Pattern Recognition 154 (2024) 110579

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

# WBNet: Weakly-supervised salient object detection via scribble and pseudo-background priors

Yi Wang [a], Ruili Wang [b,*], Xiangjian He [c], Chi Lin [a], Tianzhu Wang [b], Qi Jia [a], Xin Fan [a]

[a] School of Software, Dalian University of Technology, Dalian, Liaoning, 116600, China
[b] School of Natural and Computational Sciences, Massey University, Auckland, 0630, New Zealand
[c] School of Computer Science, University of Nottingham Ningbo China, Dalian, Ningbo, 315104, China

ARTICLE INFO

Dataset link: WBNet results (Original data)

*Keywords:*
Weakly supervision
Salient object detection
Neural networks
Transformer
Pseudo labels
Scribble labels

ABSTRACT

Weakly supervised salient object detection (WSOD) methods endeavor to boost sparse labels to get more salient cues in various ways. Among them, an effective approach is using pseudo labels from multiple unsupervised self-learning methods, but inaccurate and inconsistent pseudo labels could ultimately lead to detection performance degradation. To tackle this problem, we develop a new multi-source WSOD framework, WBNet, that can effectively utilize pseudo-background (non-salient region) labels combined with scribble labels to obtain more accurate salient features. We first design a comprehensive salient pseudo-mask generator from multiple self-learning features. Then, we pioneer the exploration of generating salient pseudo-labels via point-prompted and box-prompted Segment Anything Models (SAM). Then, WBNet leverages a pixel-level Feature Aggregation Module (FAM), a mask-level Transformer-decoder (TFD), and an auxiliary Boundary Prediction Module (EPM) with a hybrid loss function to handle complex saliency detection tasks. Comprehensively evaluated with state-of-the-art methods on five widely used datasets, the proposed method significantly improves saliency detection performance. The code and results are publicly available at https://github.com/yiwangtz/WBNet.

## 1. Introduction

Salient object detection (SOD) based deep learning requires pixel-wise dense ground-truth (GT) labels/masks. However, obtaining such labels is expensive, time-consuming, or infeasible in some practical scenarios [1,2]. To alleviate this burden, sparse label-based weakly supervised SOD (WSOD) methods [1,3] have been explored, which can achieve competitive performance with limited annotated data.

Sparse labels refer to a small subset of pixels marked as salient or non-salient regions in an image. Scribbles, bounding boxes, points, categories, and captions are widely used sparse labels for WSOD. Considering scribbles provide accurate salient foreground (salient region) and background (non-salient region) information but have similar annotation costs as other sparse labels [4], we focus on **scribble-based WSOD** in this work.

Scribble labels are comprised of two distinct strokes or lines: one is inside the salient region, and the other is inside the non-salient region. The inherent absence of salient information, such as boundaries and structural details, poses a significant challenge when training a high-performing WSOD model via merely using scribble labels. To address this limitation of scribble labels, an effective approach is using pixel-wise pseudo labels as supplementary cues for the scribbles [4,5]. Some prior studies employ Class Activation Maps (CAMs) [6] to synthesize pseudo labels from image-level category labels [7,8]. Other techniques leverage the rich appearance information available in RGB images to refine CAMs [9]. However, these pseudo labels can often appear fuzzy and imprecise, as illustrated in Fig. 1. In this figure, we color the foreground (salient region) mask red, the background (non-salient region) mask green, and the black area indicates unlabeled regions for scribble labels. For other masks, the gray value represents the probability of being the foreground, and black represents the background.

More recently, unsupervised self-learning models have made significant progress (e.g., DINO [10], SwAV [11], and MoCov2 [12]) for detection and segmentation tasks. The features learned by these models can be utilized for generating pseudo labels for SOD tasks [13]. The pseudo labels generated from self-learning features of different models often show variations, as illustrated in Fig. 1. To mitigate the biases associated with a single model, it can be helpful to leverage this diversity of pseudo labels [7]. However, it is essential to note that these

* Corresponding author.
*E-mail addresses:* dlutwangyi@dlut.edu.cn (Y. Wang), ruili.wang@massey.ac.nz (R. Wang), sean.he@nottingham.edu.cn (X. He), c.lin@dlut.edu.cn (C. Lin), wangtz.nz@gmail.com (T. Wang), jiaqi@dlut.edu.cn (Q. Jia), xin.fan@dlut.edu.cn (X. Fan).

https://doi.org/10.1016/j.patcog.2024.110579
Received 15 September 2023; Received in revised form 3 May 2024; Accepted 9 May 2024
Available online 11 May 2024

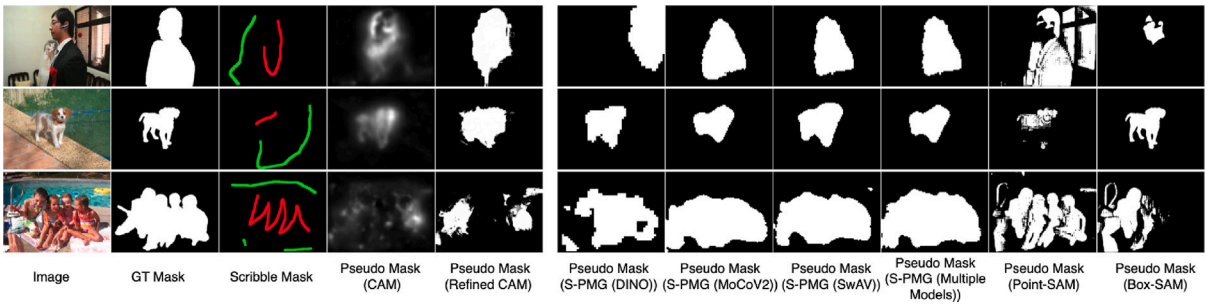



**Fig. 1.** Illustration of various masks of example images from the S-DUTS [3] dataset, including scribble masks, the pseudo masks generated from the CAMs [6] and the refined CAMs [9]. We also display the saliency pseudo masks generated by the proposed S-PMG module solely using a single self-learning model (denoted by S-PMG (DINO), S-PMG (MoCoV2), or S-PMG (SwAV)) and the combination of these three self-learning methods (denoted by S-PMG (Multiple Models)). Additionally, we include pseudo masks generated using the Segment Anything Model (SAM) [14], with prompts from either points (denoted by Point-SAM) derived from scribble labels or bounding boxes from the results of S-PMG (Multiple Models) (denoted by Box-SAM).

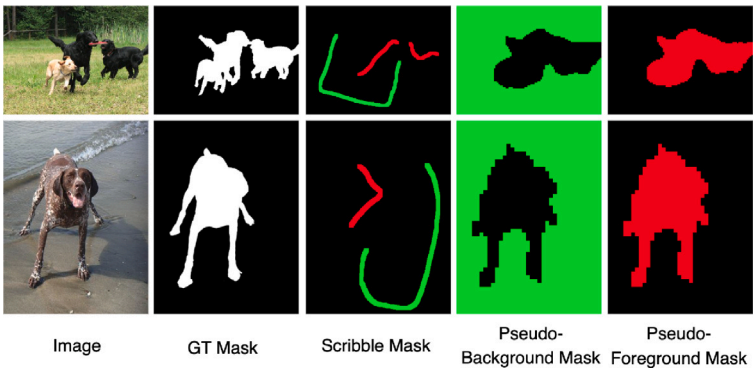



**Fig. 2.** Illustration of the proposed pseudo-background enhancing scribble masks.

pseudo-labels sometimes exhibit inconsistencies. For example, certain areas may be identified as salient by one type of pseudo label but non-salient according to another. Similarly, there may be contradictory issues between pseudo labels and precise sparse labels, potentially impacting the model's overall performance when used together.

This issue drives us to explore how to effectively harness and integrate more comprehensive and robust saliency cues from multiple pseudo labels. We also explore making pseudo labels consistent with scribble labels to improve detection performance. Specifically, we propose a self-learning feature-based pseudo mask generator (S-PMG). This generator employs clustering techniques, such as spectral clustering [15] and $k$-means, to derive candidate masks from various self-learning features from multiple models. Subsequently, we design saliency filtering and selection strategies incorporating various salient constraints to yield more comprehensive pseudo masks compatible with scribble labels.

We also observed that pseudo-background labels are more useful and robust for completing scribble labels than pseudo-full labels. This discrepancy arises because the background region is generally larger than the foreground in most scenarios, sometimes surrounding the foreground [16]. This expansive background area poses a challenge when attempting to manually encompass all relevant background features using lines or strokes alone in the annotation process. Consequently, scribble-background labels may not be effective at facilitating feature learning. As evidenced in Fig. 2, there are instances where important elements from the upper portions of images, such as the dark forest in *Row* 1 and the waves in *Row* 2, are not included in scribble labels. To address this limitation, we propose augmenting scribble-background labels with pseudo-background labels. Pseudo-background labels are advantageous as they encompass a broader spectrum of background content, thereby embedding more comprehensive background features that help distinguish non-salient areas more effectively.

In addition, we explore generating pseudo labels by leveraging the Segment-Anything Model (SAM) [14]. This technique employs salient prompts effectively extracted from the bounding boxes of pseudo foreground masks derived from self-learning methods and taking points as prompts from foreground scribble annotations, as shown in Fig. 1. While this utilization may not strictly adhere to the conventional weakly supervised SOD concept, it is valuable to explore methods that harness pre-trained large models. We consider this exploration an initial foray into this domain, recognizing the potential for future advancements.

Moreover, we propose an innovative Transformer-based weakly supervised SOD network, named WBNet, which can effectively harness pseudo labels and scribble labels in improving performance. Inspired by MaskFormer [17], which was initially designed for the semantic/instance segmentation, the main salient feature learning stream of our framework consists of a Transformer-decoder (TFD) and Pixel-level Feature Aggregation Module (FAM). The Transformer decoder captures global contextual information, while the pixel-level decoder propagates and aggregates features across multiple scales. These two decoders enable the network to capture fine-grained details and a broader global context effectively. In addition to the core salient feature learning, we incorporate a Boundary Prediction Module (EPM) to recover structural information and enhance the representation of boundary details within salient features. Then, we design a comprehensive hybrid loss function to train the network, which evaluates the prediction with multi-source masks and a series of saliency indicators.

Concretely, our contributions are as follows.

- We propose a highly effective weakly supervised salient object detection network named WBNet. A pixel-level Feature Aggregation Module (FAM), a mask-level Transformer-decoder (TFD), and an auxiliary Boundary Prediction Module (EPM) are incorporated into this network to predict saliency information with a

comprehensive hybrid loss function utilizing scribble and pseudo-background labels.
- We design a self-learning feature-based Pseudo-Mask Generator (S-PMG) that utilizes multi-source self-learning features, clustering techniques, and saliency-priors filtering strategies to produce comprehensive pseudo-masks that align consistently with scribble annotations.
- For the first time, we employ the Segment Anything Model (SAM) to generate pseudo-masks for WSOD. We design two effective prompt generation methods: One relies on foreground scribble points, and the other leverages a bounding box derived from pseudo-labels obtained through self-learning.
- We extensively evaluate WBNet on five widely used benchmark datasets with recent SOD and WSOD methods. The results demonstrate that WBNet significantly surpasses other WSOD models and could be compared to some SOD methods in complex scenarios.

The following sections of this article are: Section 2 briefly discusses weakly supervised salient object detection methods; Section 3 explains the proposed WBNet; Section 4 describes and demonstrates the proposed method through quantitative and qualitative experiments; Section 5 summarizes the main points of this article and plans for the future.

## 2. Related work

Weakly supervised SOD (WSOD) methods have become a promising way to address the challenges associated with obtaining pixel-level dense annotations in SOD. The following brief overviews analyze and summarize recent strategies in this field.

### 2.1. Single-source sparse-label based methods

Early works concentrated on identifying influential sparse labels, such as image classification labels, bounding boxes, scribbles, and captions for WSOD. Researchers delved deeply into the strengths and limitations of each type of sparse label.

In 2017, Wang et al. [18] introduced the first WSOD model via image-level labels. A foreground inference network is first trained based on FCNs. Then, the network is fine-tuned with the results in the first stage as GT maps by an iterative Conditional Random Field (CRF) [19]. In 2020, WSSA [3] first utilized scribble annotations in WSOD via an auxiliary boundary prediction task and a gated structure-aware loss. In 2021, SCWSSOD [4] aggregated multi-level features with a saliency structure consistency loss, ensuring consistent saliency maps. PSOD [20] extended the DUTS [18] dataset with point-labels (PDUTS). A transformer-based model is utilized to generate the initial predicted maps. Then, Non-Salient Suppression (NSS) optimized erroneous saliency maps in the second training round. In 2023, Zhao et al. [5] used a Cluster-based Scribble Inference (CSI) and Pooling-based Scribble Inference (PSI) to boost scribble labels, and these boosted labels assisted SCWSSOD to achieve better performance.

Single-source sparse labels, such as scribbles, often provide only coarse-level annotations, lacking precise information about salient objects' exact position and boundaries. Consequently, weakly supervised methods relying on these annotations may suffer from issues like over-segmentation or under-segmentation of salient regions, leading to imprecise saliency predictions. To mitigate the challenges posed by limited supervision, single-source weakly supervised methods sometimes necessitate additional components like a label-boosting network or an auxiliary edge prediction network. While these components enhance model performance, they make the training process more intricate and resource-intensive.

### 2.2. Multi-source sparse-label based methods

Recent advancements focus on combining multiple label sources, leveraging self-supervised learning, and exploring novel loss functions to improve robustness and accuracy.

In 2021, MFNet [7] synthesized both pixel-wise and super-pixel-wise pseudo labels from CAMs based on an image-level classification network. MFNet also uses multiple directive filters to get more accurate predictions from a few noisy pseudo annotations. MWS [21] used category labels and captions to produce pixel-level pseudo labels in one stage and then utilized the synthesized image pairs from Web images to train a SOD network using attention a shift loss and a consistency loss. In 2022, NSAL [8] was developed to guide SOD with pseudo labels obtained from the classification network and a noise-robust discriminator network. Hybrid-SOD [22] incorporated pseudo labels from unsupervised methods and 10% real labels and iteratively trained a coarse label refinement network (R-Net) and a SOD network (S-Net). Li et al. [23] used limited-labeled datasets and unlabeled datasets as training datasets to train a classification network (MFRN) to get boosted labels. In the second phase, these boosted labels supervise a salient region prediction network (SORN) with an edge enhancement branch.

While multi-source label-based methods offer advantages over single-source label-based approaches, they also have weaknesses and limitations. On the one hand, integrating diverse and potentially conflicting annotations from different sources may introduce noise and uncertainty, making the label fusion process more intricate. Besides, ensuring consistency between weak labels from various sources can be difficult, as each source may have biases and inaccuracies. Inconsistent labels can lead to conflicting information during training, potentially hampering the model's learning process. Addressing these weaknesses requires careful consideration and design of the label integration process and the learning strategies.

To tackle the above-mentioned challenges, we present an approach to enhance prediction accuracy by combining multi-source pseudo-background labels with scribble labels. This method has three new features: (i) We harness pseudo-saliency cues generated by multiple self-supervised models within saliency constraints, ensuring their comprehensiveness and alignment with scribble labels. In particular, instead of full-pseudo masks, we propose only to use pseudo-background masks to mitigate the influence of inaccuracies in pseudo-foreground labels on precise sparse labels; (ii) We introduce background saliency cues from large-scale pre-trained model SAM into the framework, which marks the inaugural application of a large-scale model in WSOD; (iii) Our primary saliency detection network comprises a pixel-level decoder, a Transformer decoder, and an edge prediction module. This combination lets us obtain local pixel-level details, global contextual information, and structural and boundary features. Moreover, we designed a comprehensive hybrid loss function for WBNet, encompassing a scribble loss, a disparity smoothness loss, a pseudo-background loss, a local saliency consistency loss, and an edge loss, to assess predictive performance from various perspectives. Experimental results demonstrate that WBNet improves saliency detection performance for WSOD tasks.

## 3. Methodology

The schematic diagram of WBNet, as depicted in Fig. 3, comprises two primary parts: a pseudo mask generation module and a saliency prediction network. Below, we introduce these two components.

### 3.1. Pseudo-mask generation

This section explains how three types of pseudo masks are generated through features learning from self-learning models, box-prompted SAM, and point-prompted SAM. It also explains why the backgrounds of these pseudo labels are used to supplement scribble labels.

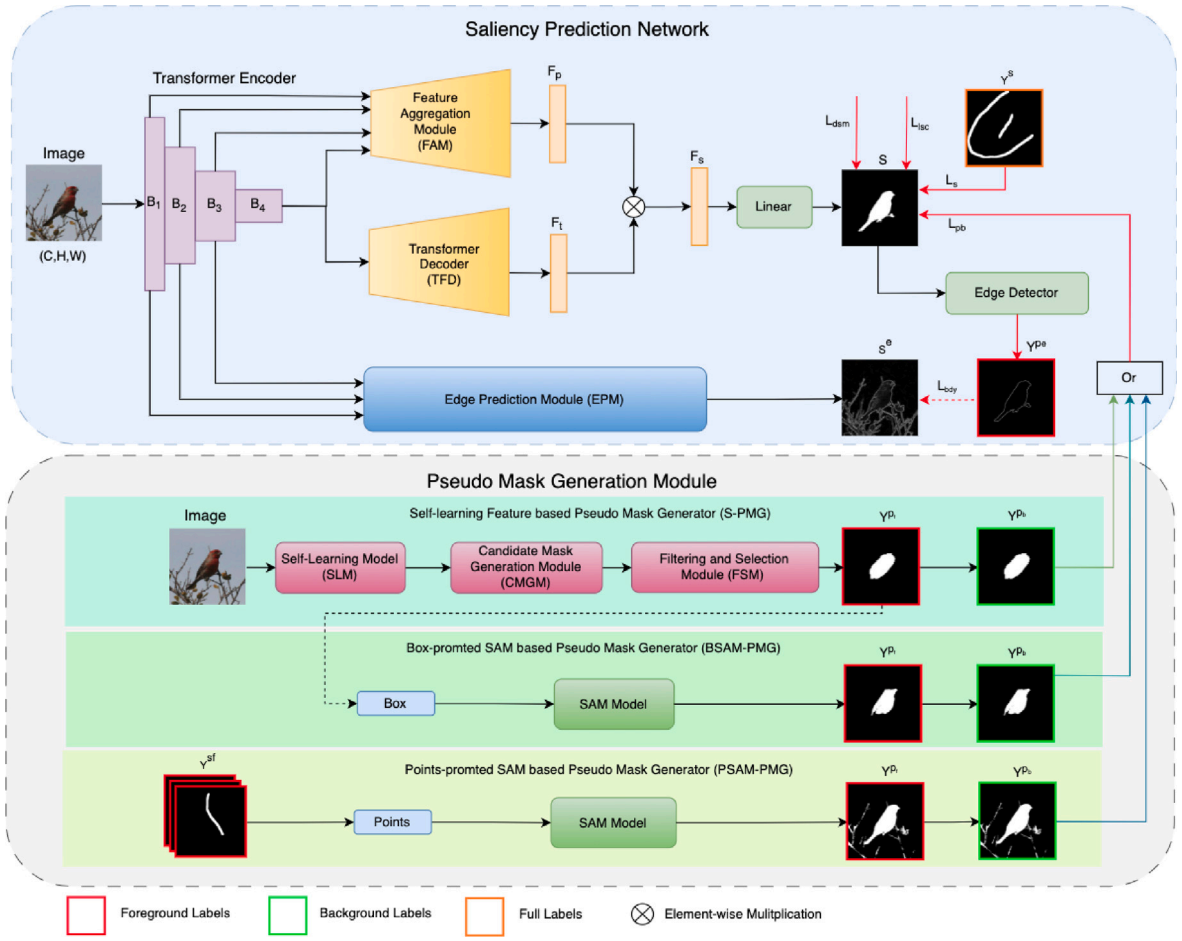



**Fig. 3.** Illustration of the proposed WBNet framework. It consists of a pseudo mask generation module and a saliency prediction network.

*3.1.1. Self-learning pseudo-mask generator (S-PMG)*

Many self-learning models have emerged to mitigate the need for extensive human annotation [1]. However, the absence of ground-truth (GT) labels results in a notable disparity in features generated by different self-supervised learning networks. Drawing inspiration from unsupervised approaches like Self-Mask [13], we leverage three self-learning models (i.e., DINO [10], MoCoV2 [12], and SwAV [11]) to amalgamate the strengths of each.

We first employ several self-learning models to generate image features. Subsequently, we utilize clustering algorithms, such as spectral clustering [15] and $k$-means, to produce multiple candidate masks for each self-learning model. Fig. 4 illustrates the process of generating 9 candidate masks (27 in total from three models) for each model (i.e., DINO, MoCoV2, and SwAV) using different numbers of clusters (i.e., $k = 2, 3, 4$). Following this, all candidate masks are fed into a Filtering and Selection Module (FSM) to output the mask that best conforms to the SOD annotation criteria.

The filtering procedure of the FSM unfolds as follows.

**Step 1: Saliency based Filtering**. Following saliency principles, salient regions are typically situated nearer the center of an image and rarely extend beyond its boundaries [1]. We select masks with the shortest average distance from their constituent pixels to the image's center. Additionally, we discard candidate masks that intersect with the image's width and/or length boundaries.

**Step2: Scribble label based Filtering**. We utilize the foreground (salient region) of scribble labels as the criterion for the saliency filtering process, expecting that the predicted pseudo masks would cover the foreground scribble mask. In this regard, candidate masks that do not completely encompass the foreground scribble mask are excluded from consideration. If none of the candidate masks meet this criterion, all proceed to the next selection process. This approach distinguishes our filtering procedure from Self-Mask, which operates without prior scribble labels and relies solely on the inherent saliency characteristics of the images in its pseudo-label filtering strategy.

**Step 3: Final Selection**. Among the remaining candidate masks, we adopt a selection criterion similar to the approach used in Self-Mask [13] to delete the masks that are excessively elongated and deviate significantly from the center. To be more specific, the mask showing the most excellent average pairwise similarity, identified as *IS*, is selected via the Intersection over Union (IoU) [24] operation, computed as follows.

$$IS = \arg \max_{i \in [1,\ldots,n]} \{ \frac{M_i{}^T \bigcap M_i}{M_i{}^T \bigcup M_i} \}, \tag{1}$$

where $M_i$ represents a candidate mask, it can also be regarded as a matrix. $M_i{}^T$ denotes the transpose of $M_i$. In situations where multiple masks share identical *IS* scores, we randomly select one of them.

From the above description, the structure of S-PMG adopts a modular design, allowing for the incorporation of any number of self-learning models, clustering methods, and filtering strategies, making it easily scalable.

*3.1.2. SAM-based pseudo-label generation*

We utilize SAM [14], the Segment Anything Model, to generate pseudo labels. SAM is a prompt-based segmentation model that works with different prompts, such as points, rectangular boxes, masks, or texts.

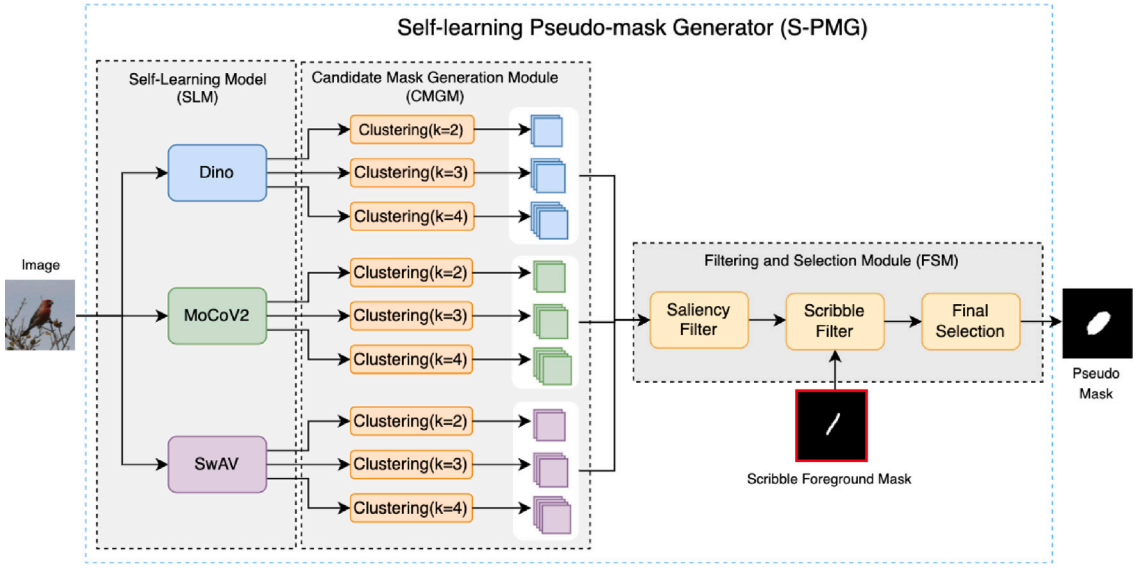



**Fig. 4.** Illustration of the Self-learning Pseudo-mask Generator (S-PMG).

**Point-prompted SAM-based pseudo-mask generator (PSAM-PMG):** We utilize foreground scribble annotations to act as point-prompts to generate pseudo-labels. SAM can simultaneously receive these points to generate a segmentation result. We refer to the WBNet with this type of pseudo label as **WBNet-PSAM**.

**Box-prompted SAM-based pseudo-mask generator (BSAM-PMG):** For the box-prompted generator, we avoid using weakly supervised GT (e.g., scribbles) to directly create boxes of salient regions, as they may not cover the entire object accurately. Instead, we use the full-pseudo mask output from the S-PMG to compute the box prompts. We denote WBNet with this kind of pseudo label as **WBNet-BSAM**.

### 3.2. Full pseudo-labels versus background pseudo-labels

We provide further discussion on pseudo labels below, starting with additional examples generated by the S-PMG (Multiple Models), Box-SAM, and Point-SAM modules as illustrated in Fig. 5. In these examples, the foreground is white, and the background is black. Except for scribble masks, the foreground label is red, and the background label is green.

Obviously, there exists a discrepancy between the pseudo masks (S-PMG masks, Box-SAM masks, and Point-SAM masks) and the pixel-level ground-truth (GT) masks when predicting the foreground region, i.e., the saliency region. Specifically, the S-PMG mask's foreground regions contain more background pixels around boundaries or lack foreground pixels in certain parts. While Point-SAM masks and Box-SAM masks provide more detailed information than S-PMG masks around boundaries, Point-SAM foreground masks exhibit a textured pattern of discrete point collections and include extra background pixels in their foreground regions. On the other hand, Box-SAM masks omit some foreground pixels. Consequently, utilizing such inaccurate foreground pseudo labels to augment accurate scribble foreground labels may introduce errors.

From these examples in Fig. 5, we also observe that background prediction is generally more accurate than foreground prediction. While the primary goal of SOD is the accurate foreground, SOD is a binary classification problem that involves predicting both foreground (salient) and background (non-salient) regions. Therefore, the richness and precision of background information play a crucial role in effectively distinguishing non-salient areas and, consequently, improving the prediction of foreground (salient) regions. Considering the inaccuracies in predicting foreground pseudo-labels, we propose extending scribble background labels with pseudo-background masks. In Section 4.4, we will conduct numerical experiments to demonstrate that pseudo-background labels significantly enhance prediction accuracy compared to pseudo-full labels.

### 3.3. Saliency prediction network

As Fig. 3 shows, the saliency prediction network consists of a Transformer Encoder, a Feature Aggregation Module (FAM), a Transformer Decoder (TFD), an Edge Prediction Module (EPM), and supervisions. Next, we delve into each component in detail.

#### 3.3.1. Encoder

Given an image $I \in \mathcal{R}^{C\times H\times W}$, multi-scale feature blocks with increasing channels ($C$) and decreasing sizes ($H\times W$) are first generated. Considering Transformer backbones (e.g., SwinV2 [25]), produce four-stage of feature blocks; we denote them by $B_i \in \mathcal{R}^{C_i\times H_i\times W_i}(i=1,\ldots,4)$ in our Transformer Encoder.

#### 3.3.2. Decoder

Inspired by MaskFormer [17], initially designed for semantic/instance segmentation tasks, the primary salient feature learning pathway in our network consists of a pixel-level Feature Aggregation Module (FAM) and a Transformer Decoder (TFD). In contrast to MaskFormer, we additionally incorporate an Edge Prediction Module (EPM) to enhance boundary precision in saliency prediction.

**Transformer Decoder (TFD)**: generate per-segment embedding (denoted by $F_t$) by standard from input image features and positional queries similar to MaskFormer [17]. Transformer generates class predictions based on global information collected from all image features. This alleviates the need for the per-pixel module for heavy context aggregation.

**Feature Aggregation Module (FAM)**: In addition to the upsampling and gradually fused operation of the pixel-level decoder in the MaskFormer, we incorporate multi-scale channel attention (MS-CAM) [26] similar to MENet [27] at each intermediate stage of aggregation to enhance salient features' ability to handle various size objects, as depicted in Fig. 6. Subsequently, $F_p$ has the exact resolution as the first feature block (i.e., $B_1$) with $C_p$ channels ($C_p$ is set to 256 in experiments). Then, the salient feature $F_s$ is obtained by matrix multiplication between $F_p$ and $F_t$, followed by Sigmoid activation. We then apply a [1 × 1] convolutional layer and an up-sampling layer, generating the final prediction $S \in \mathcal{R}^{1\times H\times W}$.

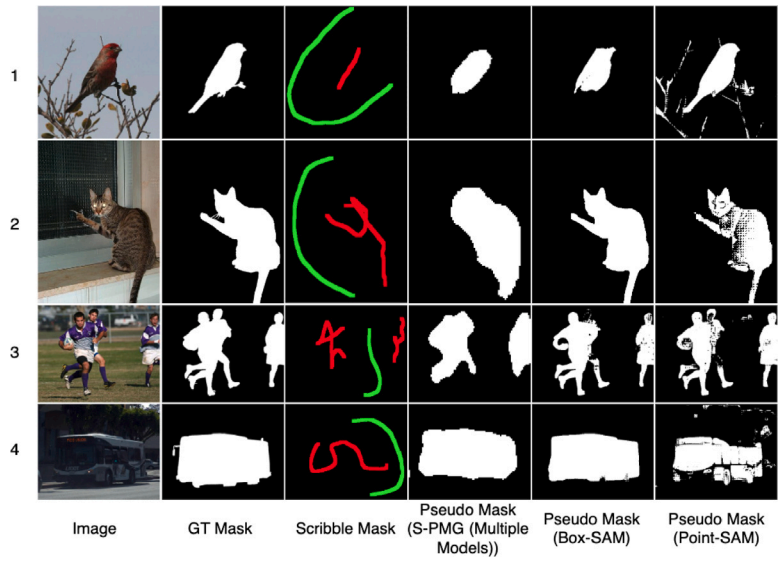



**Fig. 5.** Comparison of pseudo labels generated from S-PMG, Box-SAM, and Point-SAM modules.

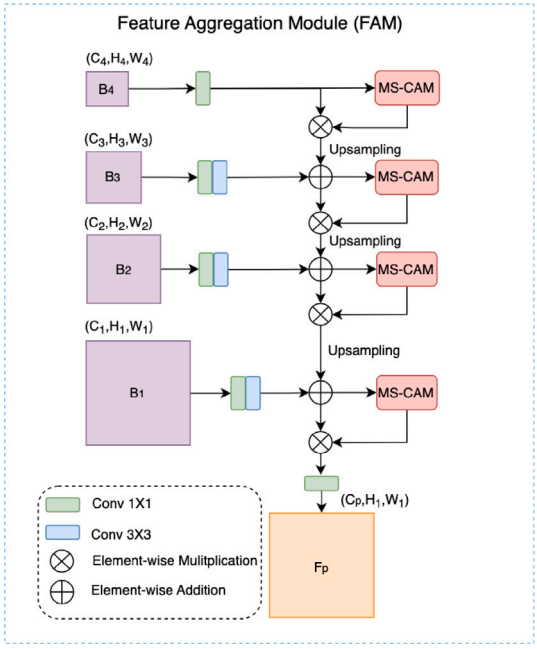



**Fig. 6.** Illustration of the Feature Aggregation Module (FAM).

**Edge Prediction Module (EPM)**: Edge prediction significantly impacts WSOD because it contributes to recovering structural information and enhancing boundary details. Many models [3,20] use edge prediction as an auxiliary task to assist salient feature learning. Our edge prediction module uses $B_1$, $B_2$, and $B_3$ feature blocks as inputs, as illustrated in Fig. 7. We first map each feature block into a channel of $C_e$ (Empirically, $C_e$ is set to be 32) by convolutions and the ReLU activation. We then upsample them into $[H \times W]$ scale and concatenate them to 96 channels. Next, a Residual Channel Attention Block (RCAB) [28] is used to suppress the non-edge information, and a classifier is used to finally produce the edge map $S^e \in \mathcal{R}^{1\times H\times W}$.

Since ground-truth boundary labels are unavailable, we adopt an edge detector to generate pseudo boundary labels, denoted as $Y^{pe}$, derived from the final prediction of the saliency map $S$. Subsequently, Binary Cross-Entropy (BCE) loss [29] is adopted to compute $\mathcal{L}_{bdy}$ by

$$\mathcal{L}_{bdy} = -\sum[Y^{pe} log S^e + (1 - Y^{pe}) log(1 - S^e)], \tag{2}$$

where $S^e$ is the predicted boundary map. To be noted, we detached $\mathcal{L}_{bdy}$ to $Y^{pe}$.

### 3.3.3. Training objective

WBNet uses a hybrid loss ($\mathcal{L}$) in training, as depicted in Fig. 3. $\mathcal{L}$ is composed of a scribble loss $\mathcal{L}_s$, a disparity smoothness loss $\mathcal{L}_{dsm}$, a pseudo-background loss $\mathcal{L}_{pb}$, and a local saliency consistency loss $\mathcal{L}_{lsc}$ for the saliency map $S$, as well as an edge loss $L_{bdy}$ for the BPM branch.

$$\mathcal{L} = \alpha_1 \mathcal{L}_s + \alpha_2 \mathcal{L}_{dsm} + \alpha_3 \mathcal{L}_{pb} + \alpha_4 \mathcal{L}_{lsc} + \alpha_5 L_{bdy}, \tag{3}$$

Empirically, we set $\alpha_1 = 2(H \times W)/(N_{sf} + N_{sb})$, $\alpha_2 = 0.3$, $\alpha_3 = 0.05$, $\alpha_4 = 1$, and $\alpha_5 = 1$. Here, $N_{sf}$ is the pixel number of the foreground of a scribble mask, while $N_{bf}$ is the pixel number of all background labels for scribble masks.

As $\mathcal{L}_{bdy}$ has been introduced in Eq. (2), the following provides details for the other four loss functions.

**Scribble Loss** $\mathcal{L}_s$: Partial cross-entropy loss is adopted for computing $\mathcal{L}_s$ for $S$ and the scribble GT maps.

$$\mathcal{L}_s = -\sum_{i\in PS}[Y_i^s log S_i + (1 - Y^s) log(1 - S_i)], \tag{4}$$

where $PS$ represents scribble labels and $Y^s$ denotes the scribble GT maps.

**Disparity Smoothness Loss** $\mathcal{L}_{dsm}$: We use an edge-aware disparity smoothness penalty to let the salient region be similar to the values in its neighbor with the closest appearance. The disparity smoothness loss is defined by

$$\mathcal{L}_{dsm} = \frac{1}{N}\sum[|\partial_x S| e^{-\|\partial_x I_g\|} + |\partial_y S| e^{-\|\partial_y I_g\|}], \tag{5}$$

where $I_g$ is the gray-scale version of the input image $I$ and $N$ denotes the total pixel count in $S$. The symbols $\partial_x$ and $\partial_y$ represent the partial derivative with respect to $x$ and $y$, respectively.

**Pseudo-background Loss** $\mathcal{L}_{pb}$: this loss consists of partial cross-entropy loss [30] for $S$ and background-pseudo masks $Y^{pb}$. Thus, $\mathcal{L}_{pb}$ is defined by

$$\mathcal{L}_{pb} = -\sum_{i\in PBG}[Y_i^{pb} log S_i + (1 - Y_i^{pb}) log(1 - S_i)], \tag{6}$$

where $PBG$ is the pixel set of the pseudo background masks.

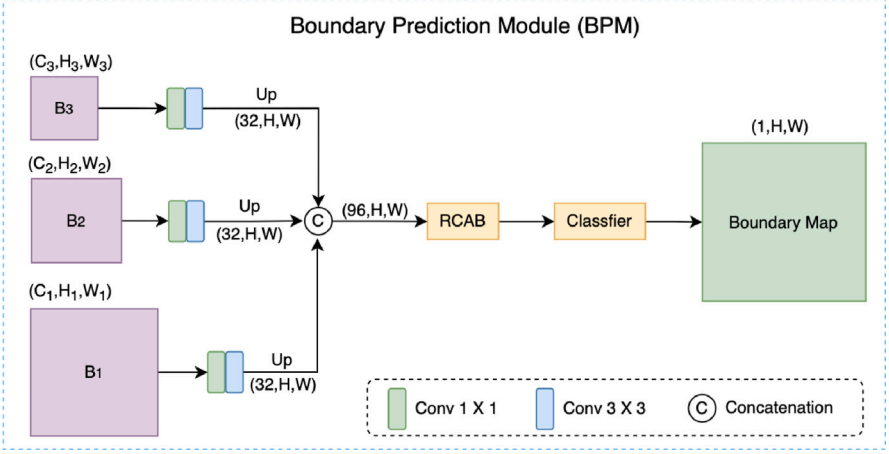



**Fig. 7.** Illustration of the Boundary Prediction Module (BPM).

**Local Saliency Coherence Loss** $\mathcal{L}_{lsc}$: To obtain better precision and enforce boundary pixels having consistent saliency scores, we follow SCWSSOD [4] and use local saliency coherence loss for $S$. Here, we take the original images as the GT maps. The input and the output are resized to a quarter area of the original sizes to make it more efficient. Therefore, $\mathcal{L}_{lsc}$ is defined by

$$\mathcal{L}_{lsc} = \sum_{p_i} \sum_{p_j \in K_i} F(p_i, p_j) D(p_i, p_j), \tag{7}$$

where $K_i$ represents a $[k \times k]$ kernel around pixel $i$; $D(i,j) = |S_i - S_j|$ is the salient difference between pixels $p_i$ and $p_j$ computed by $L_1$ distance; $S_i$ and $S_j$ are salient scores for $p_i$ and $p_j$, respectively; $F(p_i, p_j)$ is a pixel position filter using Gaussian kernels and its definition can be found in [4].

These loss functions work together to enhance the overall comprehensiveness and quality of the predicted outcome: $\mathcal{L}_s$ employs scribble annotations to guide the propagation of scribble pixels into foreground regions, while $\mathcal{L}_{pb}$ enforces background similarity, suppressing foreground expansion; $\mathcal{L}_{dsm}$ ensures the local smoothness; $\mathcal{L}_{lsc}$ guarantees coherence in saliency scores among neighboring pixels, and $\mathcal{L}_{bdy}$ enhances boundary details. Section 4.4 will give an experimental evaluation to assess the impact and effectiveness of these loss settings.

## 4. Experiments and discussion

### 4.1. Training and testing strategies

We test all the models on DUTS-TE [18], DUT-OMRON [31], HKU-IS [32], Pascal-S [33], and ECSSD [34] datasets. We employ SwinV2-Base [25] as the backbone of our models. The backbone's maximum learning rate is 0.0001, while for the other parts, it is 0.001. The momentum and weight decay are 0.9 and 0.0001, respectively. We implement the 'poly' learning rate strategy and scale the input images to [384 × 384]. The training batch size is 10, and we train the models for 99 epochs. All experiments are conducted on a server with an A100 (40G) GPU and an AMD EPYC 7763 64-Core Processor (1T).

### 4.2. Evaluation criteria

The following commonly used metrics [2] are used to evaluate the proposed method comprehensively.

**Mean Absolute Error** (*MAE*) measures the discrepancy between the GT map and the prediction map at the average pixel level and can be expressed as follows:

$$MAE = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{i=1}^{H} \left| G_{ij} - S_{ij} \right|, \tag{8}$$

where $G$ stands for a ground-truth (GT) map; $S$ represents the predicted saliency map; and $W$ and $H$ represent the dimensions of the input image. The smaller the *MAE* value, the better.

**Enhanced Alignment Measure** ($E_m$) uses an enhanced alignment matrix $\phi_s(\ ,\ )$ to capture pixel-level matching and image-level statistics for the predicted salient map foregrounds as follows:

$$E_m = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{i=1}^{H} \phi_s(i,j). \tag{9}$$

We report the average $E_m$ (denoted by $mE_m$) in the experiments.

**Structure-measure** ($S_m$) measures how similar the predicted map is to the ground-truth (GT) map by

$$S_m = \alpha \times S_o + (1 - \alpha) \times S_r, \tag{10}$$

where region-specific and object-specific structural similarities are represented here by $S_o$ and $S_r$, respectively, and $\alpha$ is usually set to 0.5. $S_m$ as a structure-wise measure complements pixel-wise errors.

**F-measure** ($F_\beta$) represents the weighted average of *Precision* and *Recall*, particularly useful for imbalanced datasets and can be mathematically represented by:

$$F_\beta = \frac{(1 + \beta^2) Precision \times Recall}{\beta^2 Precision + Recall}, \tag{11}$$

where $\beta^2$ is usually set to 0.3. We report the maximum F-measure ($F_\beta^{max}$) and mean F-measure ($mF$) in the experiments.

Precision-Recall (PR) curves and F-measure curves are also plotted to demonstrate the overall performance of SOD models. Surveys [1,2] provide detailed descriptions.

### 4.3. Comparison with the state-of-the-arts

In this section, we perform a comprehensive comparative analysis between our proposed models and the state-of-the-art SOD models. The models are classified into three categories: five fully supervised SOD models (i.e., VST [35], EBMG [36], ICON-P [37], ICON-S [37], and SelfReformer [38]), nine weakly supervised SOD models (MWS [39], WSSA [3], SCWSSOD [4], MFNet-D169 [7], MFNet [7], MFRN-SRPN [23], NSAL [8], PSOD [20], and CSI-PSI [5]), and a not strictly WSOD model (i.e., HybridSOD [22]). Here, we clarify: The pseudo-labels used by the WBNet and WBNet-K models, generated by unsupervised self-learning models, fall within the weakly supervised category. In contrast, WBNet-PSAM and WBNet-BSAM utilize pseudo-labels generated by the SAM model, which is supervised and pre-trained. HybridSOD uses 10% full pixel-wise labels in the training dataset. Hence, these models are categorized as the not-strictly weakly-supervised category. We obtain the saliency maps of these models from their papers or deployment codes for fair comparison.

**Table 1**
Quantitative comparisons with state-of-the-art models on the DUT-OMRON [31] and DUTS-TE [18] datasets. The supervision types ('Sup.') are: 'Full' (pixel-wise full labels), 'Cla' (image-level classification labels), 'UPse' (pseudo labels from unsupervised methods), 'Scr' (scribble labels), 'Point' (Point labels), 'Cap' (caption labels), 'CPse' (pseudo labels generated by CAM), 'BUPse' (background pseudo labels from unsupervised methods), and 'BPseSAM' (background pseudo labels from SAM). Symbol '-' means the method does not provide predicted saliency maps to compute certain metrics; The three top results for SOD model and WSOD are shown in red, green, and blue.

| No. | Pub. | Model | Sup. | DUT-OMRON [31] (5168 images) | | | | | DUTS-TE [18] (5019 images) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | $MAE\downarrow$ | $F_\beta^{max}\uparrow$ | $mF_\beta\uparrow$ | $mE_m\uparrow$ | $S_m\uparrow$ | $MAE\downarrow$ | $F_\beta^{max}\uparrow$ | $mF_\beta\uparrow$ | $mE_m\uparrow$ | $S_m\uparrow$ |
| | | SOD | | | | | | | | | | | |
| 1 | $ICCV_{2021}$ | VST [35] | Full | 0.0582 | 0.8245 | 0.7967 | 0.8718 | 0.8503 | 0.0372 | 0.8898 | 0.8579 | 0.9153 | 0.8963 |
| 2 | $NIPS_{2021}$ | EBMG [36] | Full | 0.0505 | 0.8386 | 0.8179 | 0.8951 | 0.8584 | 0.0288 | 0.9091 | 0.8863 | 0.9331 | 0.9088 |
| 3 | $TPAMI_{2023}$ | ICON-P [37] | Full | 0.0468 | 0.8519 | 0.8228 | 0.8951 | 0.8654 | 0.0255 | **0.9218** | 0.8932 | 0.9386 | **0.9173** |
| 4 | $TPAMI_{2023}$ | ICON-S [37] | Full | **0.0426** | **0.8546** | **0.8350** | **0.9073** | **0.8693** | **0.0242** | 0.9196 | **0.8998** | **0.9470** | 0.9171 |
| 5 | $TMM_{2024}$ | SelfReformer [38] | Full | 0.0433 | 0.8367 | 0.8189 | 0.8928 | 0.8608 | 0.0266 | 0.9155 | 0.8921 | 0.9210 | 0.9111 |
| | | WSOD | | | | | | | | | | | |
| 1 | $CVPR_{2019}$ | MWS [21] | Cla+Cap | 0.1077 | 0.7175 | 0.6443 | 0.7642 | 0.7559 | 0.0913 | 0.7671 | 0.7108 | 0.8142 | 0.7590 |
| 2 | $CVPR_{2020}$ | WSSA [3] | Scr | 0.0684 | 0.7532 | 0.7373 | 0.8448 | 0.7849 | 0.0621 | 0.7883 | 0.7723 | 0.8641 | 0.8037 |
| 3 | $AAAI_{2021}$ | SCWSSOD [4] | Scr | 0.0602 | 0.7827 | 0.7779 | 0.8699 | 0.8120 | 0.0488 | 0.8437 | 0.8389 | 0.8967 | 0.8407 |
| 4 | $ICCV_{2021}$ | MFNet-D169 [7] | CPse | 0.0867 | 0.7062 | 0.6845 | 0.8037 | 0.7419 | 0.0761 | 0.7699 | 0.7455 | 0.8373 | 0.7750 |
| 5 | $ICCV_{2021}$ | MFNet [7] | CPse | 0.0982 | 0.6847 | 0.6650 | 0.7844 | 0.7259 | 0.0787 | 0.7625 | 0.7375 | 0.8303 | 0.7781 |
| 6 | $DSP_{2022}$ | MFRN-SRPN [23] | UPse+Cla | 0.0880 | – | 0.6790 | – | 0.7570 | 0.0800 | – | 0.7240 | – | 0.7670 |
| 7 | $AAAI_{2022}$ | PSOD [20] | Point | 0.0642 | 0.8086 | 0.7836 | 0.8648 | 0.8245 | 0.0447 | 0.8578 | 0.8404 | 0.8988 | 0.8536 |
| 8 | $TMM_{2023}$ | NSAL [8] | CPse | 0.0884 | 0.7150 | 0.6918 | 0.8025 | 0.7450 | 0.0728 | 0.7808 | 0.7660 | 0.8431 | 0.7817 |
| 9 | $CAAI\text{-}TIT_{2024}$ | CSI-PSI [5] | Scr | 0.0601 | – | 0.7800 | 0.8650 | – | 0.0500 | – | 0.8330 | 0.9000 | – |
| 10 | $PR_{2024}$ | WBNet-K (Ours) | Scr+BUPse | 0.0486 | 0.8347 | 0.8119 | 0.8917 | 0.8523 | **0.0359** | **0.8789** | 0.8562 | **0.9084** | **0.8774** |
| 11 | $PR_{2024}$ | WBNet (Ours) | Scr+BUPse | **0.0479** | **0.8392** | **0.8187** | **0.8943** | **0.8550** | 0.0374 | 0.8756 | **0.8575** | 0.9083 | 0.8764 |
| | | Not strictly WSOD | | | | | | | | | | | |
| 1 | $TCSVT_{2022}$ | HybridSOD [22] | 10%Full+UPse | – | – | – | – | – | 0.0500 | 0.8030 | – | – | 0.8370 |
| 2 | $PR_{2024}$ | WBNet-BSAM(Ours) | Scr+BPseSAM | **0.0504** | **0.8374** | **0.8085** | 0.8813 | **0.8527** | 0.0391 | 0.8710 | 0.8459 | 0.8951 | **0.8738** |
| 3 | $PR_{2024}$ | WBNet-PSAM(Ours) | Scr+BPseSAM | 0.0524 | 0.8240 | 0.8070 | **0.8856** | 0.8461 | **0.0386** | **0.8730** | **0.8589** | **0.9042** | 0.8731 |

#### 4.3.1. Quantitative evaluation

Tables 1 and 2 show quantitative comparison results for five datasets. Notably, among WSOD models, the proposed WBNet and WBNet-K distinguish themselves by showcasing significantly superior results to both single-source and multi-source WSOD methods, and even comparable to some supervised methods such as EBMG [36] and VST [35]. In particular, WBNet-K, employing $k$-means in the pseudo label computing module (S-PMG), excels on the DUTS-TE and PASCAL datasets, while WBNet, utilizing spatial clustering in S-PMG, demonstrates superior performance on the DUT-OMRON, HKU-IS, and ECSSD datasets. The observed variation in results can be attributed to the diverse distribution characteristics of these datasets. Consequently, the selection of the clustering method in S-PMG may impact the quality of pseudo labels differently depending on the dataset, thereby influencing the overall algorithm accuracy. Nevertheless, WBNet and WBNet-K both surpass other methods by a large margin. Primarily, in terms of *MAE*, 20.29% and 20.32% improvements on the DUT-OMRON; 16.33% and 28.20% improvements on the DUTS-TE, 8.61% and 8.07% improvements on HKU-IS, and 10.05% and 18.44% improvements on the ECSSD are achieved by WBNet and WBNet-K, respectively.

When WBNet-PSAM, WBNet-BSAM, and HybridSOD are compared, it becomes apparent that these three methods, although not strictly WSOD, offer unique insights into label utilization strategies. As the tables indicate, WBNet-PSAM outperforms WBNet-BSAM on the DUTS-TE, HKU-IS, and ECSSD datasets. Furthermore, WBNet-PSAM and WBNet-BSAM are not better than WBNet and WBNet-K overall.

Comparisons of the F-Measure and PR curves of the WSOD models are shown in Figs. 8 and 9. WBNet and WBNet-K's F-measure curves outperform others overall on five datasets. In most cases, WBNet is slightly superior to WBNet-K, consistent with the quantitative results in Tables 1 and 2. Despite PSOD being slightly better than WBNet and WBNet-K after a certain threshold, WBNet and WBNet-K are stable across all five datasets. Based on these observations, WBNet and WBNet-K have more advantages for WSOD tasks.

#### 4.3.2. Qualitative performance comparison

We selected 13 challenging scenes from testing datasets for saliency detection comparison, as shown in Fig. 10. With high integrity and more precise boundaries, the proposed WBNet and WBNet-K achieve the most accurate overall detection results. For example, WBNet demonstrates precise localization of salient objects, avoiding missed detection in some models. In scenarios involving occlusion, WBNet can estimate the obscured portions. In scenes featuring camouflaged objects, WBNet identifies crucial entities, as seen in *Rows* 6 and 7. The boundary delineation is also notably accurate, as evident in the depiction of the monkey's fur in *Row* 5. WBNet performs exceptionally well at predicting geometric objects, particularly those with sharp and elongated features, as shown in *Rows* 12 and 13.

#### 4.3.3. Limitation

From Tables 1 and 2, it is evident that both WBNet and WBNet-K, as weakly supervised models, exhibit certain gaps in accuracy compared to supervised SOD, particularly in terms of *MAE* scores. This difference is also noticeable in Fig. 10, where the segmentation results of our model deviate from the ground-truth (GT) images, especially in cases involving geometrically complex objects in *Rows* 2, 3, 12, and 13.

Moreover, there are some notable instances of larger failures, as depicted in Fig. 11. In these instances, all WSOD models, including our WBNet and WBNet-K, exhibit the common challenge of misclassifying specific non-salient foreground objects as salient. This misclassification arises due to the inherent challenges posed by complex scenes characterized by ambiguity and subjective bias in annotations [1]. It is imperative to emphasize that these challenging instances constitute only a small fraction of the data. However, despite being a minority, they limit the model's ability to handle complex scenes. This indicates areas for improvement in future WSOD models.

### 4.4. Ablation study

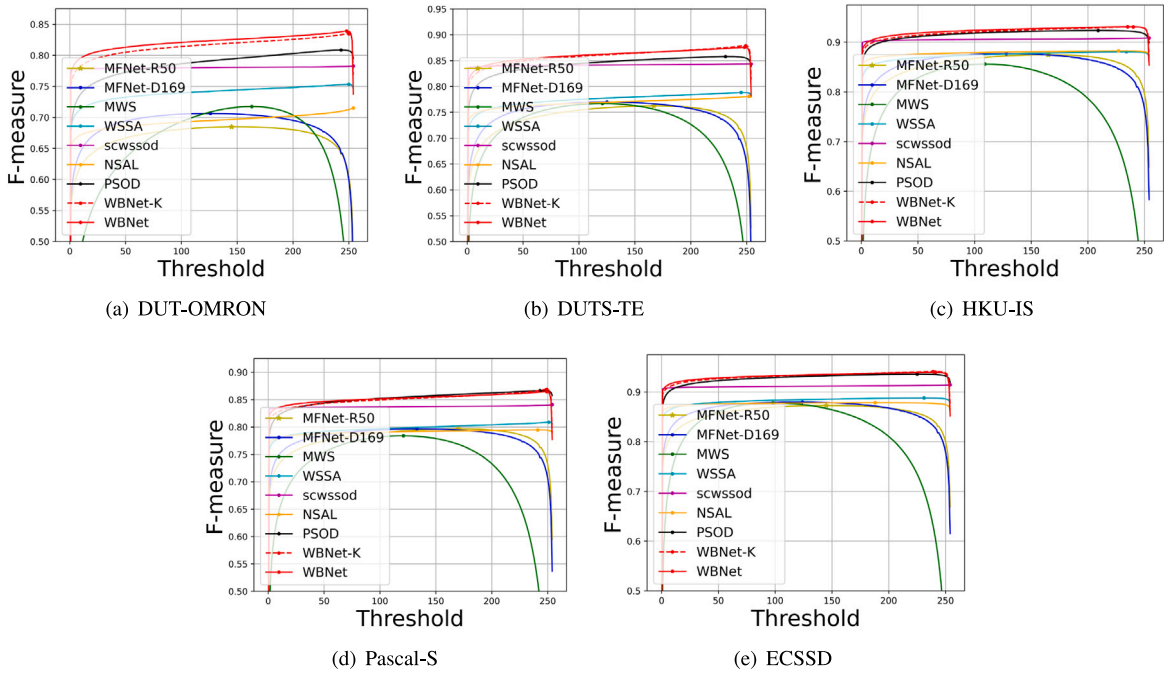
#### 4.4.1. S-PMG configuration

We examined the impact of using different combinations of self-learning models in the S-PMG, with the outcomes presented in Tables 3 and 4. The configuration of the *Row* 5 is the baseline.

Remarkably, DINO achieved superior results when using a single self-learning model, followed by SwAV. SwAV exhibits superior performance on HKU-IS and ECSSD. When combining DINO, MoCoV2, and

**Table 2**
Quantitative comparisons with the state-of-the-art on the HKU-IS [32], PASCAL-S [33], and ECSSD [34] datasets. The supervision types ('Sup.') are: 'Full' (pixel-wise full labels), 'Cla' (image-level classification labels), 'UPse' (pseudo labels from unsupervised methods), 'Scr' (scribble labels), 'Point' (Point labels), 'Cap' (caption labels), 'CPse' (pseudo labels generated by CAM), 'BUPse' (background pseudo labels from unsupervised methods), and 'BPseSAM' (background pseudo labels from SAM). Symbol '–' means the method does not provide predicted saliency maps to compute certain metrics. Each group's top three results are shown in red, green, and blue.

| No. | Pub. | Model | Sup. | HKU-IS [32] (4447 images) | | | | | PASCAL-S [33] (850 images) | | | | | ECSSD [34] (1000 images) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | $MAE\downarrow$ | $F_\beta^{max}\uparrow$ | $mF_\beta\uparrow$ | $mE_m\uparrow$ | $S_m\uparrow$ | $MAE\downarrow$ | $F_\beta^{max}\uparrow$ | $mF_\beta\uparrow$ | $mE_m\uparrow$ | $S_m\uparrow$ | $MAE\downarrow$ | $F_\beta^{max}\uparrow$ | $mF_\beta\uparrow$ | $mE_m\uparrow$ | $S_m\uparrow$ |
| | | SOD | | | | | | | | | | | | | | | | |
| 1 | $ICCV_{2021}$ | VST [35] | Full | 0.0297 | 0.9424 | 0.9129 | 0.9597 | 0.9283 | 0.0620 | 0.8755 | 0.8457 | 0.9024 | 0.8716 | 0.0337 | 0.9507 | 0.9258 | 0.9571 | 0.9323 |
| 2 | $NIPS_{2021}$ | EBMG [36] | Full | 0.0229 | 0.9466 | 0.9288 | 0.9673 | 0.9304 | 0.0542 | 0.8866 | 0.8659 | 0.9070 | 0.8765 | **0.0232** | 0.9591 | 0.9452 | 0.9632 | **0.9416** |
| 2 | $TPAMI_{2023}$ | ICON-P [37] | Full | **0.0216** | **0.9521** | 0.9325 | 0.9698 | 0.9353 | 0.0510 | 0.8927 | 0.8690 | 0.9145 | 0.8819 | 0.0240 | 0.9594 | 0.9432 | 0.9624 | 0.9401 |
| 3 | $TPAMI_{2023}$ | ICON-S [37] | Full | **0.0216** | 0.9512 | **0.9331** | **0.9717** | **0.9355** | **0.0484** | **0.8961** | **0.8767** | **0.9237** | **0.8849** | 0.0235 | **0.9608** | **0.9458** | **0.9669** | 0.9414 |
| 4 | $TMM_{2024}$ | SelfReformer [38] | Full | 0.0241 | 0.9474 | 0.9265 | 0.9606 | 0.9310 | 0.0510 | 0.8943 | 0.8736 | 0.8825 | 0.8809 | 0.0273 | 0.9577 | 0.9414 | 0.9361 | 0.9356 |
| | | WSOD | | | | | | | | | | | | | | | | |
| 1 | $CVPR_{2019}$ | MWS [21] | Cla+Cap | 0.0858 | 0.8560 | 0.7750 | 0.8957 | 0.8183 | 0.1342 | 0.7839 | 0.7136 | 0.7911 | 0.7675 | 0.0985 | 0.8779 | 0.8049 | 0.8849 | 0.8278 |
| 2 | $CVPR_{2020}$ | WSSA [3] | Scr | 0.0470 | 0.8806 | 0.8708 | 0.9322 | 0.8651 | 0.0924 | 0.8088 | 0.7954 | 0.8568 | 0.7975 | 0.0590 | 0.8880 | 0.8803 | 0.9172 | 0.8656 |
| 3 | $AAAI_{2021}$ | SCWSSOD [4] | Scr | 0.0375 | 0.9086 | 0.9031 | 0.9428 | 0.8823 | 0.0775 | 0.8411 | 0.8350 | 0.8806 | 0.8200 | 0.0489 | 0.9145 | 0.9091 | 0.9313 | 0.8820 |
| 4 | $ICCV_{2021}$ | MFNet-D169 [7] | CPse | 0.0585 | 0.8767 | 0.8533 | 0.9222 | 0.8466 | 0.1149 | 0.7967 | 0.7785 | 0.8206 | 0.7695 | 0.0843 | 0.8796 | 0.8600 | 0.8890 | 0.8347 |
| 5 | $ICCV_{2021}$ | MFNet-R50 [7] | CPse | 0.0582 | 0.8747 | 0.8504 | 0.9187 | 0.8525 | 0.1118 | 0.7968 | 0.7770 | 0.8236 | 0.7817 | 0.0841 | 0.8727 | 0.8542 | 0.8894 | 0.8368 |
| 6 | $DSP_{2022}$ | MFRN-SRPN [23] | UPse+Cla | 0.0560 | – | 0.8470 | – | 0.8480 | 0.1090 | – | 0.7730 | – | 0.7790 | 0.0600 | – | 0.8720 | – | 0.0858 |
| 7 | $AAAI_{2022}$ | PSOD [20] | Point | 0.0322 | 0.9235 | 0.9134 | 0.9581 | 0.9022 | 0.0647 | 0.8663 | 0.8499 | **0.8957** | **0.8529** | 0.0358 | 0.9359 | 0.9255 | 0.9526 | 0.9137 |
| 8 | $TMM_{2023}$ | NSAL [8] | CPse | 0.0511 | 0.8825 | 0.8759 | 0.9231 | 0.8540 | 0.1103 | 0.7947 | 0.7885 | 0.8260 | 0.7671 | 0.0777 | 0.8785 | 0.8742 | 0.8893 | 0.8338 |
| 9 | $CAAI\text{-}TIT_{2024}$ | CSI-PSI [5] | Scr | 0.0360 | – | 0.9080 | 0.9440 | – | 0.1200 | – | **0.8650** | 0.8300 | – | 0.0480 | – | 0.9140 | 0.9320 | – |
| 10 | $PR_{2024}$ | WBNet-K (Ours) | Scr+BUPse | 0.0296 | 0.9309 | 0.9177 | 0.9571 | 0.9125 | **0.0638** | **0.8691** | 0.8483 | 0.8714 | 0.8513 | **0.0296** | 0.9309 | 0.9177 | **0.9571** | 0.9125 |
| 11 | $PR_{2024}$ | WBNet (Ours) | Scr+BUPse | **0.0291** | **0.9310** | **0.9203** | **0.9585** | **0.9137** | 0.0658 | 0.8646 | 0.8499 | 0.8723 | 0.8508 | 0.0322 | **0.9398** | **0.9298** | 0.9377 | **0.9189** |
| | | Not strictly WSOD | | | | | | | | | | | | | | | | |
| 1 | $TCSVT_{2022}$ | Hybrid-SOD [22] | 10%Full+UPse | 0.0380 | 0.8920 | – | – | 0.8870 | 0.0760 | 0.8270 | – | – | 0.8280 | 0.0510 | 0.8990 | – | – | 0.8860 |
| 3 | $PR_{2024}$ | WBNet-BSAM (Ours) | Scr+BPseSAM | 0.0295 | 0.9299 | 0.9157 | 0.9546 | **0.9153** | 0.0671 | 0.8614 | 0.8419 | 0.8572 | **0.8495** | 0.0318 | 0.9398 | 0.9270 | 0.9355 | 0.9211 |
| 3 | $PR_{2024}$ | WBNet-PSAM (Ours) | Scr+BPseSAM | **0.0291** | **0.9316** | **0.9220** | **0.9582** | 0.9129 | **0.0665** | **0.8624** | **0.8467** | **0.8685** | 0.8481 | **0.0309** | **0.9412** | **0.9323** | **0.9356** | **0.9203** |




(a) DUT-OMRON (b) DUTS-TE (c) HKU-IS

(d) Pascal-S (e) ECSSD

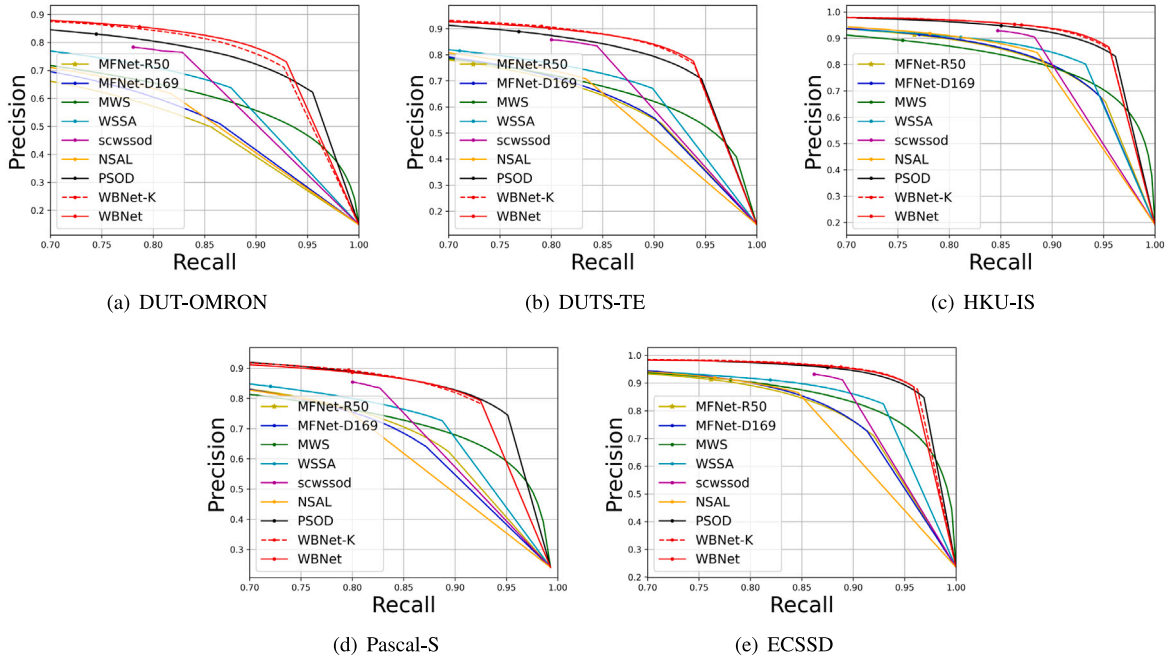
**Fig. 8.** Comparison of the F-measure curves of some state-of-the-art WSOD methods and our WBNet and WBNet-K on five test datasets.

SwAV in S-PMG, the results surpass those attained by employing any single self-learning model, confirming that multi-source pseudo labels can enhance performance. It is noteworthy that, despite DINO-V2 [40] being the advanced version of DINO, it does not outperform DINO on these five *Salient Object Detection* datasets, either used in isolation or mixture with other self-learning models for WBNet.

Fig. 12 illustrates a qualitative comparison of pseudo masks generated using DION, MoCoV2, SwAV, and their combination from the S-PMG module (corresponding to configurations 1, 3, 4, and 5, respectively, in Tables 3 and 4). The masks created using each of the three self-learning methods have their own advantages in various situations. However, the masks' effectiveness is compromised when significant errors occur, such as in *Rows* 4 and 8 for S-PMG (DINO), *Row* 7 for S-PMG (MoCoV2), and *Row* 9 for S-PMG (SwAV) in Fig. 12. Nevertheless, by combining the strengths of multiple self-learning models, the S-PMG (Multiple Models) approach can mitigate these weaknesses and result in more stable pseudo-masks. To provide more examples for comparison, we also show the corresponding pseudo masks from S-PMG (Point-SAM) and S-PMG (Box-SAM) in this figure.

#### 4.4.2. *Pseudo-full labels or pseudo-background labels*

We conduct a comparison between using pseudo-full labels (including both foreground and background masks) and using only pseudo-background labels in WBNet, as presented in *Rows* 5–8 in Tables 3 and 4. The results indicate that the use of pseudo-background labels yields significantly improved performance compared to the use of full pseudo labels. This improvement can be attributed to the nature of scribble labels, which tend to be more accurate at representing the foreground but may lack comprehensive foreground features. When combined with inaccurate pseudo-foreground labels, there is a risk of introducing erroneous foreground information into the pseudo-labels, potentially leading to imprecise feature learning. In contrast, pseudo-background labels offer broader coverage of background features, compensating for any inaccuracies in the scribble background label. Importantly, they




(a) DUT-OMRON (b) DUTS-TE (c) HKU-IS

(d) Pascal-S (e) ECSSD

**Fig. 9.** Comparison of the PR-curves of some state-of-the-art WSOD methods and our WBNet and WBNet-K on five test datasets.

**Table 3**
Comparison of choosing different combinations of unsupervised self-learning methods in the S-PMG module on the DUT-OMRON and DUTS-TE datasets. Symbol 'Bg' means background pseudo labels, and 'Full' means full pseudo labels. The top three results are red, green, and blue.

| No. | Label | Configuration | DUT-OMRON [31] | | | | | DUTS-TE [18] | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | $MAE\downarrow$ | $F_\beta^{max}\uparrow$ | $mF_\beta\uparrow$ | $mE_m\uparrow$ | $S_m\uparrow$ | $MAE\downarrow$ | $F_\beta^{max}\uparrow$ | $mF_\beta\uparrow$ | $mE_m\uparrow$ | $S_m\uparrow$ |
| 1 | Bg | DINO | 0.0483 | 0.8350 | 0.8121 | 0.8878 | 0.8526 | 0.0375 | 0.8737 | 0.8537 | 0.9010 | 0.8750 |
| 2 | Bg | DINO-V2 | 0.0625 | 0.8204 | 0.7875 | 0.8579 | 0.8367 | 0.0441 | 0.8642 | 0.8360 | 0.8832 | 0.8665 |
| 3 | Bg | MoCoV2 | 0.0548 | 0.8140 | 0.7885 | 0.8747 | 0.8375 | 0.0428 | 0.8549 | 0.8295 | 0.8958 | 0.8617 |
| 4 | Bg | SwAV | 0.0568 | 0.8226 | 0.7928 | 0.8740 | 0.8389 | 0.0429 | 0.8656 | 0.8385 | 0.8942 | 0.8654 |
| 5 | Bg | DINO, MoCoV2, SwAV (WBNet) | **0.0479** | **0.8392** | **0.8187** | **0.8943** | **0.8550** | **0.0374** | **0.8756** | **0.8575** | **0.9083** | **0.8764** |
| 6 | Bg | DINO-V2, MoCoV2, SwAV | 0.0545 | 0.8326 | 0.8058 | 0.8839 | 0.8481 | 0.0393 | 0.8740 | 0.8526 | 0.9028 | 0.8735 |
| 7 | Full | DINO, MoCoV2, SwAV | 0.0689 | 0.8292 | 0.7920 | 0.8748 | 0.8354 | 0.0508 | 0.8756 | 0.8364 | 0.8914 | 0.8626 |
| 8 | Full | DINO-V2, MoCoV2, SwAV | 0.0650 | 0.8364 | 0.7993 | 0.8826 | 0.8381 | 0.0493 | 0.8723 | 0.8358 | 0.8994 | 0.8616 |

**Table 4**
Comparison of the performance of choosing different combinations of unsupervised self-learning methods in the S-PMG module on the HKU-IS, PASCAL-S, and ECSSD datasets. Symbol 'Bg' means background pseudo labels, and 'Full' means full pseudo labels. The top three results are red, green, and blue.

| No. | Label | Configuration | HKU-IS [32] | | | | | PASCAL-S [33] | | | | | ECSSD [34] | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | $MAE\downarrow$ | $F_\beta^{max}\uparrow$ | $mF_\beta\uparrow$ | $mE_m\uparrow$ | $S_m\uparrow$ | $MAE\downarrow$ | $F_\beta^{max}\uparrow$ | $mF_\beta\uparrow$ | $mE_m\uparrow$ | $S_m\uparrow$ | $MAE\downarrow$ | $F_\beta^{max}\uparrow$ | $mF_\beta\uparrow$ | $mE_m\uparrow$ | $S_m\uparrow$ |
| 1 | Bg | DINO | 0.0310 | 0.9247 | 0.9106 | 0.9522 | 0.9103 | 0.0759 | 0.8524 | 0.8269 | 0.8351 | 0.8388 | 0.0353 | 0.9354 | 0.9206 | 0.9219 | 0.9149 |
| 2 | Bg | DINO-V2 | 0.0319 | 0.9247 | 0.9073 | 0.9482 | 0.9093 | 0.0721 | 0.8585 | 0.8309 | 0.8338 | 0.8442 | 0.0345 | 0.9344 | 0.9187 | 0.9192 | 0.9168 |
| 3 | Bg | MoCoV2 | 0.0355 | 0.9118 | 0.8973 | 0.9495 | 0.8994 | 0.0689 | 0.8549 | 0.8381 | 0.8657 | 0.8460 | 0.0400 | 0.9246 | 0.9146 | 0.9306 | 0.9057 |
| 4 | Bg | SwAV | 0.0307 | 0.9274 | 0.9112 | 0.9544 | 0.9098 | 0.0689 | 0.8588 | 0.8336 | 0.8542 | 0.8441 | 0.0343 | 0.9366 | 0.9213 | 0.9293 | 0.9159 |
| 5 | Bg | DINO, MoCoV2, SwAV (WBNet) | 0.0291 | **0.9310** | **0.9203** | **0.9585** | 0.9137 | **0.0658** | 0.8646 | **0.8499** | **0.8723** | **0.8508** | 0.0322 | 0.9398 | **0.9298** | **0.9377** | 0.9189 |
| 6 | Bg | DINO-V2, MoCoV2, SwAV | **0.0287** | 0.9308 | 0.9186 | 0.9583 | **0.9142** | 0.0683 | 0.8606 | 0.8414 | 0.8586 | 0.8474 | **0.0310** | **0.9403** | 0.9291 | 0.9363 | **0.9206** |
| 7 | Full | DINO, MoCoV2, SwAV | 0.0361 | 0.9295 | 0.9036 | 0.9528 | 0.9101 | 0.0754 | 0.8637 | 0.8330 | 0.8559 | 0.8434 | 0.0372 | 0.9363 | 0.9166 | 0.9302 | 0.9189 |
| 8 | Full | DINO-V2, MoCoV2, SwAV | 0.0365 | 0.9269 | 0.9003 | 0.9532 | 0.9071 | 0.0739 | **0.8652** | 0.8360 | 0.8690 | 0.8432 | 0.0373 | 0.9364 | 0.9149 | 0.9376 | 0.9167 |

introduce less interference with foreground information, making them a more suitable complement to scribble labels.

### 4.4.3. Network configuration

This experiment aims to elucidate the effects of three key modules of WBNet, including the Feature Aggregation Module (FAM), Transformer Decoder (TFD), and Edge Prediction Module (EPM). The full-module configuration, representing WBNet's default setting, serves as a baseline for comparison. Tables 5 and 6 present experimental results. According to the results, combining all three modules (*Row* 6) yields the most significant performance enhancement. This highlights their synergistic effects on boosting WBNet's overall performance. Using FAM alone (*Row* 1) also demonstrates comparable performance on the HKU-IS, PASCAL, and ECSSD datasets. TFD significantly enhances performance; although using it alone (*Row* 2) is not superior to using FAM independently, combining FAM and TFD (*Row* 4) notably improves performance compared to utilizing either of these two modules individually. Similarly, EPM proves effective, showing substantial improvements in various evaluation metrics when combined with FAM or TFD. Even the combination of FAM and EPM (*Row* 3) achieves the third-best results across all configurations.

**Loss Configuration:** This experiment assesses the influence of different configurations of loss functions. We establish four distinct comparison scenarios by excluding one type of loss from each setting. The full-loss configuration, WBNet's default setting, serves as the baseline for comparison. This comparative analysis is presented in Tables 7 and 8, respectively, across five datasets.

The tables show that excluding specific loss components has different impacts across datasets. For example, $\mathcal{L}_{pb}$ helps reduce *MAE* on the DUT-OMRON, DUTS-TE, and ECSSD datasets, $\mathcal{L}_{bdy}$ helps reduce *MAE*

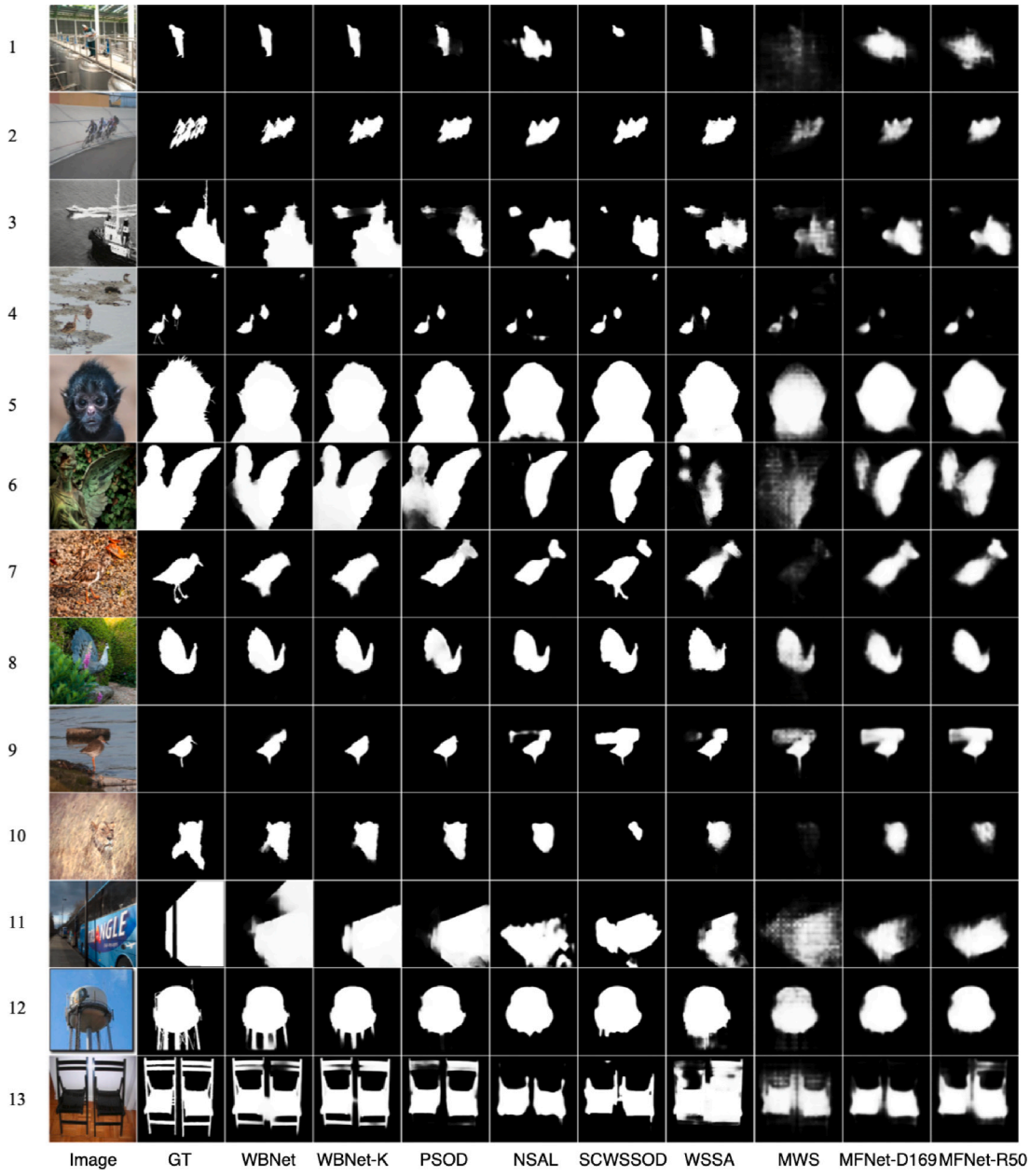



**Fig. 10.** Qualitative performance comparison of the proposed WBNet and WBNet-K with other WSOD methods.

**Table 5**
Configuration comparison of WBNet on the DUT-OMRON and DUTS-TE datasets. The top three results are highlighted in red, green, and blue.

| No. | FAM | TFD | EPM | DUT-OMRON [31] | | | | | DUTS-TE [18] | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | $MAE \downarrow$ | $F_\beta^{max} \uparrow$ | $mF_\beta \uparrow$ | $mE_m \uparrow$ | $S_m \uparrow$ | $MAE \downarrow$ | $F_\beta^{max} \uparrow$ | $mF_\beta \uparrow$ | $mE_m \uparrow$ | $S_m \uparrow$ |
| 1 | √ | – | – | 0.0572 | 0.8291 | 0.8011 | 0.8803 | 0.8452 | 0.0415 | 0.8670 | 0.8438 | 0.8984 | 0.8685 |
| 2 | – | √ | – | 0.0616 | 0.7904 | 0.7741 | 0.8730 | 0.8252 | 0.0496 | 0.8239 | 0.8093 | 0.8944 | 0.8397 |
| 3 | √ | – | √ | 0.0539 | 0.8316 | 0.8034 | 0.8813 | 0.8483 | 0.0405 | 0.8696 | 0.8454 | 0.8987 | 0.8706 |
| 4 | √ | √ | – | 0.0529 | 0.8340 | 0.8092 | 0.8836 | 0.8487 | 0.0385 | 0.8730 | 0.8539 | 0.9027 | 0.8746 |
| 5 | – | √ | √ | 0.0578 | 0.7975 | 0.7813 | 0.8793 | 0.8312 | 0.0478 | 0.8291 | 0.8142 | 0.8970 | 0.8425 |
| 6 | √ | √ | √ | **0.0479** | **0.8392** | **0.8187** | **0.8943** | **0.8550** | **0.0374** | **0.8756** | **0.8575** | **0.9083** | **0.8764** |

on the DUT-OMRON dataset but has less effect on the others, $\mathcal{L}_{lsc}$ helps reduce *MAE* on the DUT-OMRON and HKU-IS datasets significantly but hampers *MAE* on the PASCAL-S dataset, and $\mathcal{L}_{dsm}$ contributes to $S_m$ more than other metrics on five datasets. However, the full-loss

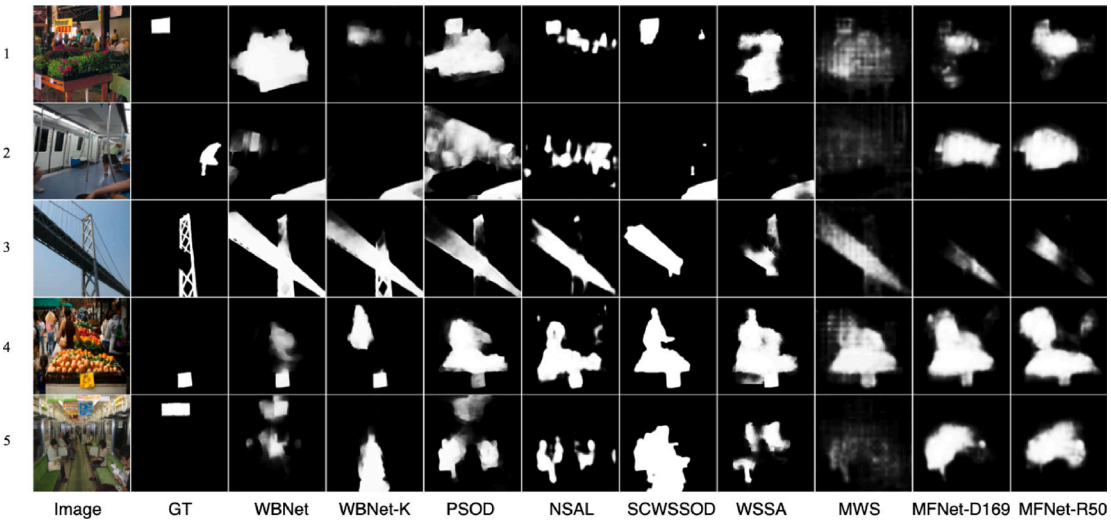



**Fig. 11.** Examples of failure cases.

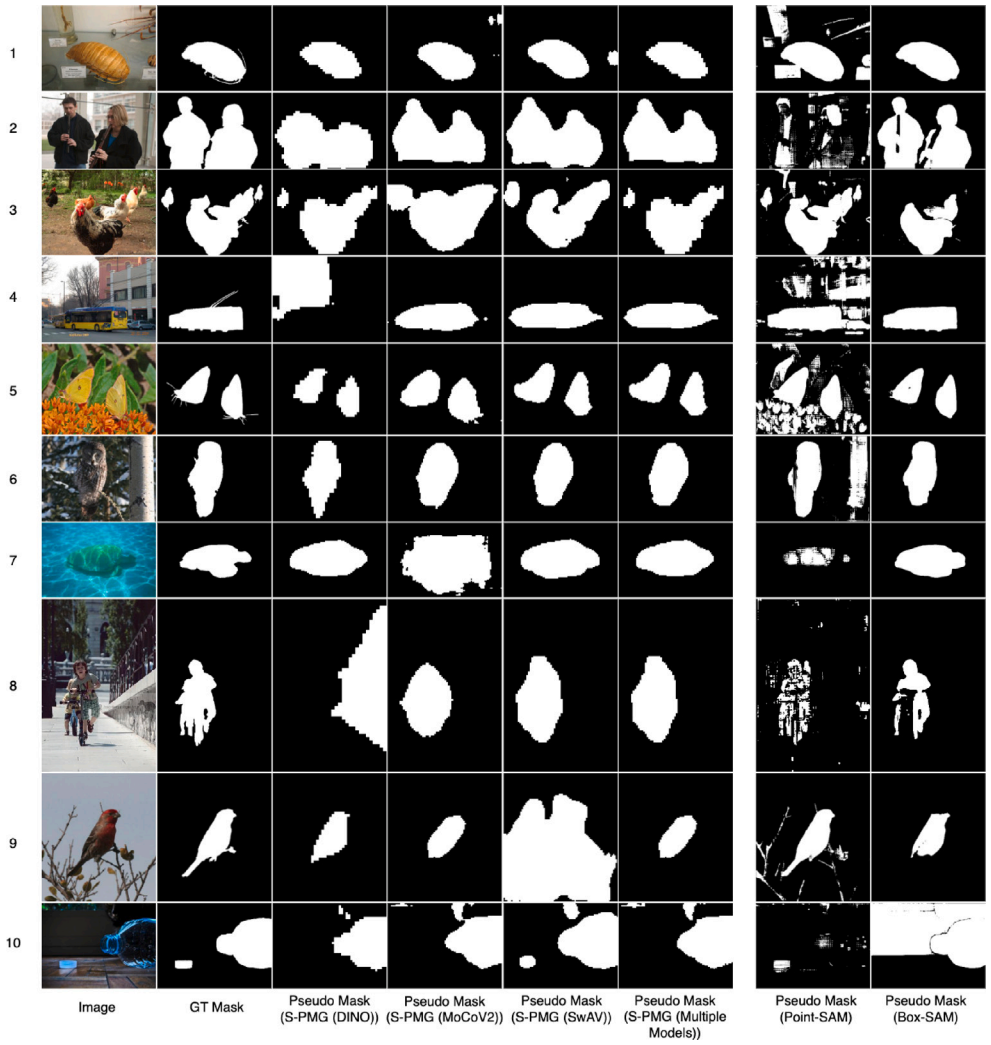



**Fig. 12.** Examples of saliency pseudo masks generated by different configurations of the S-PMG module. The corresponding pseudo masks from Point-SAM and Box-SAM are also listed for comparison.

**Table 6**
Configuration comparison of WBNet on the HKU-IS, PASCAL-S, and ECSSD datasets. The top three results are highlighted in red, green, and blue.

| No. | FAM | TFD | EPM | HKU-IS [32] | | | | | PASCAL-S [33] | | | | | ECSSD [34] | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | $MAE\downarrow$ | $F_\beta^{max}\uparrow$ | $mF_\beta\uparrow$ | $mE_m\uparrow$ | $S_m\uparrow$ | $MAE\downarrow$ | $F_\beta^{max}\uparrow$ | $mF_\beta\uparrow$ | $mE_m\uparrow$ | $S_m\uparrow$ | $MAE\downarrow$ | $F_\beta^{max}\uparrow$ | $mF_\beta\uparrow$ | $mE_m\uparrow$ | $S_m\uparrow$ |
| 1 | √ | – | – | 0.0299 | 0.9269 | 0.9148 | 0.9563 | 0.9109 | 0.0675 | 0.8580 | 0.8393 | 0.8628 | 0.8459 | **0.0321** | 0.9368 | 0.9271 | 0.9374 | **0.9191** |
| 2 | – | √ | – | 0.0397 | 0.8959 | 0.8847 | 0.9484 | 0.8830 | 0.0770 | 0.8347 | 0.8215 | 0.8684 | 0.8257 | 0.0416 | 0.9137 | 0.9038 | 0.9374 | 0.8951 |
| 3 | √ | – | √ | 0.0294 | 0.9278 | 0.9154 | 0.9573 | 0.9124 | 0.0699 | 0.8555 | 0.8365 | 0.8660 | 0.8453 | 0.0322 | 0.9373 | 0.9263 | 0.9362 | 0.9180 |
| 4 | √ | √ | – | **0.0291** | 0.9307 | 0.9197 | 0.9578 | 0.9134 | 0.0663 | **0.8651** | 0.8464 | 0.8668 | 0.8498 | **0.0321** | 0.9391 | 0.9296 | 0.9374 | 0.9185 |
| 5 | – | √ | √ | 0.0399 | 0.8964 | 0.8851 | 0.9471 | 0.8824 | 0.0786 | 0.8366 | 0.8210 | 0.8678 | 0.8239 | 0.0426 | 0.9148 | 0.9036 | 0.9347 | 0.8928 |
| 6 | √ | √ | √ | **0.0291** | **0.9310** | **0.9203** | **0.9585** | **0.9137** | **0.0658** | 0.8646 | **0.8499** | **0.8723** | **0.8508** | 0.0322 | **0.9398** | **0.9298** | **0.9377** | 0.9189 |

**Table 7**
Loss comparison on the DUT-OMRON and DUTS-TE datasets. The best two results are red and green.

| No. | $\mathcal{L}_s$ | $\mathcal{L}_{pb}$ | $\mathcal{L}_{bdy}$ | $\mathcal{L}_{dsm}$ | $\mathcal{L}_{lsc}$ | DUT-OMRON [31] | | | | | DUTS-TE [18] | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | $MAE\downarrow$ | $F_\beta^{max}\uparrow$ | $mF_\beta\uparrow$ | $mE_m\uparrow$ | $S_m\uparrow$ | $MAE\downarrow$ | $F_\beta^{max}\uparrow$ | $mF_\beta\uparrow$ | $mE_m\uparrow$ | $S_m\uparrow$ |
| 1 | √ | – | √ | √ | √ | 0.0532 | 0.8277 | 0.8039 | 0.8873 | 0.8451 | 0.0389 | 0.8682 | 0.8450 | **0.9194** | 0.8700 |
| 2 | √ | √ | – | √ | √ | 0.0529 | 0.8340 | 0.8092 | 0.8836 | 0.8487 | 0.0385 | 0.8730 | 0.8539 | 0.9027 | 0.8746 |
| 3 | √ | √ | √ | – | √ | 0.0515 | 0.8370 | 0.8042 | 0.8828 | 0.8521 | 0.0376 | **0.8773** | 0.8473 | 0.9033 | **0.8786** |
| 4 | √ | √ | √ | √ | – | 0.0539 | 0.8243 | 0.7824 | 0.8700 | 0.8356 | 0.0408 | 0.8612 | 0.8230 | 0.8990 | 0.8608 |
| 5 | √ | √ | √ | √ | √ | **0.0479** | **0.8392** | **0.8187** | **0.8943** | **0.8550** | **0.0374** | 0.8756 | **0.8575** | 0.9083 | 0.8764 |

**Table 8**
Loss comparisons on the HKU-IS, PASCAL-S, and ECSSD datasets. The best two results are red and green.

| No. | $\mathcal{L}_s$ | $\mathcal{L}_{pb}$ | $\mathcal{L}_{bdy}$ | $\mathcal{L}_{dsm}$ | $\mathcal{L}_{lsc}$ | HKU-IS [32] | | | | | PASCAL-S [33] | | | | | ECSSD [34] | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | $MAE\downarrow$ | $F_\beta^{max}\uparrow$ | $mF_\beta\uparrow$ | $mE_m\uparrow$ | $S_m\uparrow$ | $MAE\downarrow$ | $F_\beta^{max}\uparrow$ | $mF_\beta\uparrow$ | $mE_m\uparrow$ | $S_m\uparrow$ | $MAE\downarrow$ | $F_\beta^{max}\uparrow$ | $mF_\beta\uparrow$ | $mE_m\uparrow$ | $S_m\uparrow$ |
| 1 | √ | – | √ | √ | √ | **0.0287** | 0.9272 | 0.9137 | **0.9613** | 0.9111 | 0.0690 | 0.8558 | 0.8407 | 0.8981 | 0.8430 | 0.0308 | 0.9365 | 0.9243 | **0.9619** | 0.9185 |
| 2 | √ | √ | – | √ | √ | 0.0291 | 0.9307 | 0.9197 | 0.9578 | 0.9134 | 0.0663 | 0.8651 | 0.8464 | 0.8668 | 0.8498 | 0.0321 | 0.9391 | 0.9296 | 0.9374 | 0.9185 |
| 3 | √ | √ | √ | – | √ | 0.0288 | 0.9306 | 0.9137 | 0.9573 | **0.9156** | 0.0675 | 0.8598 | 0.8399 | 0.8690 | 0.8498 | **0.0306** | **0.9422** | 0.9277 | 0.9395 | **0.9226** |
| 4 | √ | √ | √ | √ | – | 0.0357 | 0.9283 | 0.9127 | 0.9372 | 0.9093 | **0.0639** | **0.9144** | **0.8910** | **0.9486** | **0.8995** | 0.0357 | 0.9283 | 0.9127 | 0.9372 | 0.9093 |
| 5 | √ | √ | √ | √ | √ | 0.0291 | **0.9310** | **0.9203** | 0.9585 | 0.9137 | 0.0658 | 0.8646 | 0.8499 | 0.8723 | 0.8508 | 0.0322 | 0.9398 | **0.9298** | 0.9377 | 0.9189 |

configuration is versatile enough to address varied databases effectively despite these variations.

## 5. Conclusions

Weakly Supervised Object Detection (WSOD) methods aim to extract more salient information from limited annotations, employing pseudo labels generated by unsupervised self-learning techniques. However, the accuracy and consistency of these pseudo-labels can hinder detection performance.

To address this challenge, this work explores the generation and utilization of multi-source pseudo-labels in WSOD. We first develop a comprehensive salient pseudo-mask generator (S-PMG), utilizing information from diverse self-learning features. We also pioneer the generation of salient pseudo-labels through a point-prompted or box-prompted Segment Anything Model (SAM), which, while not strictly conforming to conventional WSOD paradigms, marks a promising step in this direction.

Furthermore, we develop a Transformer-based WSOD network (WBNet) based on scribbles and the pseudo-background labels from S-PMG. WBNet incorporates a Feature Aggregation Module (FAM), a Transformer-Decoder (TFD), and an auxiliary Edge Prediction Module (EPM) with a multi-source hybrid loss function. Comprehensive evaluations, including comparisons with state-of-the-art WSOD methods on five widely recognized datasets, demonstrate that WBNet substantially improved performance.

However, there still exists a substantial performance gap between weakly-supervised WSOD and fully-supervised SOD at present. Prior knowledge from SAM and self-supervised pre-trained models is undoubtedly beneficial, but it also brings some concerns. One significant concern is the potential bias introduced by these priors, as they are based on the assumptions and patterns learned from standard datasets, which may limit the model's ability to generalize to unseen or specific-task datasets. It is crucial to investigate how to effectively use the general results or features they provide in conjunction with specific tasks. To overcome this challenge, we propose using clustering methods to generate candidate masks from self-learning features and designing saliency rule-based filtering strategies to get pseudo-masks for SOD tasks. In the near future, we plan to explore more advanced and optimized methods, such as multi-granularity adaptive clustering techniques and autonomous customized saliency mask filtering, to enhance the accuracy and robustness of WSOD in diverse and complex scenes.

Additionally, we believe that the use of pseudo-labels is worth exploring, especially when the accuracy of pseudo-labels is limited. For example, our study found that using pseudo-backgrounds as labels yielded better results than using pseudo-foregrounds as labels. Lastly, the choice of prompts when using large models like SAM is a crucial factor significantly influencing overall performance. Therefore, future research must investigate other sparse labels with SAM or other large pre-trained models to enhance detection accuracy.

In conclusion, leveraging existing unsupervised self-learning models or large pre-trained models to assist WSOD is a reliable and versatile approach for future practical tasks in various fields. This direction holds great promise and can lead to significant application breakthroughs.

## CRediT authorship contribution statement

**Yi Wang:** Writing – original draft, Project administration, Methodology, Funding acquisition, Conceptualization. **Ruili Wang:** Writing – review & editing, Supervision, Conceptualization. **Xiangjian He:** Writing – review & editing, Validation, Resources, Methodology, Funding acquisition. **Chi Lin:** Visualization, Funding acquisition, Formal analysis, Data curation. **Tianzhu Wang:** Writing – original draft, Visualization, Validation, Software, Resources, Formal analysis, Data curation. **Qi Jia:** Writing – original draft, Visualization, Validation, Software, Resources. **Xin Fan:** Writing – review & editing, Conceptualization.

## Declaration of competing interest

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

## Data availability

I have shared the link to my data at the attached step

WBNet results (Original data) (Results)

## Acknowledgments

This work was supported by the National Natural Science Foundation of China [grant numbers U22B2052 and 62172069], the Yongjiang Technology Innovation Project, China [grant number 2022A-097-G], and the Ningbo 2025 Key R&D Project, China [grant number 2023Z223].

## References

[1] W. Wang, Q. Lai, H. Fu, J. Shen, H. Ling, R. Yang, Salient object detection in the deep learning era: An in-depth survey, IEEE Trans. Pattern Anal. Mach. Intell. 44 (6) (2021) http://dx.doi.org/10.1109/TPAMI.2021.3051099, 3239–325.

[2] H. Zhou, Y. Lin, L. Yang, J. Lai, X. Xie, Benchmarking deep models on salient object detection, Pattern Recognit. 145 (2024) 109951, http://dx.doi.org/10.1016/j.patcog.2023.109951.

[3] J. Zhang, X. Yu, A. Li, P. Song, B. Liu, Y. Dai, Weakly-supervised salient object detection via scribble annotations, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 12546–12555, http://dx.doi.org/10.1109/CVPR42600.2020.01256.

[4] S. Yu, B. Zhang, J. Xiao, E.G. Lim, Structure-consistent weakly supervised salient object detection with local saliency coherence, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, (4) 2021, pp. 3234–3242, http://dx.doi.org/10.1609/aaai.v35i4.16434.

[5] S. Zhao, P. Cui, J. Shen, H. Liu, Local saliency consistency-based label inference for weakly supervised salient object detection using scribble annotations, CAAI Trans. Intell. Technol. 9 (1) (2024) 239–249, http://dx.doi.org/10.1049/cit2.12210.

[6] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2921–2929, http://dx.doi.org/10.1109/CVPR.2016.319.

[7] Y. Piao, J. Wang, M. Zhang, H. Lu, Mfnet: Multi-filter directive network for weakly supervised salient object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 4136–4145, http://dx.doi.org/10.1109/ICCV48922.2021.00410.

[8] Y. Piao, W. Wu, M. Zhang, Y. Jiang, H. Lu, Noise-sensitive adversarial learning for weakly supervised salient object detection, IEEE Trans. Multimed. 25 (2023) 2888–2897, http://dx.doi.org/10.1109/TMM.2022.3152567.

[9] N. Araslanov, S. Roth, Single-stage semantic segmentation from image labels, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 4252–4261, http://dx.doi.org/10.1109/CVPR42600.2020.00431.

[10] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, A. Joulin, Emerging properties in self-supervised vision transformers, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 9650–9660, http://dx.doi.org/10.1109/ICCV48922.2021.00951.

[11] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, A. Joulin, Unsupervised learning of visual features by contrasting cluster assignments, Advances in Neural Information Processing Systems, vol. 33 (2020) 9912–9924, http://dx.doi.org/10.5555/3495724.3496555.

[12] K. He, H. Fan, Y. Wu, S. Xie, R. Girshick, Momentum contrast for unsupervised visual representation learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 9729–9738, http://dx.doi.org/10.1109/CVPR42600.2020.00975.

[13] G. Shin, S. Albanie, W. Xie, Unsupervised salient object detection with spectral cluster voting, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPRW, 2022, pp. 3970–3979, http://dx.doi.org/10.1109/CVPRW56347.2022.00442.

[14] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A.C. Berg, W.-Y. Lo, P. Dollár, R. Girshick, Segment anything, 2023, http://dx.doi.org/10.48550/arXiv.2304.02643.

[15] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 888–905, http://dx.doi.org/10.1109/34.868688.

[16] Z. Chen, R. Wang, Z. Zhang, H. Wang, L. Xu, Background–foreground interaction for moving object detection in dynamic scenes, Inform. Sci. 483 (2019) 65–81.

[17] B. Cheng, A. Schwing, A. Kirillov, Per-pixel classification is not all you need for semantic segmentation, in: M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J.W. Vaughan (Eds.), in: Advances in Neural Information Processing Systems, vol. 34, 2021, pp. 17864–17875.

[18] L. Wang, H. Lu, Y. Wang, M. Feng, D. Wang, B. Yin, X. Ruan, Learning to detect salient objects with image-level supervision, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Honolulu, HI, USA, 2017, pp. 3796–3805, http://dx.doi.org/10.1109/CVPR.2017.404.

[19] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, IEEE Trans. Pattern Anal. Mach. Intell. 40 (4) (2017) 834–848.

[20] S. Gao, W. Zhang, Y. Wang, Q. Guo, C. Zhang, Y. He, W. Zhang, Weakly-supervised salient object detection using point supervison, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, (1) 2022, pp. 670–678.

[21] Y. Zeng, Y. Zhuge, H. Lu, L. Zhang, M. Qian, Y. Yu, Multi-source weak supervision for saliency detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 6074–6083, http://dx.doi.org/10.1109/CVPR.2019.00623.

[22] R. Cong, Q. Qin, C. Zhang, Q. Jiang, S. Wang, Y. Zhao, S. Kwong, A weakly supervised learning framework for salient object detection via hybrid labels, IEEE Trans. Circuits Syst. Video Technol. 33 (2) (2022) 534–548, http://dx.doi.org/10.1109/TCSVT.2022.3205182.

[23] X. Li, Y. Xu, L. Ma, Z. Yang, Z. Huang, H. Hong, J. Tian, Multi-source weakly supervised salient object detection via boosting weak-annotation source and constraining object structure, Digit. Signal Process. 126 (2022) 103461, http://dx.doi.org/10.1016/j.dsp.2022.103461.

[24] M.A. Rahman, Y. Wang, Optimizing intersection-over-union in deep neural networks for image segmentation, in: Advances in Visual Computing, Springer, 2016, pp. 234–244, http://dx.doi.org/10.1007/978-3-319-50835-1_22.

[25] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong, et al., Swin transformer v2: Scaling up capacity and resolution, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognitionn, 2022, pp. 12009–12019, http://dx.doi.org/10.1109/CVPR52688.2022.01170.

[26] Y. Dai, F. Gieseke, S. Oehmcke, Y. Wu, K. Barnard, Attentional feature fusion, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 3560–3569, http://dx.doi.org/10.1109/WACV48630.2021.00360.

[27] Y. Wang, R. Wang, X. Fan, T. Wang, X. He, Pixels, regions, and objects: Multiple enhancement for salient object detection, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2023.

[28] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, Y. Fu, Image super-resolution using very deep residual channel attention networks, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 286–301.

[29] P.-T. De Boer, D.P. Kroese, S. Mannor, R.Y. Rubinstein, A tutorial on the cross-entropy method, Ann. Oper. Res. 134 (1) (2005) 19–67.

[30] M. Tang, A. Djelouah, F. Perazzi, Y. Boykov, C. Schroers, Normalized cut loss for weakly-supervised cnn segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 1818–1827, http://dx.doi.org/10.1109/CVPR.2018.00195.

[31] C. Yang, L. Zhang, H. Lu, X. Ruan, M.-H. Yang, Saliency detection via graph-based manifold ranking, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Portland, OR, USA, 2013, pp. 3166–3173, http://dx.doi.org/10.1109/CVPR.2013.407.

[32] G. Li, Y. Yu, Visual saliency based on multiscale deep features, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Boston, MA, USA, 2015, pp. 5455–5463, http://dx.doi.org/10.1109/CVPR.2015.7299184.

[33] Y. Li, X. Hou, C. Koch, J.M. Rehg, A.L. Yuille, The secrets of salient object segmentation, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Columbus, OH, USA, 2014, pp. 280–287, http://dx.doi.org/10.1109/CVPR.2014.43.

[34] Q. Yan, L. Xu, J. Shi, J. Jia, Hierarchical saliency detection, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Portland, OR, USA, 2013, pp. 1155–1162, http://dx.doi.org/10.1109/CVPR.2013.153.

[35] N. Liu, N. Zhang, K. Wan, L. Shao, J. Han, Visual saliency transformer, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 4722–4732, http://dx.doi.org/10.1109/ICCV48922.2021.00468.

[36] J. Zhang, J. Xie, N. Barnes, P. Li, Learning generative vision transformer with energy-based latent space for saliency prediction, in: 2021 Conference on Neural Information Processing Systems, 2021.

[37] M. Zhuge, D.-P. Fan, N. Liu, D. Zhang, D. Xu, L. Shao, Salient object detection via integrity learning, IEEE Trans. Pattern Anal. Mach. Intell. 45 (3) (2023) 3738–3772, http://dx.doi.org/10.1109/tpami.2022.3179526.

[38] Y.K. Yun, W. Lin, Towards a complete and detail-preserved salient object detection, IEEE Trans. Multimed. 26 (2024) 4667–4680, http://dx.doi.org/10.1109/TMM.2023.3325731.

[39] H. Zhang, Y. Zeng, H. Lu, L. Zhang, J. Li, J. Qi, Learning to detect salient object with multi-source weak supervision, IEEE Trans. Pattern Anal. Mach. Intell. 44 (7) (2021) 3577–3589, http://dx.doi.org/10.1109/TPAMI.2021.3059783.

[40] Y. Wei, S. Ji, Scribble-based weakly supervised deep learning for road surface extraction from remote sensing images, IEEE Trans. Geosci. Remote Sens. 60 (2022) 1–12, http://dx.doi.org/10.1109/TGRS.2021.3061213.

**Yi Wang** received B.E. and Ph.D. degrees in computer science and technology from Jilin University, China, in 2002 and 2009, respectively. Since 2009, she has been with the Dalian University of Technology, China. She is currently an Associate Professor. Her research interests include machine learning, image processing, and computer vision.

**Ruili Wang** is currently Professor of Artificial Intelligence and Chair of Research in the School of Mathematical and Computational Sciences, at Massey University, New

Zealand. His current research areas include machine learning, speech processing, image processing, language processing, video processing, and computer vision.

**Xiangjian He** is currently a Professor in Computer Science and leads the Computer Vision and Intelligent Perception Lab at the University of Nottingham Ningbo, China. His research interests include image processing, pattern recognition, and computer vision.

**Chi Lin** received B.S. and Ph.D. degrees from Dalian University of Technology, China, in 2008 and 2013, respectively. Since 2017, he has been an Associate Professor at Dalian University of Technology. His research interests include machine learning, pervasive computing, and wireless sensor networks.

**Tianzhu Wang** received a Ph.D. degree from Jilin University, China. Since 2016, he has been a freelance researcher in machine learning and is currently a casual researcher at Massey University, New Zealand. His research interests include pattern recognition and computer vision.

**Qi Jia** received a Ph.D. degree in computer science and technology from Dalian University of Technology, China. Since 2008, she has been at the Dalian University of Technology. She is currently an Associate Professor. His research interests include pattern recognition and computer vision.

**Xin Fan** received B.E. and Ph.D. degrees from Xian Jiaotong University, Xian, China, in 1998 and 2004, respectively. He was at Oklahoma State University, Stillwater, and the University of Texas Southwestern Medical Center, Dallas, from 2006 to 2009, as a post-doctoral research fellow. He joined Dalian University of Technology, Dalian, China, in 2009, where he is currently a full professor. His current research interests include image processing and machine vision.