A Contrastive-Learning Framework for Unsupervised Salient Object Detection

Huankang Guan, Jiaying Lin, and Rynson W.H. Lau

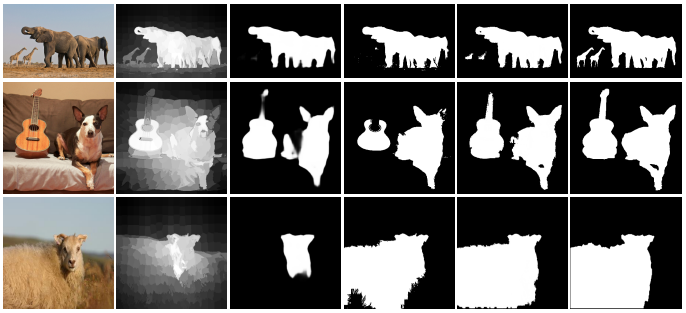
***Abstract*—Existing unsupervised salient object detection (USOD) methods usually rely on low-level saliency priors, such as center and background priors, to detect salient objects, resulting in insufficient high-level semantic understanding. These low-level priors can be fragile and lead to failure when the natural images do not satisfy the prior assumptions, *e.g.*, these methods may fail to detect those off-center salient objects causing fragmented objects in the segmentation. To address these problems, we propose to eliminate the dependency on flimsy low-level priors, and extract high-level saliency from natural images through a contrastive learning framework. To this end, we propose a Contrastive Saliency Network (CSNet), which is a prior-free and label-free saliency detector, with two novel modules: i) a Contrastive Saliency Extraction (CSE) module to extract high-level saliency cues, by mimicking the human attention mechanism within an instance discriminative task through a contrastive learning framework, and ii) a Feature Re-Coordinate (FRC) module to recover spatial details, by calibrating high-level features with low-level features in an unsupervised fashion. In addition, we introduce a novel local appearance triplet (LAT) loss to assist the training process by encouraging similar saliency scores for regions with homogeneous appearances. Extensive experiments show that our approach is effective and outperforms state-of-the-art methods on popular SOD benchmarks.**

***Index Terms*—Salient object detection, contrastive learning, unsupervised learning.**

I. Introduction

SALIENT object detection (SOD) is a fundamental computer vision task that aims to identify the most visually conspicuous objects in a scene. It has contributed to various tasks, including object detection [4], semantic segmentation [5]–[8], and image captioning [9]. However, the performance of state-of-the-art SOD methods [10]–[22] is hindered by the high cost in annotating large-scale, clean datasets for supervised training. It is therefore important to develop unsupervised salient object detection (USOD) approaches to eliminate the high manual labeling cost.

Earlier USOD methods [1], [23]–[27] heavily rely on low-level saliency priors, such as center prior [1], [24], [25], boundary prior [1], [23], and background prior [26], [27]. These traditional methods suffer from insufficient high-level semantic understanding, leading to fragmented objects in scenarios with significant appearance differences within the salient objects themselves. In addition, these methods may fail if the input images do not satisfy the prior assumptions. For example, center-prior-based methods [1], [24], [25] may not perform well if the salient objects are near to the image borders.



(a) Input **(b)** RBD [1] **(c)** UMNet [2] **(d)** Found [3] **(e)** Ours **(f)** GT

Fig. 1. Hand-crafted methods, *e.g.*, (b) RBD [1], usually lose sight of salient objects close to the border (the giraffes in the 1^{st} row) and output fragmented objects (the dog in the 2^{nd} row and the goat in the 3^{rd} row). Such limitations are unavoidably passed to recent deep-learning-based USOD methods, *e.g.*, (c) UMNet [2], which are trained with pseudo labels from hand-crafted techniques. Recent self-supervised clustering-based methods, *e.g.*, (d) Found [3], tend to suffer from under-detection due to the semantic/appearance gaps among visual tokens. In contrast, our method in (e) can consistently detect the salient objects from the background by excavating high-level saliency in a prior-free contrastive learning framework.

Recently, a number of works [2], [28]–[33] have attempted to train deep neural networks with noisy labels produced by traditional methods, *e.g.*, [1], [26], [27]. Despite the success, these deep models inevitably inherit the limitations of traditional hand-crafted techniques, such as the reliance on low-level prior assumptions and the failure to segment complete objects. For example, as shown in Figure 1(c), UMNet [2], which is trained with noisy labels from hand-crafted methods, fails to detect the off-center giraffes in the first row, and cannot accurately segment the dog in the second row and the goat in the third row. Another line of works [3], [34]–[36] propose clustering strategies to discover salient objects using self-supervised models [37]–[39]. While this approach may be able to highlight foreground objects in natural images, a recent self-supervised clustering-based method, Found [3], tends to suffer from under-detection, especially when there are large semantic/appearance gaps among salient objects/parts, as shown in Figure 1(d). This is because clustering-based methods hinge on the similarity among salient tokens to determine salient regions.

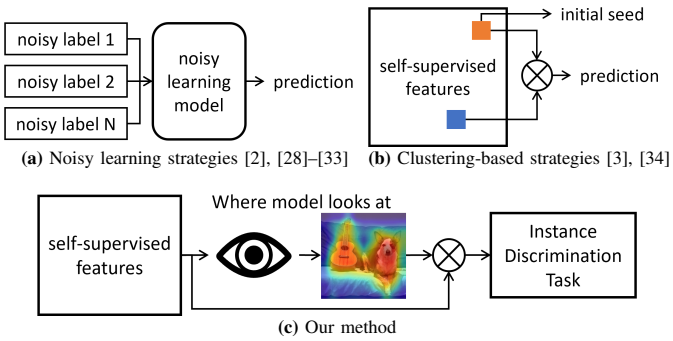
In this work, instead of learning from noisy labels [2], [28]–[33] or focusing on self-supervised feature clustering [3], [34]–[36], we propose to excavate high-level saliency cues by mimicking human attention mechanisms in an instance

Huankang Guan is with the Department of Computer Science, City University of Hong Kong (Huankang.Guan@my.cityu.edu.hk).

Jiaying Lin is with the Department of Computer Science, City University of Hong Kong (csjylin@gmail.com).

Rynson W.H. Lau is the corresponding author. He is with the Department of Computer Science, City University of Hong Kong (Rynson.Lau@cityu.edu.hk).

Manuscript received mm-dd-yyyy; revised mm-dd-yyyy.



(a) Noisy learning strategies [2], [28]–[33] **(b)** Clustering-based strategies [3], [34]

(c) Our method

Fig. 2. Comparison of (a) noisy learning strategies, (b) clustering-based strategies, and (c) our approach.

discriminative task through a contrastive learning framework. We compare our approach with the two types of strategies in Figure 2. Specifically, we propose Contrastive Saliency Network (CSNet), which is a prior-free and label-free salient object detection method. CSNet contains two novel modules to achieve the goal: 1) a Contrastive Saliency Extraction (CSE) module, and 2) a Feature Re-Coordinate (FRC) module. Unlike previous USOD methods [1], [35], our CSE module does not make any prior assumptions concerning the distribution or frequency of salient objects in the scene. Instead, we use a global saliency query to identify potential salient regions, and optimize the CSE module to perform an instance discriminative task based on the contrastive loss [40]. We also refine the saliency maps extracted by the CSE module using the proposed FRC module, which aims to align high-level features with low-level features to facilitate fine-grained saliency map output. Finally, we propose a novel local appearance triplet (LAT) loss to encourage local regions with homogeneous appearances to have similar saliency scores during training. We conduct extensive experiments to demonstrate the effectiveness of our approach, and our model produces new state-of-the-art results on the existing SOD benchmarks.

In summary, our contributions are three-fold:

1) We propose to extract high-level saliency from input images in an instance discriminative task, while eliminating dependencies on low-level prior assumptions or any kind of labels through a contrastive learning framework.
2) We propose a Constrative Saliency Network (CSNet) with two novel modules, the CSE module and the FRC module, for unsupervised salient object detection. We also propose a novel local appearance triplet (LAT) loss to assist the training process by encouraging consistency between local appearances and saliency scores.
3) Extensive experiments are conducted to confirm the effectiveness of our approach. Our model achieves new state-of-the-art results on six popular SOD benchmarks.

II. Related Works

Traditional SOD Methods. Earlier salient object detection approaches are mainly based on local appearances [24], [41] and low-level cues (*e.g.*, center prior [1], [24], [25], boundary prior [1], [23], and background prior [26], [27]) to locate the saliency regions. However, these low-level cues may not always hold for complex natural images. Besides, inconsistent local appearances may degrade these traditional/hand-crafted methods, causing fragmented objects.

Fully-supervised Deep SOD Methods. In contrast to traditional methods, deep learning-based techniques have significantly improved SOD performances. In particular, these saliency detectors [10]–[14], [16]–[20], [42]–[46] are mostly trained in a fully-supervised fashion with pixel-level labels, leading to impressive results. For example, Zhuge *et al.* [18] propose the ICON model, which focuses on both micro- and macro-levels of salient objects by introducing three key components, diverse feature aggregation, integrity channel enhancement, and part-whole verification, for achieving integral SOD. Wang *et al.* [17] present Multiple Enhancement Network for SOD. They use a multi-level hybrid loss to guide the network in learning pixel-level, region-level, and object-level features, and a multi-scale feature enhancement module to refine global and detailed features. However, these fully-supervised methods require labor-intensive pixel-level labels.

Weakly-supervised Deep SOD Methods. To alleviate the pixel-level labeling cost, weakly-supervised approaches are proposed, utilizing scribbles [22], [47], [48], image-level labels [21], [49]–[51] or points supervision [52] to train the models. For example, Wang *et al.* [22] propose WBNet, a weakly-supervised SOD model trained using scribble annotations and multi-source pseudo-background labels. Yu *et al.* [47] aggregate multi-level features with a local coherence loss and a saliency structure consistency loss to predict pixel-wise saliency masks, utilizing only scribble-level labels during training. Zhang *et al.* [48] propose learning from scribble annotations for dense saliency prediction and using edge detection to capture the structure of the whole object. Piao *et al.* [21] propose a noise-robust adversarial learning framework for weakly supervised SOD, which leverages image-level labels to train a saliency network and a noise-robust discriminator network, effectively mitigating the impact of noisy pseudo labels. Despite the success, there is still a high cost to manually label large-scale image-level or scribble annotations.

Unsupervised Deep SOD Methods. To tackle the labeling cost, Zhang *et al.* [28] make an early attempt to learn from noisy labels generated by several unsupervised traditional methods. Subsequently, a line of works [2], [29]–[33], [53]–[55] aim to predict clear saliency masks through noise modeling or learning from class activation maps (CAM) [56]. Nguyen *et al.* [30] directly refine noisy labels in an iterative manner. Wang *et al.* [2] estimate the uncertainty maps of multiple noisy labels for clear saliency map predictions. Yasarla *et al.* [53] propose a self-supervised SOD method that integrates a self-supervised classification branch to generate CAMs as pseudo-GT labels, which are then enhanced with image edges to produce fine-grained outputs. Zhou *et al.* [54], [55] propose to utilize the activation maps from MOCO-v2 [39] as the noisy saliency guidance, and to enhance the predictions with more saliency cues from multi-modal data.

Instead of learning from noisy labels or CAM data, another line of works [3], [34]–[36], [57] propose to detect salient objects using clustering-based strategies with self-supervised models [37]–[39], which can be used for foreground object

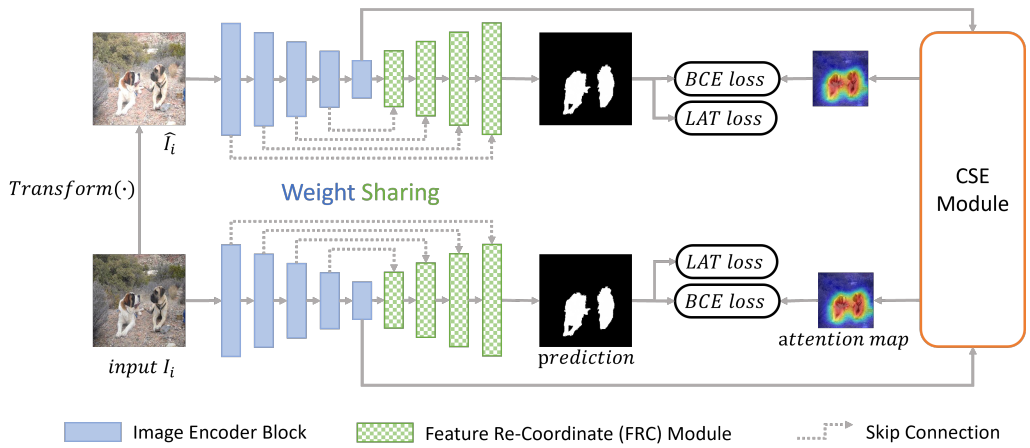


Fig. 3. **CSNet Architecture.** For simplicity, we illustrate CSNet for the case of using just one pair of images. For each input image I_i, we apply a random transformation to obtain a new view $\hat{I}_i$. Both I_i and $\hat{I}_i$ are processed by the same image encoder and pixel decoder. The pixel decoder, comprising four FRC modules, focuses on recovering spatial details, while the CSE module functions as the engine, providing high-level saliency to guide the model to predict. A LAT loss is proposed to assist the training for a fine-grained prediction.

discovery. For example, Siméoni *et al.* [3] propose the FOUND model, which involves identifying a background seed and then applying a re-weighting and clustering strategy over DINO [37] features to explicitly group all background pixels, thereby distinguishing the foreground objects. Wang *et al.* [35] construct a similarity graph based on self-supervised features, and treat SOD as a graph-cut problem. These methods mainly adopt a clustering-based strategy to group the salient regions together. However, we observe that these clustering-based methods usually suffer from under-detection.

Unlike the above methods, we detect salient objects according to where the model looks at when we force the model to perform an instance discrimination task based on a contrastive learning framework [40], and learn a fine-grained saliency mask using our proposed FRC decoder.

III. Our Approach

The proposed CSNet is a self-supervised model that aims to detect salient objects from RGB images without using any labels. Figure 3 shows the architecture. For each training iteration, we first randomly sample n images (denoted as $I_1, I_2, ..., I_n$) from the training set and create new views of these images individually using random transformations, to form the input of a batch size of $2n$, denoted as $I_1, I_2, ..., I_n, \hat{I}_1, \hat{I}_2, ..., \hat{I}_n$, where $\hat{I}_i$ is transformed from I_i:

$$\hat{I}_i = \text{Transform}(I_i). \tag{1}$$

We then feed these $2n$ images to an image encoder for feature extraction, and the image features from the last layer of the encoder are forwarded to the CSE module, to mimic the human visual attention mechanism through an instance discriminative task, providing high-level saliency guidance for the pixel decoder. We construct the pixel decoder using four FRC modules to facilitate a fine-grained output. The proposed FRC module aims to align high-level features with low-level features, thereby enriching spatial information. In addition, we propose a LAT loss to encourage nearby pixels with homogeneous appearances to have similar saliency scores.

Notably, the CSNet is a self-supervised model. While it does not require any external labels or annotations, it is able to output high-quality predictions.

The rest of this section is organized as follows. We describe the CSE module in Section III-A, the FRC module in Section III-B, the LAT loss in Section III-C, and the training strategy in Section III-D.

A. Contrastive Saliency Extraction (CSE) Module

The proposed CSE module is designed to mimic human visual attention for high-level saliency extraction. By optimizing the module to perform an instance discriminative task, the module can learn to focus on the most salient and informative regions to match the same images under different views. Figure 4 shows the design of the CSE module. The CSE module takes the dense image features $F \in R^{2n\times d\times hw}$ from the encoder, where $2n$ is the batch size, d is the feature dimension, and hw is the spatial resolution. First, we introduce an aggregation function to generate an image-level saliency query, $query \in R^{2n\times d}$, which acts as a prompt for saliency extraction. $query$ should effectively capture the most salient objects in the image while retaining partial information about the object structures, so that it can help retrieve the integral salient objects. Since global max pooling identifies high-response pixels and global average pooling represents a broader area, we adopt a balanced strategy that lies between global max pooling and global average pooling:

$$\text{agg}(f) = \text{softmax}(f)^T \cdot f, \tag{2}$$

where $f \in R^{hw}$ is a single-channel feature map in F. We apply $\text{agg}(\cdot)$ to each single-channel feature map in F to obtain the image-level saliency query $query \in R^{2n\times d}$. Our aggregation function enables the saliency query to consider all pixels in the feature maps, with high-response regions having a greater impact on the saliency query. Thereafter, we employ a standard cross-attention layer [58] followed by a multi-

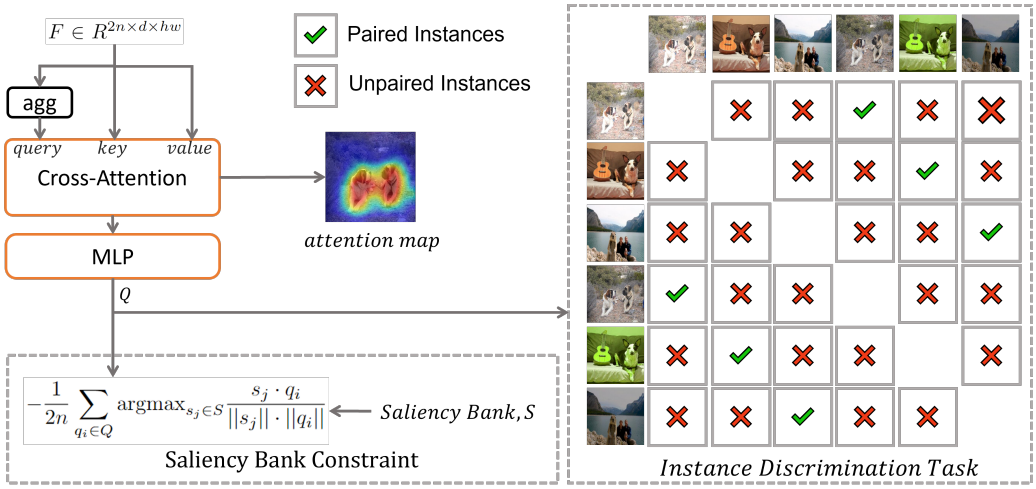


Fig. 4. **The CSE Module.** We first generate a global saliency query from the image features F via an aggregation function, *i.e.*, $query = \text{agg}(F)$. We incorporate a cross-attention layer and a multi-layer perceptron to mimic the human visual attention. We add a saliency bank S to accelerate the convergence and prevent model from corruption by encouraging the sharing of common attributes among the saliency queries $Q \in R^{2n\times d}$. We optimize the module by connecting it to an instance discriminative task. agg is an aggregation function to map the image dense features F to the image-level saliency query.

layer perceptron to mimic human visual attention mechanisms. Mathematically, the attention layer is represented as:

$$attn_i = \text{softmax}(\frac{query_i \cdot F_i}{\sqrt{d}}), \tag{3}$$

$$q_i = \text{MLP}(F_i \cdot attn_i), \tag{4}$$

where $query_i \in R^d$ and $F_i \in R^{d\times hw}$ are the i^{th} item of $\text{agg}(F)$ and the i^{th} feature maps of F. $attn_i$ is the attention map of the i^{th} image, and it is subsequently reshaped to $h \times w$ to indicate where the model should focus. We further update the saliency query to output $Q \in R^{2n\times d}$, which is a weighted sum of the attention map $attn$ and image features F, followed by a multi-layer perceptron (MLP).

We train the CSE module to perform an instance discriminative task in a contrastive learning framework [40], whose objective is to bring the queries of the same image under different transformations closer, while pushing away the queries from other images. Mathematically, the adopted contrastive loss is defined as:

$$\ell_{CL} = \frac{1}{n}\sum_{i=1}^{n} -\log\frac{e^{\text{sim}(q_i,\hat{q_i})/\tau}}{\sum_{j=1}^{n}\mathbb{1}_{[j\neq i]}e^{\text{sim}(q_i,q_j)/\tau} + e^{\text{sim}(q_i,\hat{q_j})/\tau}}, \tag{5}$$

where $q_*, \hat{q_*}$ are the queries of image I_* and $\hat{I}_*$ respectively. τ is a temperature parameter. $\text{sim}(\cdot,\cdot)$ is a cosine similarity function, and $\mathbb{1}_{[cond]}$ is the indicator function that outputs 1 when the condition is true and 0 otherwise. By minimizing this contrastive loss, ℓ_{CL}, the module is tasked with selecting the correct image, which is a random view of itself, from the batch. As the training process unfolds, the attention map will gradually converge on the salient objects. This process mimics the human visual mechanism, which tends to assign visual attention to salient objects.

To accelerate the convergence and prevent model corruption, we introduce a saliency bank $S \in R^{m\times d}$ to encourage image-level saliency queries Q to share some common attributes:

$$\ell_{bank} = -\frac{1}{2n}\sum_{q_i\in Q}\text{argmax}_{s_j\in S}\frac{s_j\cdot q_i}{||s_j||\cdot||q_i||}, \tag{6}$$

where $q_i \in R^d$ is the i^{th} entry of the saliency query Q, $s_j \in R^d$ is the i^{th} entry in S. m is the size of the saliency bank. We minimize ℓ_{bank} to force each saliency query $q_i \in Q$ close to some entries in S, and we force the saliency queries to share common attributes with each other. Additionally, this term can be treated as a regularization term, preventing the attention map from being trapped in a trivial solution. For example, when the attention map is active for all pixels, the queries will become more diverse, and exhibit fewer common attributes. Consequently, ℓ_{bank} will increase, which is opposite to the optimization direction.

To provide a more precise saliency map for the pixel decoder, we apply a min-max normalization to the attention map followed by an up-sampling and a Conditional Random Field (CRF) [59] operation, as:

$$saliency = \text{CRF}(\text{Up}(\text{MinMaxNorm}(attention\,map))). \tag{7}$$

We use these coarse saliency maps to guide the pixel decoder to output more fine-grained predictions.

B. Feature Re-Coordinate (FRC) Module

We adopt a pixel decoder to recover spatial details from low-level features, as low-level features contain richer object details and spatial information. However, owing to the down-sampling operations in the bottom-up path, spatial misalignment exists between high-level features f_{high} and low-level features f_{low}, hindering the detection around object boundaries. Previous alignment methods, *e.g.*, [60], require high-resolution object masks to predict the offsets, which are not available in the unsupervised domain. Hence, we propose a novel FRC module to handle spatial misalignment in an unsupervised manner.

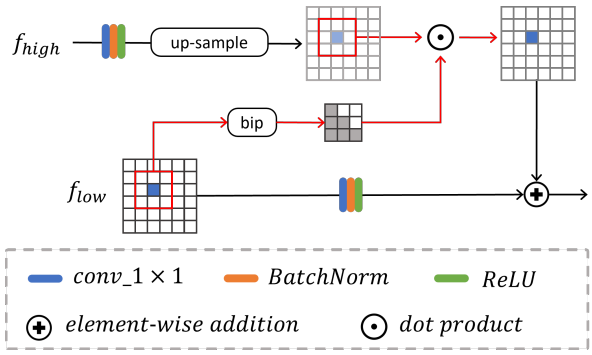


Fig. 5. **The FRC Module.** We attend to every pixel in the up-sampled f_{high} and recompute its value as a weighted sum of its surrounding pixels (including itself), where the weight assigned to each pixel is computed by a bip function conditioned on f_{low}. bip is a bipartition function to compute a weight matrix, enabling the re-weighting of the high-level features to address the spatial misalignment problem.

As shown in Figure 5, we use a sliding window of size $k \times k$ (red box) moving over the up-sampled f_{high} and f_{low} synchronously. For each movement, we bipartition the pixels inside the sliding window over f_{low} via a bip function and select the partition containing the central pixel to guide the spatial alignment. We define the bip function as:

$$\text{binary\_bip}(x)_{i,j} = \begin{cases} \mathbb{1}_{[x_{i,j} > \bar{x}]} & \text{if } x_c > \bar{x} \\ \mathbb{1}_{[x_{i,j} \leq \bar{x}]} & \text{otherwise,} \end{cases} \tag{8}$$

$$\text{bip}(x)_{i,j} = \frac{\text{binary\_bip}(x)_{i,j}}{\text{sum}(\text{binary\_bip}(x))}, \tag{9}$$

where $x \in R^{k \times k}$ is the input matrix of the bip function. $\bar{x}$ is the mean of the elements of x. x_c is the value of the central pixel of x, and $\text{sum}(\text{binary\_bip}(x))$ sums up the elements of $\text{binary\_bip}(x)$ for the normalization purpose. Thus, each pixel of the up-sampled high-level features turns out to be a weighted sum of its surrounding pixels (including itself), where the weight assigned to it is conditioned on the low-level features, which contain rich spatial details. Compared to simple up-sampling methods, *e.g.*, bilinear, our FRC module corrects the spatial misalignment dynamically conditioned on spatial-enrich low-level features. Finally, we fuse the high-level features with low-level features through an element-wise addition. We stack four FRC modules to build the pixel decoder to work in a similar way as the Feature Pyramid Networks (FPNs) [61].

C. Local Appearance Triplet (LAT) Loss

We propose a LAT loss to assist the FRC module in predicting fine-grained saliency maps. The proposed LAT loss encourages the nearby pixels with homogeneous appearances to have similar saliency scores with a triplet loss [62], which aims to minimize the distance between the anchor and the positive, and maximize the distance between the anchor and the negative.

Given a saliency map prediction $P \in R^{h \times w}$ and the anchor $P_{i,j} \in [0, 1]$ indicating the saliency score, we first compute the saliency appearance a_s and background appearance a_b as:

$$a_s = \frac{1}{w_s} \sum_{i'=i-r}^{i+r} \sum_{j'=j-r}^{j+r} P_{i',j'} * C_{i',j'}, \tag{10}$$

$$a_b = \frac{1}{w_b} \sum_{i'=i-r}^{i+r} \sum_{j'=j-r}^{j+r} (1 - P_{i',j'}) * C_{i',j'}, \tag{11}$$

where $C_{i',j'}$ represents the RGB color of the selected pixel. r is the radius of a local window that we are processing. $\frac{1}{w_s}, \frac{1}{w_b}$ are the normalized weights. If the anchor $P_{i,j} > 0.5$, we suppose that the anchor pixel is more likely to be a salient pixel, and could define the positive as saliency appearance a_s while the negative as a_b, and vice versa. Therefore, we can define the positive, p^+, and the negative, p^-, for each $r \times r$ local window, conditioned on $P_{i,j}$ as:

$$p^+ = \begin{cases} a_s & \text{if } P_{i,j} > 0.5 \\ a_b & \text{otherwise,} \end{cases} \quad p^- = \begin{cases} a_b & \text{if } P_{i,j} > 0.5 \\ a_s & \text{otherwise.} \end{cases} \tag{12}$$

Now, we can compute the triplet loss for $P_{i,j}$ as:

$$\ell_{P_{i,j}} = \max(d(C_{i,j}, p^+) - d(C_{i,j}, p^-) + \epsilon, 0), \tag{13}$$

where ϵ is the separation margin. $d(\cdot)$ is the distance metric using a kernel method, *i.e.*, $d(x, y) = -e^{-\kappa ||x-y||^2}$. κ is a hyperparameter. We visit all possible anchors in the image to compute the LAT loss. In addition, we add an $L1$ regularization term to encourage no confusion between saliency parts and the background. Thus, the LAT loss is defined as:

$$\ell_{LAT} = \frac{1}{hw} \sum_{P_{i,j} \in P} (\ell_{P_{i,j}} - |P_{i,j} - 0.5|). \tag{14}$$

D. Training Strategy

Loss Functions. We use the binary cross-entropy (BCE) loss (denoted as ℓ_{bce}) to connect the CSE module and the pixel decoder when doing backward optimization. Besides, we employ a consistent loss (denoted as ℓ_{cons}) between the predictions of two different views as:

$$\ell_{cons} = \frac{1}{n} \sum_{i=1}^{n} |P_i - \hat{P}_i|, \tag{15}$$

where $P_*, \hat{P}_*$ are the predictions of I_* and $\hat{I}_*$. We use the $L1$ loss to ensure consistent predictions for the same image. Thus, the total loss of the CSNet is:

$$\ell_{total} = \alpha \ell_{CL} + \gamma \ell_{bank} + \lambda \ell_{LAT} + \theta \ell_{bce} + \delta \ell_{cons}, \tag{16}$$

where $\alpha, \gamma, \lambda, \theta, \delta$ are hyper-parameters for loss balance.

Self-Training. To make our saliency detector generalize well to unseen images, we adopt a self-training stage following previous works [3], [36], [54]. Specifically, we employ Self-Mask [66] as our detector, and fine-tune it on our predictions as pseudo labels.

TABLE I
QUANTITATIVE COMPARISON ON SALIENT OBJECT DETECTION. FS: FULLY-SUPERVISED METHODS THAT REQUIRE PIXEL-LEVEL HUMAN LABELS OR SYNTHESIS LABELS. WS: WEAKLY-SUPERVISED METHODS THAT REQUIRE IMAGE-LEVEL OR SCRIBBLE-LEVEL LABELS. US: UNSUPERVISED METHODS THAT DO NOT REQUIRE ANY LABELS. **OURS**$_{attn}$: EVALUATION ON THE ATTENTION MAPS FROM THE CSE MODULE. **OURS**$_{init}$: EVALUATION ON THE PREDICTIONS FROM CSNET WITHOUT THE SELF-TRAINING STAGE. **OURS**$_{full}$: OUR FULL MODEL WITH THE SELF-TRAINING STAGE. THE BEST RESULTS OF EACH CHUNK ARE IN **BOLD**.

Method	Year-Pub	Mode	DUTS-TE [63]						ECSSD [64]					
			ACC↑	IoU↑	F_β ↑	MAE↓	S_m ↑	E_m ↑	ACC↑	IoU↑	F_β ↑	MAE↓	S_m ↑	E_m ↑
EGNet [10]	ICCV-2019	FS	96.5	78.2	0.857	0.039	0.887	0.907	96.7	87.8	0.933	0.037	0.925	0.947
LDF [13]	CVPR-2020	FS	96.7	79.8	0.867	0.034	0.892	0.929	96.8	88.0	0.933	0.034	0.924	0.951
GateNet [14]	ECCV-2020	FS	96.6	79.0	0.866	0.039	0.890	0.905	96.7	87.6	0.933	0.038	0.924	0.945
VST [16]	ICCV-2021	FS	96.7	80.2	0.869	0.038	0.896	0.916	97.3	89.3	0.936	0.034	0.932	0.957
UDASOD [65]	AAAI-2022	FS	95.0	72.6	0.799	0.050	0.846	0.896	95.8	84.5	0.895	0.043	0.899	0.940
ICON [18]	TPAMI-2023	FS	96.4	79.2	0.861	0.037	0.888	0.919	97.0	88.8	0.935	0.032	0.929	0.954
MENet [17]	CVPR-2023	FS	97.3	82.5	0.885	0.028	0.904	0.942	97.1	88.9	0.934	0.031	0.928	0.954
VSCode [19]	CVPR-2024	FS	**97.8**	**86.2**	**0.913**	**0.024**	**0.926**	**0.954**	**98.2**	**92.3**	**0.954**	**0.022**	**0.949**	**0.969**
MSW [50]	CVPR-2019	WS	91.8	51.7	0.691	0.092	0.759	0.815	92.2	66.4	0.828	0.099	0.827	0.884
SODSA [48]	CVPR-2020	WS	93.9	63.8	0.753	0.062	0.803	0.869	94.2	77.4	0.871	0.059	0.865	0.917
MFNet [49]	ICCV-2021	WS	92.6	57.9	0.724	0.079	0.778	0.832	92.2	71.2	0.850	0.084	0.837	0.889
NSAL [21]	TOM-2023	WS	92.8	60.1	0.738	0.073	0.781	0.850	92.3	72.0	0.857	0.078	0.834	0.889
WBNet [22]	PR-2024	WS	**96.3**	**76.8**	**0.852**	**0.038**	**0.876**	**0.915**	**96.9**	**86.7**	**0.929**	**0.032**	**0.918**	**0.937**
RBD [1]	CVPR-2014	US	77.9	36.6	0.444	0.305	0.567	0.665	82.5	52.8	0.626	0.271	0.668	0.707
SBF [28]	ICCV-2017	US	-	-	-	-	-	-	92.4	70.6	0.832	0.091	0.832	0.876
EDNS [31]	ECCV-2020	US	94.1	65.5	0.779	0.065	0.820	0.850	93.8	77.4	0.882	0.068	0.871	0.906
TokenCut [35]	CVPR-2022	US	92.0	64.4	0.759	0.128	0.760	0.757	93.5	77.4	0.875	0.128	0.834	0.854
UMNet [2]	CVPR-2022	US	93.5	63.1	0.758	0.067	0.803	0.863	93.8	77.3	0.884	0.064	0.868	0.904
SelfMask [66]	CVPRW-2022	US	93.8	69.0	0.809	0.062	0.819	0.881	93.6	79.5	0.895	0.064	0.864	0.911
A2S [54]	TCSVT-2023	US	93.2	65.0	0.740	0.069	0.805	0.847	94.5	80.1	0.889	0.056	0.877	0.921
CutLER [36]	CVPR-2023	US	86.9	57.3	0.660	0.131	0.737	0.767	92.3	76.7	0.850	0.077	0.846	0.887
Found [3]	CVPR-2023	US	94.3	67.9	0.784	0.057	0.815	0.876	95.1	81.3	**0.915**	0.049	0.881	**0.930**
HEAP [57]	AAAI-2024	US	94.9	68.7	-	-	-	-	**96.2**	**82.3**	-	-	-	-
3SD [53]	WACV-2024	US	93.1	66.5	0.755	0.087	0.812	0.798	95.2	**82.3**	0.898	0.077	0.885	0.891
Ours$_{attn}$	-	US	85.3	43.9	0.515	0.257	0.604	0.667	87.4	60.5	0.693	0.252	0.692	0.691
Ours$_{init}$	-	US	94.4	64.1	0.776	0.056	0.800	0.859	94.7	79.3	0.896	0.053	0.871	0.917
Ours$_{full}$	-	US	**95.7**	**73.4**	**0.852**	**0.043**	**0.853**	**0.915**	95.3	82.0	0.912	**0.048**	**0.886**	0.922
Method	Year-Pub	Mode	HKU-IS [67]						PASCAL-S [68]					
			ACC↑	IoU↑	F_β ↑	MAE↓	S_m ↑	E_m ↑	ACC↑	IoU↑	F_β ↑	MAE↓	S_m ↑	E_m ↑
EGNet [10]	ICCV-2019	FS	97.3	85.6	0.917	0.031	0.918	0.955	92.9	75.6	0.833	0.074	0.852	0.877
LDF [13]	CVPR-2020	FS	97.4	86.4	0.919	0.028	0.919	0.960	94.2	77.8	0.847	0.060	0.863	0.905
GateNet [14]	ECCV-2020	FS	97.4	86.1	0.921	0.031	0.921	0.956	94.0	77.3	0.847	0.065	0.865	0.889
VST [16]	ICCV-2021	FS	97.6	87.5	0.926	0.030	0.928	0.960	94.3	79.2	0.847	0.062	0.872	0.902
UDASOD [65]	AAAI-2022	FS	96.5	83.4	0.885	0.035	0.897	0.947	92.2	72.6	0.792	0.078	0.824	0.876
ICON [18]	TPAMI-2023	FS	97.3	86.5	0.919	0.029	0.920	0.958	93.7	77.8	0.843	0.064	0.861	0.893
MENet [17]	CVPR-2023	FS	97.8	88.0	0.929	0.023	0.927	0.965	94.8	79.7	0.858	0.053	0.872	0.913
VSCode [19]	CVPR-2024	FS	**98.1**	**90.0**	**0.943**	**0.021**	**0.940**	**0.972**	**95.1**	**81.8**	**0.870**	**0.051**	**0.886**	**0.922**
MSW [50]	CVPR-2019	WS	93.5	64.3	0.813	0.086	0.818	0.895	88.8	57.5	0.741	0.134	0.768	0.791
SODSA [48]	CVPR-2020	WS	95.4	76.6	0.863	0.047	0.865	0.932	90.8	66.9	0.778	0.092	0.797	0.857
MFNet [49]	ICCV-2021	WS	94.8	73.0	0.854	0.058	0.852	0.919	89.4	62.0	0.763	0.112	0.782	0.824
NSAL [21]	TOM-2023	WS	94.9	74.3	0.867	0.051	0.854	0.923	89.0	61.7	0.759	0.110	0.767	0.826
WBNet [22]	PR-2024	WS	**97.2**	**85.2**	**0.920**	**0.029**	**0.913**	**0.958**	**93.5**	**76.4**	**0.840**	**0.066**	**0.851**	**0.872**
RBD [1]	CVPR-2014	US	82.3	49.9	0.583	0.270	0.649	0.739	78.5	45.2	0.563	0.297	0.620	0.645
SBF [28]	ICCV-2017	US	-	-	-	-	-	-	87.9	59.1	0.724	0.133	0.758	0.790
EDNS [31]	ECCV-2020	US	95.9	78.8	0.891	0.046	0.884	0.933	91.4	68.5	0.813	0.094	0.820	0.846
TokenCut [35]	CVPR-2022	US	93.6	67.2	0.831	0.122	0.778	0.851	89.7	67.5	0.782	0.151	0.771	0.776
UMNet [2]	CVPR-2022	US	96.0	79.6	0.892	0.041	0.887	0.939	89.6	63.9	0.771	0.105	0.785	0.830
SelfMask [66]	CVPRW-2022	US	95.7	78.7	0.895	0.043	0.870	0.926	90.1	69.6	0.806	0.099	0.797	0.860
A2S [54]	TCSVT-2023	US	95.9	80.0	0.873	0.041	0.881	0.936	90.1	67.0	0.774	0.100	0.794	0.839
CutLER [36]	CVPR-2023	US	92.7	75.4	0.837	0.073	0.844	0.886	89.1	67.7	0.764	0.109	0.786	0.834
Found [3]	CVPR-2023	US	96.0	79.7	0.903	0.040	0.877	0.936	92.5	72.0	0.821	0.075	0.820	0.884
3SD [53]	WACV-2024	US	95.9	80.2	0.877	0.068	0.877	0.906	90.6	69.1	0.791	0.117	0.807	0.812
Ours$_{attn}$	-	US	87.2	55.8	0.633	0.249	0.666	0.726	83.2	51.9	0.604	0.277	0.643	0.602
Ours$_{init}$	-	US	96.2	78.9	0.895	0.038	0.875	0.929	91.9	69.5	0.797	0.081	0.806	0.864
Ours$_{full}$	-	US	**96.9**	**83.3**	**0.926**	**0.031**	**0.901**	**0.949**	**92.9**	**73.4**	**0.834**	**0.071**	**0.831**	**0.885**

IV. EXPERIMENTS

A. Implementation Details

We adopt a ResNet-50 [70] pretrained with DINO [37] as the encoder, while the remaining parameters are initialized randomly. To create new views $\hat{I}_*$ from the original input images I_*, we apply a family of color distortions, including color shifting, contrast changes, brightness changes and random hues, on the original input images. We set the hyper-parameters as follows: temperature parameter τ = 0.1, saliency bank size m = 512, sliding window size k = 3, radius of each local window r = 5, separation margin ϵ = 0.5, and κ = 22. In terms of loss balance weights, $\alpha, \gamma, \lambda, \theta, \delta$ are empirically set to 1. We train CSNet for 13,200 iterations with a batch size of $2n$ = 32. We adopt the SGD optimizer with a weight decay 5e-4. The learning rate is set to 2.5e-3

TABLE II
QUANTITATIVE COMPARISON (CONTINUE).

Method	DUT-OMRON [69]					
	ACC↑	IoU↑	F_β ↑	MAE↓	S_m ↑	E_m ↑
EGNet [10]	95.1	70.2	0.767	0.053	0.841	0.874
LDF [13]	95.0	71.1	0.773	0.052	0.839	0.881
GateNet [14]	95.0	70.3	0.776	0.055	0.840	0.869
VST [16]	94.6	73.1	0.791	0.058	0.850	0.872
UDASOD [65]	94.1	66.3	0.727	0.059	0.808	0.849
ICON [18]	94.5	72.3	0.786	0.057	0.844	0.879
MENet [17]	95.6	73.1	0.785	0.045	0.850	0.891
VSCode [19]	**95.9**	**77.9**	**0.831**	**0.043**	**0.877**	**0.910**
MSW [50]	91.8	54.5	0.666	0.108	0.756	0.764
SODSA [48]	93.3	61.2	0.707	0.068	0.785	0.845
MFNet [49]	90.6	49.8	0.632	0.098	0.726	0.784
NSAL [21]	91.2	54.8	0.656	0.088	0.745	0.802
WBNet [22]	**95.3**	**73.8**	**0.804**	**0.048**	**0.855**	**0.894**
CutLER [36]	84.1	52.4	0.590	0.159	0.696	0.717
Found [3]	92.2	61.3	0.704	0.078	0.772	0.822
HEAP [57]	92.9	64.6	-	-	-	-
3SD [53]	92.4	**65.1**	0.720	0.094	**0.798**	0.790
Ours$_{attn}$	82.6	40.9	0.485	0.275	0.592	0.651
Ours$_{init}$	91.9	53.6	0.663	0.081	0.734	0.778
Ours$_{full}$	**93.2**	64.2	**0.760**	**0.068**	0.794	**0.846**
Method	SOC [46]					
	ACC↑	IoU↑	F_β ↑	MAE↓	S_m ↑	E_m ↑
ICON [18]	**90.4**	**65.4**	**0.736**	**0.097**	**0.781**	**0.823**
NSAL [21]	**84.0**	**45.6**	**0.592**	**0.161**	**0.659**	**0.719**
CutLER [36]	84.2	52.6	0.622	0.158	0.680	0.732
Found [3]	88.1	57.5	0.678	0.119	0.721	0.785
3SD [53]	86.1	55.0	0.662	0.158	0.716	0.730
Ours$_{attn}$	75.8	40.0	0.481	0.331	0.561	0.521
Ours$_{init}$	**88.3**	58.1	0.689	**0.117**	0.728	0.796
Ours$_{full}$	**88.3**	**60.2**	**0.704**	**0.117**	**0.738**	**0.804**

without a decay strategy. We apply random horizontal flipping, random cropping, and multi-scale input images on the mini-batch for data augmentation. We use Conditional Random Field (CRF) [59] to post-process the predictions.

B. Datasets and Evaluation Metrics

Following previous USOD methods [2], [3], [17], [19], [54], we train our CSNet on the *DUTS-TR* [63] dataset, which contains 10,553 diverse images, and conduct testing on six widely used saliency detection benchmarks:

1) *DUTS-TE* [63] consisting of 5,019 images captured from various challenging scenes,
2) *ECSSD* [64] containing 1,000 images with the salient objects mostly located around the center of the image,
3) *HKU-IS* [67] containing 4,447 images in which most of them have multiple salient objects, and
4) *PASCAL-S* [68] containing 850 images covering a broad range of object categories.
5) *DUT-OMRON* [69] containing 5,168 images with relatively complex background.
6) *SOC* [46] containing 1,200 testing images captured in clutter environments.

We emphasize that our model requires only input images for training. It does not rely on any external annotations. We report accuracy, intersection over union, mean F_β, mean absolute error, S-measure [71], and E-measure [72] for the comparison with existing methods. The mean F_β is computed as:

$$F_\beta = (1+\beta^2) * \frac{Precision * Recall}{\beta^2 * Precision + Recall}, \qquad (17)$$

where $\beta^2 = 0.3$. All images share a fixed threshold, *i.e.*, 0.5, when computing precision, recall, accuracy and IoU scores.

C. Quantitative Results

To fully understand the effectiveness of our approach, we compare it with 24 SOD methods, including 8 fully-supervised methods [10], [13], [14], [16]–[19], [65], 5 weakly-supervised methods [21], [22], [48]–[50], and 11 unsupervised methods [1]–[3], [28], [31], [35], [36], [53], [54], [57], [66].

As shown in Tables I and II, our method makes a significant improvement over existing unsupervised SOD methods on nearly all metrics across six benchmarks. Specifically, when compared to 3SD [53], our method (*i.e.*, **Ours**$_{full}$) improves the IoU score by 10.4%, the mean F_β score by 12.8%, the S_m score by 5.0% and the E_m score by 14.7%, on *DUTS-TE*. When compared to Found [3], our method improves the IoU score by 8.1% and the F_β score by 8.7%, on *DUTS-TE*. Our method also consistently surpasses those unsupervised methods [2], [28], [31] that are trained on noisy labels produced by traditional methods. For example, comparing with UMNet [2], our method enhances the IoU score by an average of 10.5% and the F_β score by 6.9%.

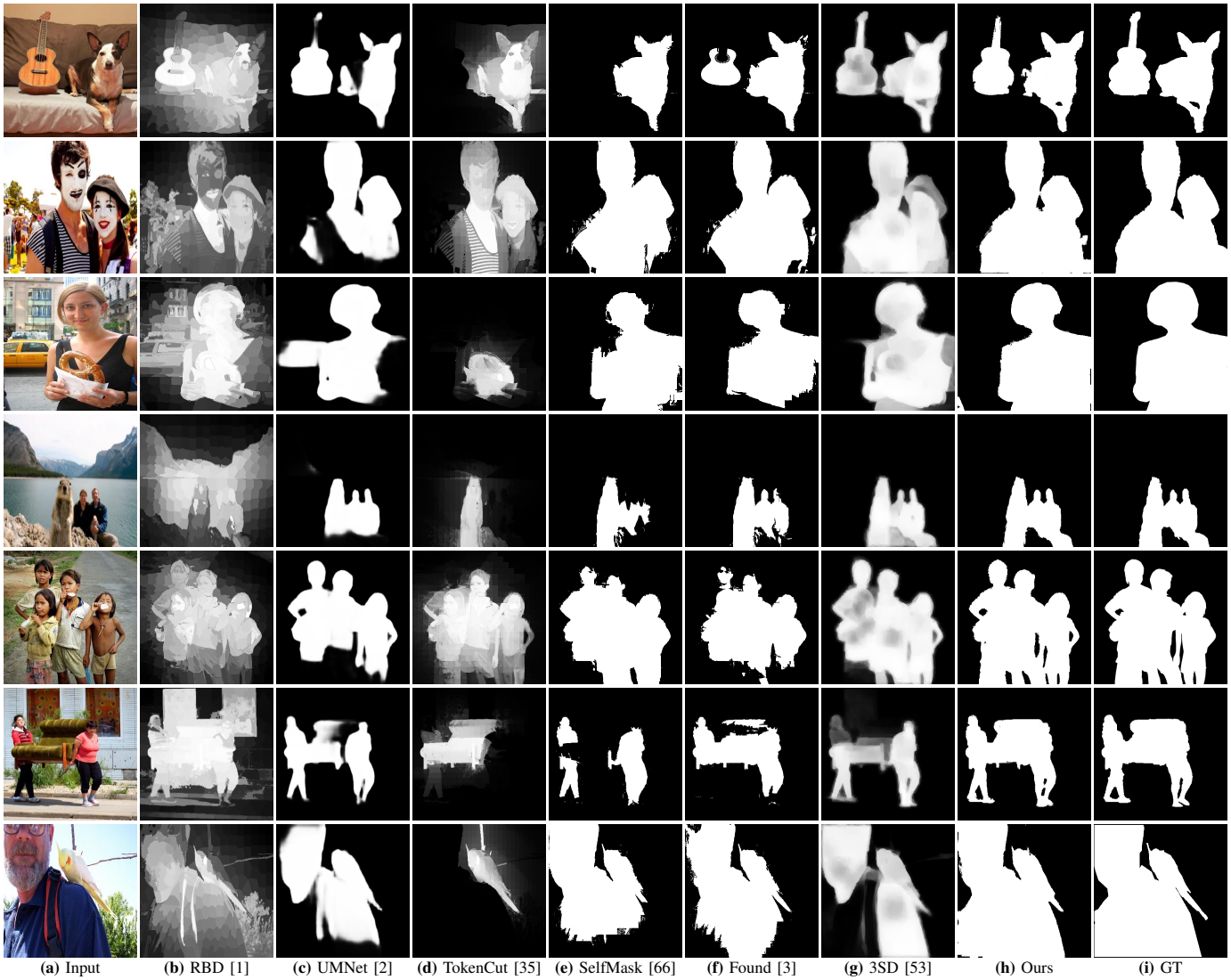
When compared to weakly-supervised methods [21], [22], [48]–[50] that are trained with image-level or scribble-level annotations, our method is comparable to WBNet [22], but surpasses the other weakly-supervised approaches, including NSAL [21] and MFNet [49] by a clear margin.

We also observe that our method only slightly improves the MAE score and S-measure score on the *ECSSD* benchmark. We think that this is due to the strong center bias, as the salient objects in *ECSSD* tend to be located at the center. This center bias benefits all methods, resulting in a smaller performance gap. With regard to the challenging SOC (Salient Object in Clutter) benchmark, our method consistently improves all metrics compared to state-of-the-art unsupervised approaches, as shown in Table II.

We have also included two intermediate versions of our model, *i.e.*, $Ours_{attn}$ and $Ours_{init}$, in the comparison, where $Ours_{attn}$ reports the evaluation results of the attention maps by the CSE module and $Ours_{init}$ does not include a self-training stage. We find that $Ours_{attn}$ is much better than the hand-crafted method RBD [1]. In addition, $Ours_{init}$, which omits the self-training stage, still performs comparably to Found [3]. These two comparisons suggest that our method excels at extracting high-quality saliency maps, thus leading to an improved salient object detector.

D. Visual Results

Qualitative Comparison. Figure 6 shows the qualitative comparison. We can see that the hand-crafted method RBD [1] can only output object fragments, *e.g.*, the persons in the 2^{nd} row, and may lose sight of salient objects near the image boarders, *e.g.*, the squirrel in the 4^{th} row. UMNet [2], which is an unsupervised method trained with noisy labels from hand-crafted techniques, fails to detect the complete salient object, *e.g.*, the dog in the 1^{st} row, the girl in the 3^{rd} row and the



(a) Input **(b)** RBD [1] **(c)** UMNet [2] **(d)** TokenCut [35] **(e)** SelfMask [66] **(f)** Found [3] **(g)** 3SD [53] **(h)** Ours **(i)** GT

Fig. 6. **Qualitative Comparison.** Our approach can generally produce more favorable results in various challenging scenes. In contrast, RBD [1] and UMNet [2] tend to output object fragments. TokenCut [35], SelfMask [66], Found [3] and 3SD [53] suffer from under-detection.

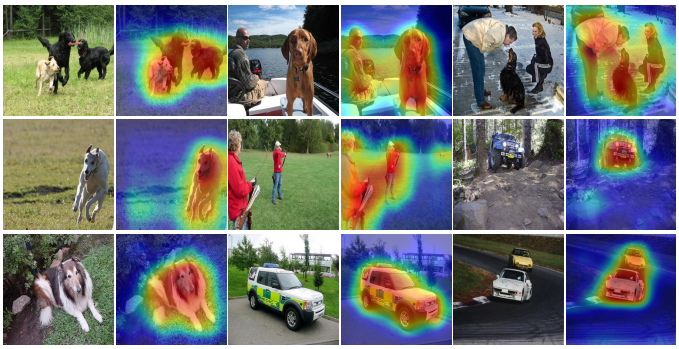


Fig. 7. **Attention Map Visualization.** The CSE module can generally produce high-quality saliency maps from the input images, where there can be multiple salient objects (1^{st} row), the salient objects may be off-centered (2^{nd} row) and the salient objects/parts may have diverse appearances (3^{rd} row).

squirrel in the 4^{th} row. It may also detect the background objects by mistake, *e.g.*, the taxi in the 3^{rd} row.

Unsupervised clustering-based models [3], [35], [66] often suffer from under-detection. For example, TokenCut [35] tends to output a single object, *e.g.*, $1^{st}, 3^{rd}$, 4^{th}, 6^{th} and 7^{th} rows, as it assumes that only one partition is salient in a scene. Found [3], on the other hand, tends to miss some parts of the objects when there are large semantic/appearance gaps between image patches, *e.g.*, the guitar in the 1^{st} row and the persons in the 5^{th} and 6^{th} rows. 3SD [53] may also suffer from under-detection, producing low-confidence saliency scores. For example, it fails to detect the persons in the last two rows.

In contrast, our approach in Figure 6(h) generally yields more accurate salient object masks with clear object boundaries in various challenging scenes. These scenes might include multiple salient objects shown in the 5^{th} and 6^{th} rows, complex backgrounds shown in the 2^{nd} and 3^{rd} rows, or a substantial appearance gap between salient objects or parts shown in the $1^{st}, 2^{nd}$ and 3^{rd} rows.

Attention Map Visualization. To provide a more intuitive understanding of the attention maps by the CSE module, we upscale the low-resolution attention maps and visualize them in Figure 7. The results show that our approach can effec-

TABLE III
COMPLEXITY ANALYSIS. WE EVALUATE THE MODELS BELOW ON A PC WITH AN RTX4090 GPU AND AN INTEL I7-13700 CPU.

Method	Heavy Part	FLOPs	Parameters	Runtime
VSCode [19] (2024)	Swin-T [73]	72.8G	54.0M	31ms
3SD [53] (2024)	U2Net [74]	52.8G	58.6M	18ms
CSNet (Ours)	ResNet-50 [70]	12.8G	27.7M	5ms
Our Encoder	ResNet-50 [70]	10.203G	23.508M	-
Our Decoder	FRC Module	2.463G	4.160M	-
Our CLS Head	MLP	0.137G	0.018M	-

tively mimic the human visual attention mechanism and can highlight the salient semantic objects in challenging scenes.

E. Complexity Analysis

We further conduct a complexity analysis on CSNet. Our CSNet has 27.7M parameters and an inference time of $5ms$ per image on an Intel i7-13700 PC with a RTX4090 GPU. We include a complexity comparison with VSCode [19] (2024) and 3SD [53] (2024) in Table III. Overall, our method demonstrates superior efficiency. In addition, we provide a deeper complexity analysis on the key components of our CSNet, which comprises an image encoder, a pixel decoder and a classification head. It shows that the proposed FRC decoder is very lightweight, with only 4.16M parameters.

F. Ablation Study

We further conduct experiments to gain a better understanding of the effectiveness of each component in our model. For simplicity, all the models presented below are trained without the self-training stage, unless stated otherwise.

Choices of the Aggregation Function. The CSE module introduced in Section III-A uses an aggregation function, *i.e.*, agg, to generate saliency query, which can be interpreted as a prompt for high-level saliency extraction. In our design, we choose the weighted sum strategy, as depicted in Eq. 2. However, we have experimented with another three options to generate the saliency query. The first one is the global average pooling. The second one is the global max pooling under the hypothesis that salient objects have higher responses in the high-level feature maps. We further select the central pixel among the high-level feature maps as the saliency query under the center-prior assumption [23]. The experimental results are presented in Table IV. Our weighted sum solution stands out as the best, followed by the center-prior assumption, while the global average/max pooling methods do not perform well.

We also show some visual examples of these options in Figure 8. We can see that the max pooling method often focuses on the most salient pixels, which can lead to under-detect the salient objects, *e.g.*, the dog in the 1^{st} row and the woman in the 2^{nd} row. Conversely, the average pooling method tends to over-detect the salient objects due to its smoothing effect and equal weighting over the whole scene, *e.g.*, the roadside sign in the 1^{st} row. The center-prior assumption method may predict false positives in the center region of the scene, *e.g.*, the 2^{nd} row. In contrast, our weighted sum solution consistently yields more accurate results, which are also demonstrated by the accuracy, IoU and F_β scores in Table IV.

TABLE IV
THE CHOICES OF AGGREGATION FUNCTION. WE EXPERIMENT WITH FOUR OPTIONS: GLOBAL MAX POOLING (MAX-POOL), GLOBAL AVERAGE POOLING (AVG-POOL), CENTRAL-PRIOR ASSUMPTION (CENTRAL) AND OUR WEIGHTED SUM STRATEGY (OURS).

Choices	DUTS-TE [63]			ECSSD [64]		
	ACC↑	IoU↑	F_β ↑	ACC↑	IoU↑	F_β ↑
Max-Pool	91.6	55.2	0.714	89.4	61.9	0.790
Avg-Pool	88.6	59.7	0.686	92.2	77.4	0.841
Central	90.9	62.1	0.722	93.5	78.9	0.864
Ours	**94.4**	**64.1**	**0.776**	**94.7**	**79.3**	**0.896**

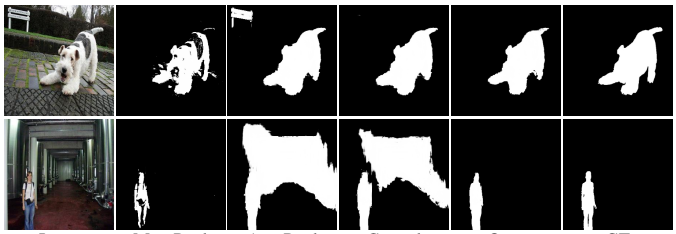


Fig. 8. **Visual Results of Different Aggregation Functions.** We can see that while Max-Pool tends to suffer from under-detection, Avg-Pool tends to suffer from over-detection due to its equal weighting over the whole scene. The central-prior assumption often works well, *e.g.*, the dog in the first row, but may produce false positives around the center region of the image, *e.g.*, the second row. In contrast, our weighted sum solution generally produces more accurate predictions.

The Effectiveness of the FRC Module. Our pixel decoder comprises four FRC modules, which are proposed to align the high-level features with low-level features in an unsupervised manner. We study the effectiveness of the proposed FRC module by replacing our pixel decoder with a standard FPN [61] decoder. Besides, we also compare it with a baseline, where no pixel decoders are used. For a better understanding of how well the spatial details are recovered, we do not apply CRFs [59] as post-processing for these models.

As shown in Table V, our FRC decoder (*i.e.*, FRC (Ours)) outperforms the baseline (*i.e.*, None) by a significant margin, achieving a 10% improvement in accraucy and a remarkable 48% improvement in F_β on *DUTS-TE*. Besides, when compared to the FPN decoder method, our FRC decoder achieves a consistent improvement with all evaluation metrics. Moreover, we plot the accuracy and F_β curves in Figure 9 to understand the performance gain after each training epoch. We observe that the proposed FRC module accelerates convergence and helps improve the final performance. Specifically, our pixel decoder achieves accuracy/F_β of 89.3/0.623 after the first epoch (*i.e.*, 660 iterations), while FPN [61] achieves accuracy/F_β of 84.3/0.538. The above suggests that our FRC modules contribute to faster improvement and better performances.

The Impact of the Saliency Bank. We enhance our CSE module with a saliency bank, and implement it as a regularization term, as depicted in Eq 6. We conduct experiments to understand the impact of using saliency bank by comparing the performance of our model with and without the saliency bank. As shown in Figure 10, we observe a significant improvement in performance during the early stage when enabling the saliency bank, which also greatly accelerates the convergence. Additionally, we investigate the effect of varying the size of the saliency bank. We observe that the performance gain in

TABLE V
THE EFFECTIVENESS OF THE FRC MODULE. CRF [59] IS EXCLUDED TO BETTER ASSESS SPATIAL DETAIL RECOVERY. NONE: NO PIXEL DECODER. FPN: USES FEATURE PYRAMID NETWORKS [61]. FRC (OURS): STACKS FOUR FRC MODULES AS THE PIXEL DECODER.

Pixel Decoder	DUTS-TE [63]			ECSSD [64]		
	ACC↑	IoU↑	F_β ↑	ACC↑	IoU↑	F_β ↑
None	85.3	43.9	0.515	87.4	60.5	0.693
FPN [61]	93.7	61.1	0.750	94.2	77.1	0.882
FRC (Ours)	**93.9**	**62.5**	**0.761**	**94.2**	**77.4**	**0.885**

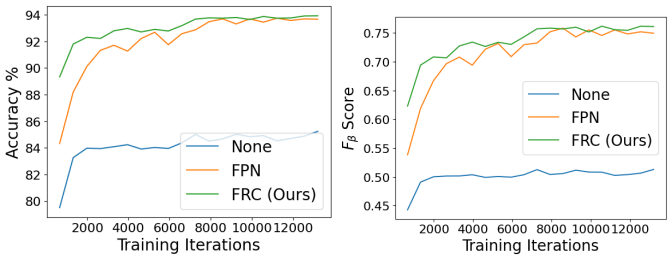


Fig. 9. **Accuracy/F_β Curves of Different Pixel Decoders.** We examine the accuracy and F_β scores after each epoch. Our FRC module (in green) demonstrates faster improvement compared to the Feature Pyramid Networks (FPNs) [61] (in orange) and using no pixel decoders (in blue). The left figure shows the accuracy curve, and the right figure shows the F_β curve. Each epoch contains 660 iterations.

the early stage is small if the size of the saliency bank is very small, *e.g.*, m=1. However, when m exceeds 64, the behaviors of the models become similar. We hypothesize that the regularization effect becomes overly strong when m is too small, thereby hindering the convergence process.

The Effectiveness of Different Loss Components. Since our model is a self-supervised model with no external supervisions, we would like to study how different loss components contribute to the detection of saliency and the enhancement of performances. As shown in Table VI, we first analyze the necessity of the instance discriminative task by removing the contrastive loss, *i.e.*, ID-1. The results show that the model becomes corrupted, and we find that the predictions are all zeros. We suppose that the learnable parameters in the CSE module cannot be optimized in an effective way when ℓ_{CL} is disable, leading to corrupted outputs.

We then experiment with removing all learnable parameters in the CSE module and disabling both ℓ_{CL} and ℓ_{bank}, *i.e.*, ID-2. We further remove ℓ_{LAT} from ID-2 to form ID-3. The results show that the ID-2 model is corrupted, but the ID-3 model is not corrupted. The distinction between ID-2 and ID-3 is that the ID-3 model omits the LAT loss, suggesting that the LAT loss can lead to model corruption if excluding ℓ_{CL} and ℓ_{bank}. We believe that the ID-3 model is not corrupted because of the placement of the pixel decoder between the predictions and the attention maps, effectively impeding the corruption process. However, the ID-3 model is much worse than the other versions that include the instance discriminative task, including ID-4/5/6 and Ours.

We next analyze the effectiveness of pixel-level loss components, ℓ_{bce}, ℓ_{LAT} and ℓ_{cons}. Initially, we disable all pixel-level loss components, *i.e.*, ID-4. Since the ID-4 model does not rely on the pixel-level decoding process, we evaluate the attention maps with CRFs as post-processing for the

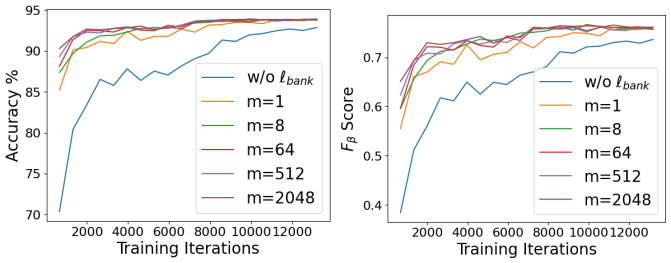


Fig. 10. **Accelerating Convergence with the Saliency Bank.** We investigate the impact of incorporating a saliency bank during training. Various sizes of the saliency bank are also examined. Our models with the saliency bank, denoted as $m = 1/8/64/512/2048$, exhibit faster improvement compared to the version without it (in blue). When m exceeds 64 (where m is the size of the saliency bank), we observe minor differences in the convergence curves. For clarity, the above models do not use CRFs [59] for post-processing. The left figure shows the accuracy curve, and the right figure shows the F_β curve. Each epoch contains 660 iterations.

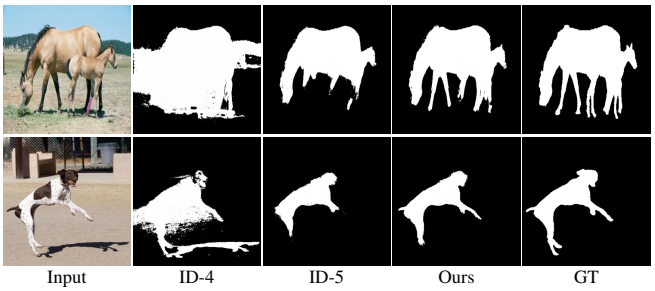


Fig. 11. **Visual Analysis on the Pixel Decoder and LAT Loss.** The ID-4 model lacks a pixel decoder and disables the LAT loss. The ID-5 model only omits the LAT loss. Our model, which incorporates both pixel decoder and LAT loss, produces better segmentation results. The improvement is particularly obvious near object boundaries. This demonstrates the effectiveness of our pixel decoder and LAT loss in recovering spatial details.

ID-4 model. Subsequently, we disable ℓ_{LAT} and ℓ_{cons} to produce ID-5 and ID-6, respectively. Our results show that the pixel-decoder with the LAT loss, *i.e.*, ID-6, can greatly improve the segmentation quality compared to ID-4 and ID-5. Specifically, ID-6 improves the accuracy/IoU scores by 5.7%/13.5% when compared to ID-4. The LAT loss also shows a 5.4% improvement in the IoU score when compared to Ours and ID-5.

Finally, we conduct a visual analysis between our model and ID-4 & ID-5, as shown in Figure 11. It indicates that the pixel decoder is essential in producing a fine-grained segmentation, and the LAT loss contributes to predicting sharper saliency masks. For example, the ID-4 model can only output a coarse salient object location and fails to outline object boundaries, while ID-5, which omits the LAT loss, can segment salient objects but is worse than our model, which is also revealed in the accuracy, IoU and F_β scores in Table VI.

Limitations. Our method does have limitations. Figure 12 shows that our model cannot segment small and thin objects well. This occurs despite our incorporation of the FRC module and the use of Conditional Random Fields [59] as a post-processing step. As a future work, we would like to develop a more precise salient object detector that is capable of handling diverse scenes and identifying objects of varying sizes.

TABLE VI
THE EFFECTIVENESS OF DIFFERENT LOSS COMPONENTS. WE STUDY THE EFFECTIVENESS OF DIFFERENT LOSS COMPONENTS BY REMOVING SOME OF THEM AT A TIME. A ✓ INDICATES THAT THE CORRESPONDING LOSS IS ENABLE, WHILE A × INDICATES THAT THE CORRESPONDING LOSS COMPONENT IS NOT USED. A ⋆ DENOTES THAT ALL LEARNABLE PARAMETERS OF THE CSE MODULE ARE REMOVED.

Method	ℓ_{CL}	ℓ_{bank}	ℓ_{bce}	ℓ_{LAT}	ℓ_{cons}	DUTS-TE [63]			ECSSD [64]		
						ACC↑	IoU↑	F_β ↑	ACC↑	IoU↑	F_β ↑
Ours	✓	✓	✓	✓	✓	**94.4**	**64.1**	**0.776**	**94.7**	**79.3**	**0.896**
ID-1	×	✓	✓	✓	✓	85.2	0.0	0.0	76.5	0.0	0.0
ID-2⋆	×	×	✓	✓	✓	85.2	0.0	0.0	76.5	0.0	0.0
ID-3⋆	×	×	✓	×	✓	89.7	36.2	0.509	85.4	45.1	0.608
ID-4	✓	✓	×	×	×	89.3	56.2	0.656	91.2	70.8	0.805
ID-5	✓	✓	✓	×	✓	93.6	60.8	0.748	93.9	75.9	0.865
ID-6	✓	✓	✓	✓	×	**94.4**	**63.8**	**0.777**	**94.2**	**77.4**	**0.886**

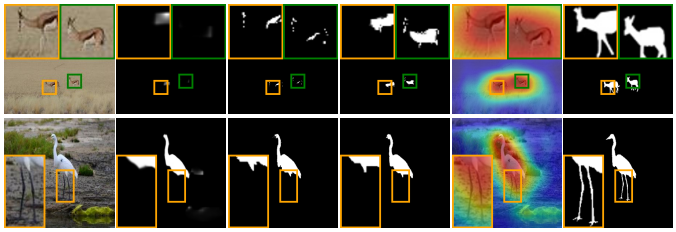


Fig. 12. **Failure Cases.** Our model fails to segment the legs of the antelopes in the 1^{st} row, and the thin legs of the stork in the 2^{nd} row. $Attn.$ is the attention map by our CSE module.

V. CONCLUSION

In this work, we have proposed a novel approach for unsupervised salient object detection under a contrastive learning framework. We have presented the idea with CSNet, a self-supervised salient object detection method. CSNet comprises two key components: the CSE module, which mimics the human attention mechanism to extract high-level saliency by performing an instance discriminative task, and the FRC module, which aids in fine-grained saliency mask prediction. In addition, we have also proposed a LAT loss to enhance the training of CSNet. We have conducted extensive evaluations on our approach and the proposed components. The experimental results demonstrate the superiority of our approach, establishing a new state-of-the-art in the field of unsupervised salient object detection.

ACKNOWLEDGMENTS

This work is in part supported by two GRF grants from the Research Grants Council of Hong Kong (Ref.: 11220724 and 11211223) and an Adobe Gift.

REFERENCES

[1] W. Zhu, S. Liang, Y. Wei, and J. Sun, "Saliency optimization from robust background detection," in *CVPR*, 2014.
[2] Y. Wang, W. Zhang, L. Wang, T. Liu, and H. Lu, "Multi-source uncertainty mining for deep unsupervised saliency detection," in *CVPR*, 2022.
[3] O. Siméoni, C. Sekkat, G. Puy, A. Vobeckỳ, É. Zablocki, and P. Pérez, "Unsupervised object localization: Observing the background to discover objects," in *CVPR*, 2023.
[4] W. Diao, X. Sun, X. Zheng, F. Dou, H. Wang, and K. Fu, "Efficient saliency-based object detection in remote sensing images using deep belief networks," *IEEE Geosci. Remote. Sens. Lett.*, vol. 13, no. 2, pp. 137–141, 2016.
[5] W. Wang, J. Shen, R. Yang, and F. Porikli, "Saliency-aware video object segmentation," *IEEE TPAMI*, 2018.
[6] L. Hoyer, M. Munoz, P. Katiyar, A. Khoreva, and V. Fischer, "Grid saliency for context explanations of semantic segmentation," in *NeurIPS*, 2019.
[7] S. Lee, M. Lee, J. Lee, and H. Shim, "Railroad is not a train: Saliency as pseudo-pixel supervision for weakly supervised semantic segmentation," in *CVPR*, 2021.
[8] W. V. Gansbeke, S. Vandenhende, S. Georgoulis, and L. V. Gool, "Unsupervised semantic segmentation by contrasting object mask proposals," in *ICCV*, 2021.
[9] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *ICML*, 2015.
[10] J. Zhao, J. Liu, D. Fan, Y. Cao, J. Yang, and M. Cheng, "Egnet: Edge guidance network for salient object detection," in *ICCV*, 2019.
[11] Z. Chen, Q. Xu, R. Cong, and Q. Huang, "Global context-aware progressive aggregation network for salient object detection," in *AAAI*, 2020.
[12] J. Wei, S. Wang, and Q. Huang, "F^{3}net: Fusion, feedback and focus for salient object detection," in *AAAI*, 2020.
[13] J. Wei, S. Wang, Z. Wu, C. Su, Q. Huang, and Q. Tian, "Label decoupling framework for salient object detection," in *CVPR*, 2020.
[14] X. Zhao, Y. Pang, L. Zhang, H. Lu, and L. Zhang, "Suppress and balance: A simple gated network for salient object detection," in *ECCV*, 2020.
[15] M. Ma, C. Xia, and J. Li, "Pyramidal feature shrinking for salient object detection," in *AAAI*, 2021.
[16] N. Liu, N. Zhang, K. Wan, L. Shao, and J. Han, "Visual saliency transformer," in *ICCV*, 2021.
[17] Y. Wang, R. Wang, X. Fan, T. Wang, and X. He, "Pixels, regions, and objects: Multiple enhancement for salient object detection," in *CVPR*, 2023.
[18] M. Zhuge, D.-P. Fan, N. Liu, D. Zhang, D. Xu, and L. Shao, "Salient object detection via integrity learning," *IEEE TPAMI*, 2023.
[19] Z. Luo, N. Liu, W. Zhao, X. Yang, D. Zhang, D.-P. Fan, F. Khan, and J. Han, "Vscode: General visual salient and camouflaged object detection with 2d prompt learning," in *CVPR*, 2024.
[20] Y. Mao, J. Zhang, Z. Wan, X. Tian, A. Li, Y. Lv, and Y. Dai, "Generative transformer for accurate and reliable salient object detection," *IEEE TCSVT*, 2024.
[21] Y. Piao, W. Wu, M. Zhang, Y. Jiang, and H. Lu, "Noise-sensitive adversarial learning for weakly supervised salient object detection," *IEEE Transactions on Multimedia*, 2023.
[22] Y. Wang, R. Wang, X. He, C. Lin, T. Wang, Q. Jia, and X. Fan, "Wbnet: Weakly-supervised salient object detection via scribble and pseudo-background priors," *Pattern Recognition*, 2024.
[23] Y. Wei, F. Wen, W. Zhu, and J. Sun, "Geodesic saliency using background priors," in *ECCV*, 2012.
[24] M. Cheng, J. Warrell, W. Lin, S. Zheng, V. Vineet, and N. T. Crook, "Efficient salient region detection with soft image abstraction," in *ICCV*, 2013.
[25] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *CVPR*, 2013.
[26] X. Li, H. Lu, L. Zhang, X. Ruan, and M. Yang, "Saliency detection via dense and sparse reconstruction," in *ICCV*, 2013.
[27] B. Jiang, L. Zhang, H. Lu, C. Yang, and M. Yang, "Saliency detection via absorbing markov chain," in *ICCV*, 2013.
[28] D. Zhang, J. Han, and Y. Zhang, "Supervision by fusion: Towards unsupervised learning of deep salient object detector," in *ICCV*, 2017.
[29] J. Zhang, T. Zhang, Y. Dai, M. Harandi, and R. Hartley, "Deep unsupervised saliency detection: A multiple noisy labeling perspective," in *CVPR*, 2018.

[30] T. Nguyen, M. Dax, C. Mummadi, N. Ngo, T. Nguyen, Z. Lou, and T. Brox, "Deepusps: Deep robust unsupervised saliency prediction via self-supervision," in *NeurIPS*, 2019.
[31] J. Zhang, J. Xie, and N. Barnes, "Learning noise-aware encoder-decoder from noisy labels by alternating back-propagation for saliency detection," in *ECCV*, 2020.
[32] J. Zhang, Y. Dai, T. Zhang, M. Harandi, N. Barnes, and R. Hartley, "Learning saliency from single noisy labelling: A robust model fitting perspective," *IEEE TPAMI*, 2021.
[33] X. Lin, Z. Wu, G. Chen, G. Li, and Y. Yu, "A causal debiasing framework for unsupervised salient object detection," in *AAAI*, 2022.
[34] O. Siméoni, G. Puy, H. Vo, S. Roburin, S. Gidaris, A. Bursuc, P. Pérez, R. Marlet, and J. Ponce, "Localizing objects with self-supervised transformers and no labels," in *BMVC 2021*.
[35] Y. Wang, X. Shen, S. Hu, Y. Yuan, J. Crowley, and D. Vaufreydaz, "Self-supervised transformers for unsupervised object discovery using normalized cut," in *CVPR*, 2022.
[36] X. Wang, R. Girdhar, S. Yu, and I. Misra, "Cut and learn for unsupervised object detection and instance segmentation," in *CVPR*, 2023.
[37] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *ICCV*, 2021.
[38] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jégou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "Dinov2: Learning robust visual features without supervision," *arXiv:2304.07193*, 2023.
[39] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," *arXiv:2003.04297*, 2020.
[40] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*, 2020.
[41] R. Achanta, S. Hemami, F. Estrada, and S. Süsstrunk, "Frequency-tuned salient region detection," in *CVPR*, 2009.
[42] D.-P. Fan, Z. Lin, Z. Zhang, M. Zhu, and M.-M. Cheng, "Rethinking rgb-d salient object detection: Models, data sets, and large-scale benchmarks," *IEEE Transactions on neural networks and learning systems*, 2020.
[43] D.-P. Fan, Y. Zhai, A. Borji, J. Yang, and L. Shao, "Bbs-net: Rgb-d salient object detection with a bifurcated backbone strategy network," in *ECCV*, 2020.
[44] D.-P. Fan, W. Wang, M.-M. Cheng, and J. Shen, "Shifting more attention to video salient object detection," in *CVPR*, 2019.
[45] D.-P. Fan, M.-M. Cheng, J.-J. Liu, S.-H. Gao, Q. Hou, and A. Borji, "Salient objects in clutter: Bringing salient object detection to the foreground," in *ECCV*, 2018.
[46] D.-P. Fan, J. Zhang, G. Xu, M.-M. Cheng, and L. Shao, "Salient objects in clutter," *IEEE TPAMI*, 2022.
[47] S. Yu, B. Zhang, J. Xiao, and E. G. Lim, "Structure-consistent weakly supervised salient object detection with local saliency coherence," in *AAAI*, 2021.
[48] J. Zhang, X. Yu, A. Li, P. Song, B. Liu, and Y. Dai, "Weakly-supervised salient object detection via scribble annotations," in *CVPR*, 2020.
[49] Y. Piao, J. Wang, M. Zhang, and H. Lu, "Mfnet: Multi-filter directive network for weakly supervised salient object detection," in *ICCV*, 2021.
[50] Y. Zeng, Y. Zhuge, H. Lu, L. Zhang, M. Qian, and Y. Yu, "Multi-source weak supervision for saliency detection," in *CVPR*, 2019.
[51] G. Li, Y. Xie, and L. Lin, "Weakly supervised salient object detection using image labels," in *AAAI*, 2018.
[52] S. Gao, H. Xing, W. Zhang, Y. Wang, Q. Guo, and W. Zhang, "Weakly supervised video salient object detection via point supervision," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 3656–3665.
[53] R. Yasarla, R. Weng, W. Choi, V. M. Patel, and A. Sadeghian, "3sd: Self-supervised saliency detection with no labels," in *WACV*, 2024.
[54] H. Zhou, P. Chen, L. Yang, X. Xie, and J. Lai, "Activation to saliency: Forming high-quality labels for unsupervised salient object detection," *IEEE TCSVT*, 2023.
[55] H. Zhou, B. Qiao, L. Yang, J. Lai, and X. Xie, "Texture-guided saliency distilling for unsupervised salient object detection," in *CVPR*, 2023.
[56] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *CVPR*, 2016.
[57] X. Zhang, J. Xie, Y. Yuan, M. Mi, and R. Tan, "Heap: Unsupervised object discovery and localization with contrastive grouping," *AAAI*, 2024.
[58] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017.
[59] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," in *NeurIPS*, 2011.
[60] S. Huang, Z. Lu, R. Cheng, and C. He, "Fapn: Feature-aligned pyramid network for dense image prediction," in *ICCV*, 2021.
[61] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *CVPR*, 2017.
[62] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *CVPR*, 2015.
[63] L. Wang, H. Lu, Y. Wang, M. Feng, D. Wang, B. Yin, and X. Ruan, "Learning to detect salient objects with image-level supervision," in *CVPR*, 2017.
[64] J. Shi, Q. Yan, L. Xu, and J. Jia, "Hierarchical image saliency detection on extended CSSD," *IEEE TPAMI*, 2016.
[65] P. Yan, Z. Wu, M. Liu, K. Zeng, L. Lin, and G. Li, "Unsupervised domain adaptive salient object detection through uncertainty-aware pseudo-label learning," in *AAAI*, 2022.
[66] G. Shin, S. Albanie, and W. Xie, "Unsupervised salient object detection with spectral cluster voting," in *CVPR Workshops*, 2022.
[67] G. Li and Y. Yu, "Visual saliency based on multiscale deep features," in *CVPR*, 2015.
[68] Y. Li, X. Hou, C. Koch, J. Rehg, and A. Yuille, "The secrets of salient object segmentation," in *CVPR*, 2014.
[69] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *CVPR*, 2013.
[70] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.
[71] D.-P. Fan, M.-M. Cheng, Y. Liu, T. Li, and A. Borji, "Structure-measure: A new way to evaluate foreground maps," in *ICCV*, 2017.
[72] D. Fan, C. Gong, Y. Cao, B. Ren, M. Cheng, and A. Borji, "Enhanced-alignment measure for binary foreground map evaluation," in *IJCAI*, 2018.
[73] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *ICCV*, 2021.
[74] X. Qin, Z. Zhang, C. Huang, M. Dehghan, O. R. Zaiane, and M. Jagersand, "U2-net: Going deeper with nested u-structure for salient object detection," *Pattern Recognition*, 2020.

VI. Authors' Brief Biographies

Huankang Guan is a Ph.D. candidate at City University of Hong Kong. He obtained a B.Eng from Wuhan University in 2020. His research interests include deep learning, computer vision and their practical applications.

Jiaying Lin received his PhD degree in Computer Science from the City University of Hong Kong. He holds a B.Eng. degree in Computer Science and Technology from South China University and Technology. His research focuses on computer vision and computer graphics. He actively contributes as a program committee member and reviewer for prestigious conferences and journals, including CVPR, ECCV, IEEE Transactions on Image Processing and IEEE Transactions on Circuits and Systems for Video Technology.

Rynson W.H. Lau received his Ph.D. degree from the University of Cambridge. He has been on the faculty of Durham University, City University of Hong Kong, and The Hong Kong Polytechnic University. He is an Honorary Professor at Swansea University, United Kingdom.

He currently serves on the Editorial Boards of the International Journal of Computer Vision (IJCV) and IET Computer Vision.

He served as the Guest Editor of a number of journal special issues, including ACM Trans. on Internet Technology, IEEE Trans. on Multimedia, IEEE Trans. on Visualization and Computer Graphics, and IEEE Computer Graphics Applications. He also served on the committee of a number of conferences, including Program Co-chair of ACM VRST 2004, ACM MTDL 2009, IEEE U-Media 2010, and Conference Co-chair of CASA 2005, ACM VRST 2005, ACM MDI 2009, ACM VRST 2014. Rynson's research interests include computer graphics and computer vision.