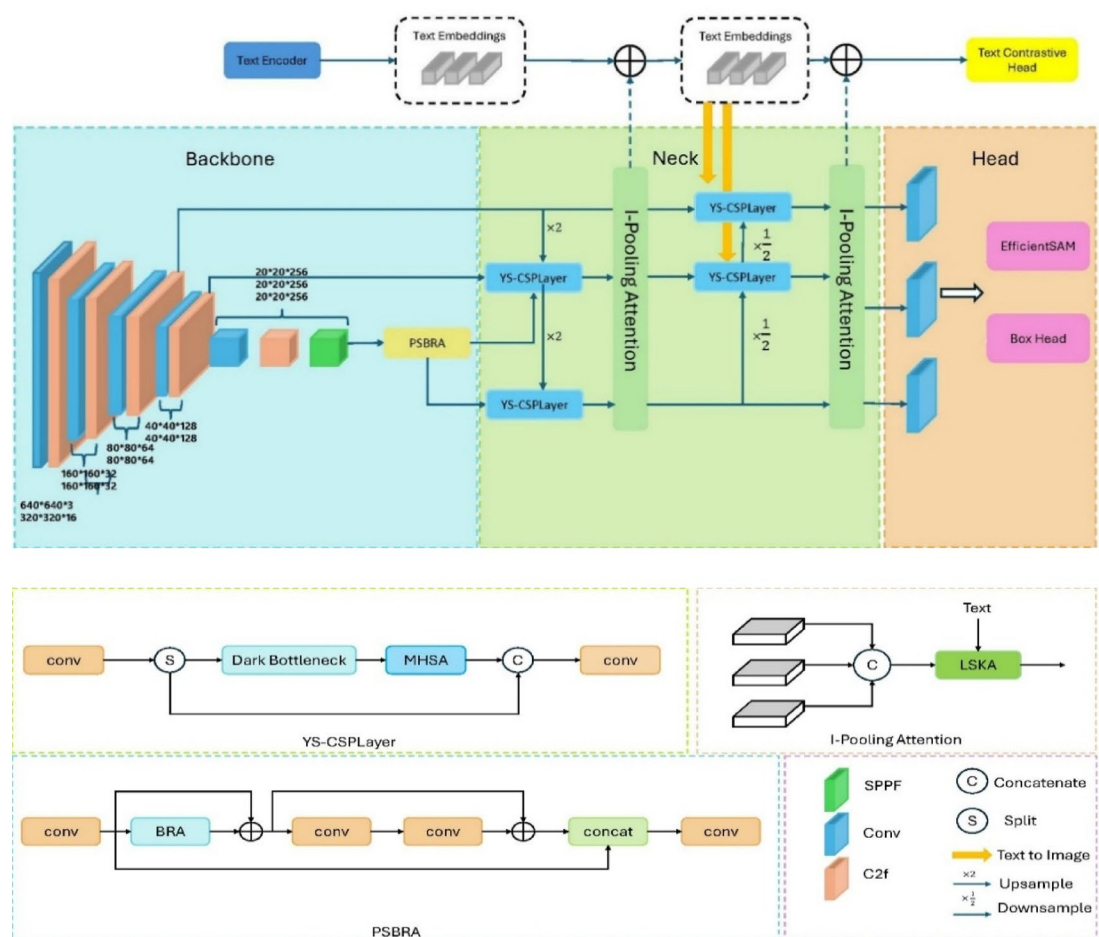


Fig. 1. The structure of the YOLO-SAM model.

effectively lower the quadratic increase in the parameter count associated with the increase in the convolutional kernel size. While maintaining performance, LSKA can capture both local and global feature information simultaneously through cascaded one-dimensional convolutional kernels.

The input feature maps undergo depthwise convolution with a kernel size of $1 \times (2d - 1)$ to convolve each channel, enhancing local feature information while effectively controlling the computational load. Subsequently, depthwise convolution with a kernel size of $(2d - 1) \times 1$ is applied to further extract local spatial information from the feature maps to obtain more detailed information about the channels. In the following convolution with a kernel size of $1 \times \frac{k}{d}$, depthwise separable convolution (DW-D-Conv) is performed. Then, another depthwise separable convolution with a kernel size of $\frac{k}{d} \times 1$ is applied to further process and integrate channel and spatial information. Finally, a 1×1 convolution is used to adjust the number of channels to obtain the final output feature map. The attention weights generated in this process are then applied to the input features through element-wise multiplication (Hadamard product), enabling the network to re-emphasize the most informative spatial regions and channels.

The LSKA module ensures the computational efficiency of depthwise convolution while increasing the kernel size to achieve a larger receptive field. The image patch tokens, obtained through feature extraction and max pooling, are utilized as keys and values in the Large Separable Kernel Attention module, which refines the original text embeddings W to produce the updated representations W' . The text embeddings are then updated as follows:

$$W' = W + \text{LargeSeparableKernelAttention} \quad (1)$$

PSBRA module

Self-attention mechanisms are widely used in various visual tasks because of their excellent capability for global modeling. However, they are associated with high computational complexity and high memory consumption. To solve this issue, the PSBRA module is proposed in this section because attention heads are the prevalent redundancy. The module aims to address the computational complexity of traditional self-attention mechanisms with long sequences. It reduces unnecessary computations while enhancing the ability of the model to process global information. The model maintains efficiency and performance in tasks involving long sequences and large-scale data.

As shown in Fig. 2, the PSBRA module adopts multibranch fusion and a two-level routing mechanism. The model can filter and weight information at each level through multilevel branch processing. This design not only optimizes computational efficiency but also enables the model to process global information. The input features first pass through an initial convolutional layer to extract local features. The feature maps are subsequently divided into multiple branches and processed through the two-level routing mechanism. One branch further extracts local features through convolutional operations and passes the extracted feature maps to subsequent convolutional layers for in-depth processing. Another branch introduces the BRA (bi-level routing attention) mechanism, which dynamically adjusts the focus on different parts of the input features by weighting them. Q (query), K (key), and V (value) are the three core components of the self-attention mechanism and are used to compute attention weights. Q and K are used to calculate the attention matrix A , with weights obtained through matrix multiplication (mm) and normalized by Softmax. This process helps to compute the similarity between different positions to determine which position should focus on which information of other positions. A is the normalized attention matrix, which helps the model adaptively adjust the weights of different features and ultimately computes the weighted output. The attention matrix A is multiplied by V (value), and the final weighted output O is obtained, which weights the features according to attention weights. The final output O is further processed through matrix multiplication (mm) to generate the final feature representation.

For a given 2D input feature map $X \in R^{H \times W \times C}$, we first divide it into $S \times S$ nonoverlapping regions, each containing $\frac{HW}{S^2}$ feature vectors. Subsequently, we derive $Q, K, V \in R^{\frac{HW}{S^2} \times \frac{HW}{S^2} \times C}$, via linear projections:

$$Q = X^r W^q, K = X^r W^k, V = X^r W^v \quad (2)$$

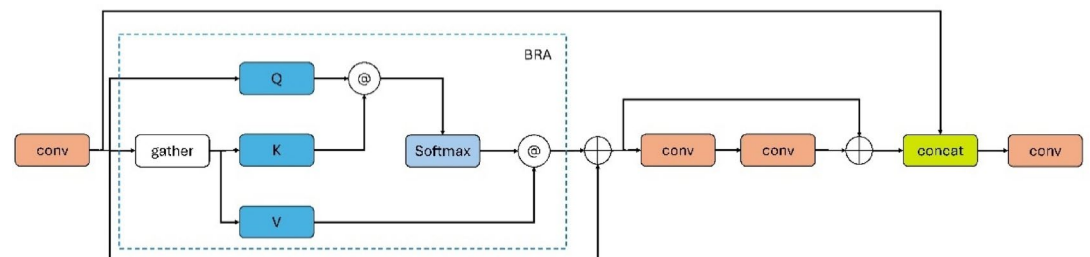


Fig. 2. Architecture of the PSBRA module, which combines Polarized Self-Attention (PSA) for fine-grained local feature modeling with Bi-level Routing Attention (BRA) for efficient global context aggregation. The BRA sub-module employs index selection to reduce the number of key-value pairs in attention computation, thereby lowering complexity while retaining essential long-range dependencies.

Here, $W^q, W^k, W^v \in R^{C \times C}$ are the projection weights for Q, K , and V , respectively. First, the region-level values of Q and K and $Q^r, K^r \in R^{S^2 \times S^2}$ are derived by computing the average value of each region of Q and K . Then, the adjacency matrix $A^r \in R^{S^2 \times S^2}$ of the region-to-region affinity graph is derived via matrix multiplication between Q^r and the transpose of K^r .

$$A^r = Q^r (K^r)^T \quad (3)$$

Based on the affinity matrix A^r , a region-to-region routing index matrix I_r is constructed, where $I_r(i, j)$ stores the index of the j -th most relevant region to region i . This routing index guides which regions should exchange information in the subsequent attention computation. Since directly attending to scattered regions on the feature map is inefficient on modern GPUs, we first gather the relevant key and value tensors according to I_r :

$$K^g = \text{gather}(K, I_r), V^g = \text{gather}(V, I_r) \quad (4)$$

where $K^g, V^g \in R^{S^2 \times \frac{HW}{S^2} \times C}$ are the gathered key and value tensors. The output of the BRA module is then computed as:

$$O = \text{Attention}(Q, K^g, V^g) + \text{LCE}(V) \quad (5)$$

where $\text{LCE}(\cdot)$ denotes a local context enhancement term implemented via depth-wise convolution with kernel size 5, which injects additional local spatial cues to complement the sparse global routing.

Then, a 1×1 convolution is used to connect and integrate with the subsequent feed-forward network (FFN). The residual output feature map is further processed through a convolutional operation to adjust its channel dimensions, yielding the final output of the module. In this work, the proposed PSBRA module is designed and integrated by substituting it for the terminal stage of the backbone network as well as the initial stage of the feature fusion layer. The improved model has the following advantages:

- (1) The bi-level routing mechanism can enhance information filtering capabilities. BRA (Bi-level Routing Attention) achieves finer contextual modeling and information filtering through dynamic routing at both the global and local levels, effectively reducing unnecessary computations.
- (2) Higher computational efficiency. PSBRA introduces a bi-level routing attention model to replace the traditional multihead self-attention (MHSA). For long sequences and large-scale data, it significantly reduces the computational complexity $O(N^2)$ and memory consumption while maintaining efficient global information capture.
- (3) Better feature fusion and contextual understanding. Through cascading and dynamic routing, the PSBRA can more effectively aggregate multiscale and multilevel feature information, improving the effectiveness of feature fusion and the understanding of global contextual relationships.

YS-CSPLayer module

The feature fusion of YOLO-World is implemented primarily by stacking feature maps of different scales from the channel dimension. During the feature fusion process, the lack of weighting each feature map can lead to insufficient fusion among features. Feature maps of different scales may contain complementary information. The multihead self-attention (MHSA) mechanism is introduced to solve this issue.

First, the input feature maps are processed through a convolutional layer to extract the initial feature and adjust the channel number, which allows the model to obtain richer local information. Second, the feature map is divided into different parts through the convolutional layer. Then, the segmented feature maps enter the dark bottleneck structure, which is used to further compress and process the features, reduce redundancy and extract more useful feature information. The multihead self-attention (MHSA) mechanism dynamically selects the most important feature region on the basis of adaptive attention weights, divides the input features into multiple subspaces and computes different attention weights in parallel, thus flexibly fusing information from multiple-scale feature maps. Then, the processed feature maps are concatenated. The concatenated feature map passes through another convolutional layer for channel integration and feature transformation; finally, the output feature representation is obtained. The fusion procedure is refined through the incorporation of an attention mechanism, facilitating the comprehensive integration of hierarchical features. The resulting fused representation concurrently preserves the fine-grained details from lower-level feature maps and the abstract semantic information from higher-level feature maps. By adaptively modulating the weights across spatial regions and feature channels, the attention mechanism accentuates salient information while effectively mitigating redundancy and suppressing irrelevant interference.

In YOLO-SAM, YS-CSPLayer is positioned at the junction between the detection and segmentation branches, where it fuses multi-scale features from YOLO-World and EfficientSAM. At this stage, the goal is to maximize the precision of cross-branch feature alignment and information exchange. While the PSBRA module is more computationally efficient, its hierarchical routing inevitably filters out some token interactions, which may weaken fine-grained correspondence between modalities. In contrast, multi-head self-attention (MHSA) in YS-CSPLayer preserves complete pairwise interactions across all spatial positions, ensuring that potentially important cross-modal associations are retained. Given that the feature resolution at this fusion stage is relatively low, the computational overhead of MHSA remains acceptable, making it a suitable choice for enhancing fusion accuracy without significantly impacting efficiency.

Method	Backbone	Params	FLOPs	Pretrained Data	AP	APr	APf	APc
MDETR	R-101	169	-	GoldG	16.7	11.2	14.6	19.5
GLIP-T	Swin-T	232	-	O365	24.9	17.7	19.5	31
Grounding-DINO-T	Swin-T	172	-	O365	25.6	14.4	16.9	32.2
DetCLIP-T	Swin-T	155	-	O365	34.4	26.9	33.9	36.3
mm-Grounding-DINO-T	Swin-T	173	-	O365	35.7	28.1	30.2	42
YOLO-World-s	YOLOv8-s	77	297G	O365	26.2	19.1	23.6	29.8
YOLO-SAM(ours)	YOLOv8-s	93	318G	O365	28.3	19.8	25.1	30

Table 1. Comparative experimental Results.

Method	Params (M)	FLOPs (G)	AP	APr	APc	APf
GLIP-T	232	-	21.5	13.8	20.4	26.5
Grounding-DINO-T	172	-	23.4	15	22.8	28
YOLO-World-S	77	297	25.3	17.2	24.8	30.4
YOLO-SAM (ours)	93	318	27.2	20.2	27.8	32.5

Table 2. Zero-shot evaluation on LVIS.

Experiments

The effectiveness of the improved algorithm is verified on the COCO dataset. The COCO dataset is a large-scale dataset that is widely used to train and evaluate computer vision models, especially for tasks such as object detection, segmentation, and image captioning. It contains over 330,000 images (including 220,000 that are annotated), with 1.5 million objects, 80 object categories (such as pedestrians, cars, elephants), and 91 material categories (such as grass, walls, sky). Each image includes five sentence descriptions. Each object is not only annotated with a bounding box but also provided with pixel-level segmentation masks to distinguish different objects in the image, even if they belong to the same category. Moreover, the COCO²⁵ dataset contains complex scenes; for example, it often contains multiple objects and multiple instances of the same category. In this work, we used COCO train2017 to train the model, which includes 118,287 images and their corresponding annotations, and COCO val2017 to evaluate the performance of the model.

The experiments were performed on an Intel(R) Xeon(R) Gold 6230R CPU with 13 cores at 2.10 GHz and an NVIDIA A100 Tensor Core GPU to train and test the module. The experiments were implemented via a partially pretrained YOLO-World model on Python 3.9 and the PyTorch 2.0.0 framework. Pretrained weights were utilized when training the YOLO-SAM model. The initial learning rate was set to 0.01, with a weight decay factor of 0.0005. Reparameterization fine-tuning was employed to integrate text embeddings as parameters of the model during training, which further optimized the model deployment and zero-shot capabilities.

Here, the experimental approach consists of three major stages:

- (1) **Comparative Analysis:** A comprehensive evaluation is conducted by benchmarking the proposed algorithm against several existing object detection approaches, thereby demonstrating its superior performance.
- (2) **Ablation studies:** A series of extensive ablation experiments are carried out by systematically removing individual components of the model and assessing the corresponding performance impact, which rigorously substantiates the contribution of each module to the overall detection efficacy.
- (3) **Qualitative evaluation:** To visually highlight the effectiveness of the enhanced algorithm, representative samples from the dataset are selected for comparative detection analysis, providing an intuitive illustration of its improved detection capabilities.

Comparative experiment

The effectiveness of the YOLO-SAM model is validated on the COCO training 2017 dataset with 200 epochs.

Table 1 shows the results of comparative experiments on the COCO dataset. Compared with models such as MDETR²⁶, GLIP-T²⁷, Grounding-DINO-T²⁸, DetCLIP-T²⁹, mm-Grounding-DINO-T, and YOLO-World-S, YOLO-SAM achieves a 2.1% improvement over YOLO-World-S, an 11.6% improvement over MDETR, a 3.4% improvement over GLIP-T, and a 2.7% improvement over Grounding-DINO-T. Although mm-Grounding-DINO-T has a greater AP value than YOLO-SAM does, YOLO-SAM has a significant advantage in terms of the number of parameters. With fewer parameters than mm-Grounding-DINO-T does, the YOLO-SAM has higher computational efficiency.

The experimental results indicate that YOLO-SAM achieves higher detection accuracy on the COCO dataset than mainstream one-stage models and some vision-language models do, particularly demonstrating superior detection precision in the rare (APr) and frequent (APf) categories. Furthermore, to verify the generalization capability of YOLO-SAM beyond COCO, we also conduct zero-shot evaluations on the LVIS dataset. The LVIS dataset encompasses 1203 long-tail object categories and presents significant challenges in recognizing rare and common categories. Following the evaluation protocol in¹⁹, we report Fixed AP on the LVIS minival set with 1000 predictions per image. Table 2 shows that YOLO-SAM achieves a + 1.9% AP improvement over YOLO-

World-S, with consistent gains across rare, common, and frequent categories, indicating its superior capability in long-tail object detection scenarios.

Ablation experiment

This paper designs comprehensive ablation studies to thoroughly assess the effectiveness of the enhanced modules within the YOLO-SAM algorithm. Using YOLO-World-S as the baseline, the experiments individually validate the contributions of the three key improved modules: LSKA, YS-CSPLayer, and the PSBRA attention mechanism. The experiments are performed on the COCO dataset, and then, systematic evaluations are carried out from multiple dimensions, including detection accuracy, number of parameters, and FLOPs. The experiments use the original YOLO-World-S as the baseline. One model is tested: the effect of only EfficientSAM is tested, and two models are tested: different combinations of the EfficientSAM with the PSBRA, YS-CSPLayer, or LSKA are also tested to verify their performance. Finally, complete model testing is performed, and the YOLO-SAM integrated with all the improved modules is evaluated. The results of these experiments are presented in Table 3.

The results of the ablation studies demonstrate the effectiveness and synergistic effects of the individual improved modules. The integration of EfficientSAM leads to improvements, with mAP@0.5 increasing from 57.0 to 57.3 and mAP@0.5:0.95 increasing from 48.3 to 48.5. Despite the increase in the number of parameters and number of FLOPs, the detection performance of the model improved. Both the PSBRA and YS-CSPLayer methods also yield good performance improvements when they are added individually; in particular, the PSBRA model has a more significant contribution to enhancing accuracy. Although the gain from LSKA is relatively small, it exhibits better synergistic effects when combined with other modules.

The combination of modules shows significant synergistic benefits, especially when the EfficientSAM is combined with PSBRA, YS-CSPLayer, and LSKA, where the mAP@0.5 is increased to 58.8 and the mAP@0.5:0.95 is 49.1. Compared with the baseline model, the YOLO-SAM (full model), which integrates all the modules, substantially outperforms the baseline model: the mAP@0.5 is increased by 1.8%, and the mAP@0.5:0.95 is increased by 0.8%. Moreover, despite the increase in the parameter size from 77 M to 93 M and the increase in the number of FLOPs from 297G to 318G, a favorable balance between computational resources and effectiveness is maintained.

Although the combination of EfficientSAM + PSBRA + LSKA achieves slightly higher mAP values than the full model in certain metrics, the CSPLayer is retained in YOLO-SAM for its critical role in cross-branch multi-scale feature fusion. Positioned at the junction between the detection and segmentation branches, CSPLayer ensures complete pairwise interactions across spatial positions via MHSA, preserving fine-grained feature alignment that is particularly beneficial for small objects and complex boundaries. In repeated experiments, removing CSPLayer led to less stable multi-task optimization and degraded segmentation quality, even when detection accuracy remained competitive. Therefore, the inclusion of CSPLayer reflects a trade-off that prioritizes stability, cross-task consistency, and generalization over marginal gains in individual benchmarks.

Finally, the research confirms the effectiveness of the individual modules. These modules not only yield significant performance gains when they are applied individually but also exhibit stronger complementary effects when they are combined. Ultimately, this leads to a substantial improvement in the detection accuracy and overall performance of the YOLO-SAM on the COCO dataset.

Visual comparative analysis

In complex scenarios involving multiple object interactions and occlusions, the feature extraction capability of YOLO-World-S is limited, leading to significant missed detections and false alarms, as shown in the upper part of Fig. 3. In contrast, the improved YOLO-SAM has enhanced feature extraction capabilities; it accurately detects objects in such scenarios and avoids missed detections with higher detection reliability, as demonstrated in the lower part of Fig. 3.

In addition to detection, Fig. 4 illustrates the instance segmentation performance of YOLO-SAM by comparing the predicted masks with ground truth annotations. The masks generated by YOLO-SAM closely adhere to object boundaries, even in crowded or occluded scenes, confirming its ability to produce fine-grained and accurate segmentations. Furthermore, its open-vocabulary detection capability enables YOLO-SAM to segment additional object categories beyond the annotated labels, underscoring its strong generalization capacity.

Model Configuration	Params (M)	FLOPs (G)	mAP@0.5	mAP@0.5:0.95	IoU(%)	mIoU(%)	FPS
YOLO-World-S	77	297	57	48.3	-	-	373
+EfficientSAM	90	318	57.3	48.5	22.9	75.6	319
+EfficientSAM + PSBRA	91	318	58.1	48.9	24	76.6	315
+EfficientSAM + YS-CSPLayer	91	318	57.7	48.6	23.4	76.1	315
+EfficientSAM + LSKA	90	318	57.6	48.6	23.1	75.6	318
+EfficientSAM + PSBRA + LSKA	92	318	58.3	49	26.5	82.8	312
+EfficientSAM + PSBRA + YS-CSPLayer	93	318	58.5	49	26.2	82.6	310
+EfficientSAM + LSKA + YS-CSPLayer	91	318	58	48.7	24.3	78.1	315
YOLO-SAM(Full Model)	93	318	58.8	49.1	26.2	81.3	308

Table 3. Ablation study results of the YOLO-SAM.



Fig. 3. Comparison of the effects of YOLO-World-S (top) and YOLO-SAM (bottom).

The YOLO-SAM not only has higher reliability of detection objects but also reduces the occurrence of missed detections. In some detection scenarios, YOLO-SAM can more clearly distinguish between different individuals. For example, in the third row, YOLO-SAM shows higher reliability in detecting snowboards and overlapping elephants and avoids false alarms and bounding box fragmentation. To identify smaller objects, YOLO-SAM successfully detects multiple targets that the original model missed.

Compared with traditional methods, the YOLO-SAM not only has excellent real-time performance but also demonstrates superior capabilities in handling complex scenarios. The improved model inherits YOLO-World's remarkable zero-shot learning ability in open scenarios because it is an improvement over YOLO-World. The YOLO-World series models have already been proven to have strong real-time performance and high-precision detection capabilities in object detection tasks, such as an average precision (mAP) of over 50% on the COCO dataset. Moreover, EfficientSAM provides more precise segmentation of object contours, particularly in complex scenes. This paper aims to overcome the shortcomings of traditional segmentation methods (such as Mask R-CNN) in terms of real-time performance and efficiency by combining YOLOWorld and EfficientSAM.

The effectiveness of the proposed YOLO-SAM has been verified through experiments, such as improvements in detection accuracy and speed.

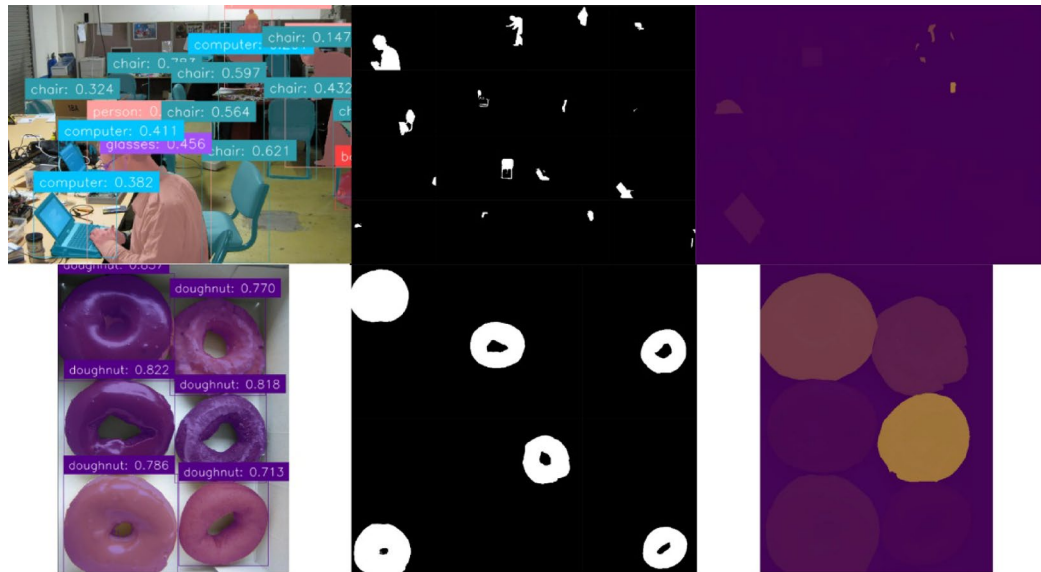


Fig. 4. Comparison of segmentation results of YOLO-SAM: predicted masks with detections (left), predicted masks only (middle), and ground truth annotations (right).

Conclusion

This paper proposes an improved model based on YOLO-World, namely, the YOLO-SAM. The model integrates an EfficientSAM (efficient segmentation auxiliary module), an optimized feature fusion structure, YS-CSPLayer, a separable LSKA (large-kernel convolution attention module), and an efficient postprocessing module, PSBRA. These components effectively improve the object detection performance of the model and its robustness in complex scenarios.

First, YOLO-SAM introduces the EfficientSAM module, which uses segmentation to enhance the detection capabilities and accuracy of detecting small objects and occluded objects. Second, the YS-CSPLayer optimizes the feature fusion structure, enhancing feature representation to make bounding boxes more precise and reduce false positives and false negatives. Moreover, the LSKA module employs a separable large-kernel convolution attention mechanism, making the improved model effectively capture global information while maintaining computational efficiency and improving detection capabilities for different scale objects. Finally, the PSBRA module combines partial self-attention and bi-level routing attention to further optimize object feature extraction and confidence calculations of bounding boxes, thus enhancing the final detection results.

The experimental results on the COCO dataset show that the performance of YOLO-SAM is better than that of YOLO-World-S in terms of both $mAP@0.5$ and $mAP@0.5:0.95$ metrics, and ablation studies validate the effectiveness of each improved module. Specifically, EfficientSAM significantly enhances the edge perception of objects, YS-CSPLayer strengthens feature representation, LSKA maintains robust detection stability in complex scenes, and PSBRA further optimizes the confidence calculation of bounding boxes.

However, the model still has several limitations. On the one hand, the number of parameters and computational complexity have increased, resulting in relatively lower real-time detection efficiency. Therefore, the model is not well suited for deployment on lightweight devices. On the other hand, experiments were restricted on the COCO and LVIS datasets, which contains abundant image-text pairs. The model's generalizability to other datasets or tasks remains to be verified.

Future research directions can focus on optimizing the model for adverse weather conditions such as heavy rain, dense fog, strong glare, and snow, which may degrade image quality and cause target blurring. Additionally, current experiments are primarily based on the COCO and LVIS datasets, but real-world applications may involve significantly different data distributions. Future work can expand the training data by incorporating more diverse datasets, such as those for autonomous driving, drone aerial imagery, and nighttime surveillance, to increase the model's generalizability. With respect to the development, additional strategies like structured pruning of YS-CSPLayer or quantization-aware training of LSKA may further reduce model size and computational cost, making the framework more suitable for mobile or edge applications.

Data availability

The datasets analysed during the current study are available in the COCO repository, [<https://cocodataset.org>], and the LVIS repository [<https://www.lvisdataset.org>].

Received: 20 May 2025; Accepted: 14 October 2025

Published online: 19 November 2025

References

1. Girshick, R. Fast r-cnn, in *Proceedings of the IEEE international conference on computer vision*, 1440–1448 (2015).
2. Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. You only look once: Unified, real-time object detection, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 779–788 (2016).
3. Liu, Y. et al. YOLOv8 Model for Weed Detection, in Wheat Fields Based on a Visual Converter and Multi-Scale Feature Fusion. *Sensors* (14248220), **24**(13). (2024). <https://doi.org/10.3390/s24134379>
4. Ren S, He K, Girshick R, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[J]. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, **39**(6):1137–1149. <https://doi.org/10.1109/TPAMI.2016.2577031> (2017).
5. Zhang H, Li F, Liu S, et al. DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection[J]. *arXiv e-prints*, <https://doi.org/10.48550/arXiv.2203.03605> (2022).
6. Zhu, X. et al. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv* 2020. *arXiv preprint arXiv:2010.04159*, 3, (2010).
7. Cheng, T. et al. Yolo-world: Real-time open-vocabulary object detection, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16901–16911 (2024).
8. Hendry, Chen R C. Automatic License Plate Recognition Via Sliding-Window Darknet-Yolo Deep Learning[J]. *Image and Vision Computing*, **87**, 47–56. <https://doi.org/10.1016/j.imavis.2019.04.007> (2019).
9. Wang, C. Y. et al. CSPNet: A new backbone that can enhance learning capability of CNN. (2019). <https://doi.org/10.48550/arXiv.1911.11929>
10. Ding, X. et al. RepVGG: making VGG-style ConvNets great Again. (2021). <https://doi.org/10.1109/CVPR46437.2021.01352>
11. Xiong, Y. et al. EfficientSAM: leveraged masked image pretraining for efficient segment Anything. (IEEE, 2023). <https://doi.org/10.1109/CVPR52733.2024.01525>
12. Li C, Li L, Jiang H, et al. YOLOv6: A single-stage object detection framework for industrial applications[J]. *arXiv preprint arXiv:2209.02976*, (2022).
13. Wang, C. Y., Bochkovskiy, A. & Liao, H. Y. M. YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv e-prints*, (2022). <https://doi.org/10.48550/arXiv.2207.02696>
14. Glenn Jocher, A., Chaurasia, Jing & Qiu Ultralytics yolov8. (2023). <https://github.com/ultralytics/ultralytics>,
15. Wang, C., Yao, I. H., Yeh & Mark Liao, H. Y. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. *European Conference on Computer Vision* (Springer, Cham, 2025).
16. Radford, A. et al. Learning transferable visual models from natural Language Supervision. (2021). <https://doi.org/10.48550/arXiv.2103.00020>
17. Zareian, A. et al. Open-vocabulary object detection using captions. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 14393–14402. (2021).
18. Gu X, Lin T Y, Kuo W, et al. Open-vocabulary object detection via vision and language knowledge distillation[J]. *arXiv preprint arXiv:2104.13921*, (2021).
19. Kuo W, Cui Y, Gu X, et al. F-vlm: Open-vocabulary object detection upon frozen vision and language models[J]. *arXiv preprint arXiv:2209.15639*, (2022).
20. Wang, M. et al. ESAM-CD: Fine-Tuned EfficientSAM Network with LoRA for Weakly Supervised Remote Sensing Image Change Detection (IEEE Transactions on Geoscience and Remote Sensing, 2024).
21. Zhang, Z., Cai, H. & Han, S. Efficientvit-sam: Accelerated segment anything model without performance loss. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7859–7863. (2024).
22. Chen, Z. et al. Recursive generalization transformer for image Super-Resolution. *ArXiv*, 2023, <https://doi.org/10.48550/arXiv.2303.06373>
23. Yan, L. et al. LS-MambaNet: integrating large strip Convolution and Mamba network for remote sensing object. *Detection Remote Sens.* **17** (10). <https://doi.org/10.3390/rs17101721> (2025).
24. Yan, L. et al. An intelligent weighted object detector for feature extraction to enrich global image Information. *Applied sciences*, **12**(15), 2076–3417 (2022). <https://doi.org/10.3390/app12157825>
25. Lin, T. Y. et al. Microsoft coco: Common objects in context, in *Computer Vision–ECCV : 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V* **13**. 740–755. (Springer, 2014).
26. Kamath, A. et al. Mdetr-modulated detection for end-to-end multi-modal understanding, in *Proceedings of the IEEE/CVF international conference on computer vision*, 1780–1790. (2021).
27. Li, L. H. et al. Grounded language-image pre-training, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10965–10975. (2022).
28. Liu S, Zeng Z, Ren T, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection[C]//European conference on computer vision. Cham: Springer Nature Switzerland, 38–55 (2024).
29. Yao, L. et al. Detclip: Dictionary-enriched visual-concept paralleled pre-training for open-world detection. *Adv. Neural. Inf. Process. Syst.* **35**, 9125–9138 (2022).
30. Peiyuan, Jiang Daji, Ergu Fangyao, Liu Ying, Cai Bo, Ma (2022) A Review of Yolo Algorithm Developments *Procedia Computer Science* 1991066-1073 [10.1016/j.procs.2022.01.135](https://doi.org/10.1016/j.procs.2022.01.135)

Author contributions

Pu XiaHua wrote the entire manuscript. Li XiuMei reviewed the manuscript and provided valuable suggestions for revisions. Ling WenChao and Song XuDong also reviewed the manuscript. All the authors participated in the review process of the manuscript to ensure its quality and integrity.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to X.L.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025