Unified Unsupervised Salient Object Detection via Knowledge Transfer

Yao Yuan, Wutao Liu, Pan Gao*, Qun Dai and Jie Qin*
Nanjing University of Aeronautics and Astronautics
{ayews233, wutaoliu, pan.gao, daiqun}@nuaa.edu.cn, qinjiebuaa@gmail.com

Abstract
Recently, unsupervised salient object detection (USOD) has gained increasing attention due to its annotation-free nature. However, current methods mainly focus on specific tasks such as RGB and RGB-D, neglecting the potential for task migration. In this paper, we propose a unified USOD framework for generic USOD tasks. Firstly, we propose a Progressive Curriculum Learning-based Saliency Distilling (PCL-SD) mechanism to extract saliency cues from a pre-trained deep network. This mechanism starts with easy samples and progressively moves towards harder ones, to avoid initial interference caused by hard samples. Afterwards, the obtained saliency cues are utilized to train a saliency detector, and we employ a Self-rectify Pseudo-label Refinement (SPR) mechanism to improve the quality of pseudo-labels. Finally, an adapter-tuning method is devised to transfer the acquired saliency knowledge, leveraging shared knowledge to attain superior transferring performance on the target tasks. Extensive experiments on five representative SOD tasks confirm the effectiveness and feasibility of our proposed method. Code and supplement materials are available at https://github.com/I2-Multimedia-Lab/A2S-v3.

1 Introduction
Salient object detection (SOD) aims to identify the most visually significant objects in images. Supervised SOD methods have achieved excellent results, but due to their heavy reliance on pixel-level annotations for salient objects, unsupervised SOD (USOD) has been gaining increasing attention. USOD not only eliminates the need for annotated data but also exhibits strong generalization performance when applied to other tasks [Niu et al., 2021; Wu et al., 2021].

Traditional SOD methods rely heavily on hand-crafted features, such as color and contrast, for saliency extraction. Although these methods prove effective for simple scenes, they encounter difficulties in complex scenes due to the absence of high-level semantic information. Existing deep learning-based USOD methods [Nguyen et al., 2019; Zhang et al., 2018] mostly utilize the predictions generated by traditional SOD methods as saliency cues and incorporate semantic information to generate refined saliency predictions. Recently, based on the observation that CNNs pre-trained on large-scale data usually produce high activations on some primary objects, A2S [Zhou et al., 2023a] have developed a method to distill saliency from the activation maps of deep networks and generate high-quality pseudo-labels. However, we found that during the initial training phase, the presence of hard samples in complex scenes or along object boundaries results in the accumulation of irreparable errors.

Unsupervised SOD is generally considered to exhibit strong generalization and transferability due to its annotation-free nature. However, prevailing USOD methodologies predominantly focus on the Natural Still Image (NSI) domain, exemplified by RGB, RGB-D, and RGB-T. Consequently, USOD on non-NSI domain, encompassing video SOD and Remote Sensing Image (RSI) SOD, remains largely unexplored, presenting a notable research gap in the field. We believe that different SOD tasks share common knowledge, and exploiting this shared knowledge can benefit transfer performance. On the other hand, compared to NSI SOD, the available datasets for video SOD or RSI SOD are relatively small...
and burdensome to obtain. As a result, training models from scratch on these tasks to obtain satisfactory performance is currently deemed impractical. Therefore, we advocate for the investigation of a more universally applicable unsupervised saliency knowledge transfer method.

To address the aforementioned issues, we design a unified framework for generic unsupervised SOD tasks. Firstly, we propose the Progressive Curriculum Learning-based Saliency Distilling (PCL-SD) mechanism to guide the extraction of saliency cues. At the early stages of training, we only extract preliminary saliency cues from easy samples. As the training progresses, we progressively incorporate hard samples to mine deeper saliency knowledge. The employment of PCL-SD effectively mitigates the initial accumulation of errors, leading to a more stable and robust training process. Next, we utilize the obtained saliency cues to train a saliency detector and design a Self-rectify Pseudo-label Refinement (SPR) mechanism to improve the quality of pseudo-labels.

On one hand, the proposed SPR employs the saliency predictions of the model during training to rectify incorrect predictions within the pseudo-labels. On the other hand, it incorporates the prior knowledge of the input image to prevent the model from becoming complacent. The SPR mechanism demonstrates a strong capability in self-supervised learning, resulting in improved pseudo-label quality. Finally, we devise an adapter-tuning method to transfer the acquired saliency knowledge to non-NSI SOD tasks, such as video SOD and RSI SOD. Specifically, we selectively fine-tune the deep features, ensuring effective adaptation of the model to the target task while mitigating the risk of model degradation.

Our main contributions can be summarized as follows:

- We propose the Progressive Curriculum Learning-based Saliency Distilling (PCL-SD) mechanism to extract saliency cues from easy samples to hard ones.
- We design the Self-rectify Pseudo-label Refinement (SPR) mechanism to gradually improve the quality of pseudo-labels during the training process.
- We devise an adapter-tuning method to transfer saliency knowledge from NSI SOD to non-NSI SOD tasks, achieving impressive transfer performance.

Note that we are the first to consider knowledge transfer from NSI domain to non-NSI domain, and develop a unified framework for generic USOD tasks. Experiments on RGB, RGB-D, RGB-T, video SOD and RSI SOD benchmarks confirm the state-of-the-art USOD performance of our method.

2 Proposed Method

3 Related Works

2.1 Unsupervised Salient Object Detection

Traditional SOD methods rely on hand-crafted features to extract saliency cues. For instance, [Perazzi et al., 2012] estimates saliency by evaluating the contrast in uniqueness and spatial distribution within the image. [Jiang et al., 2011] employ a combination of bottom-up salient stimuli and object-level shape prior to segment salient objects. Although these approaches perform well in simple scenes, they face challenges in handling complex scenes due to the lack of high-level semantic information.

Existing deep learning-based methods for USOD typically involve two stages. In the first stage, pseudo-labels are obtained, while in the second stage, a network is trained using these pseudo-labels. For instance, [Zhang et al., 2017] fuses multiple noisy saliency cues to generate supervisory signals for training the deep salient object detector. In [Nguyen et al., 2019], a set of refinement networks are initially trained to enhance the quality of these saliency cues, and the refined pseudo-labels are subsequently utilized to train a saliency detector. A more recent approach, A2S [Zhou et al., 2023a], proposes a method to distill saliency from the activation maps of deep networks, achieving high-quality pseudo-labels.

2.2 Knowledge Transfer in SOD

Knowledge transfer involves applying models or features trained in one task or domain to another related task or domain. A typical example is fine-tuning a deep network that was pre-trained on large-scale data for a specific target task. However, the exploration of knowledge transfer across different SOD tasks remains insufficient. Among the limited studies, [Fu et al., 2022] addresses the RGB-D SOD task as a few-shot learning problem and enhances performance by incorporating knowledge from RGB SOD. [Zhou et al., 2023b] employs data from multiple SOD tasks to train a generalized saliency detector. Nevertheless, when extending to generic SOD tasks, the inherent gap between various SOD tasks can impede effective model training. Consequently, it becomes crucial to devise a knowledge transfer approach that is rooted in shared knowledge.

3 Proposed Method

Figure 2 illustrates the proposed two-stage framework. In stage 1, we train a saliency cue extractor (SCE) to transfer saliency knowledge from a pre-trained deep network. The proposed Progressive Curriculum Learning-based Saliency Distilling is employed to mitigate the initial accumulation of errors in training and ensure the stability and robustness of the training process. In stage 2, we utilize the obtained saliency cues as initial pseudo-labels to train a saliency detector (SD). CRF [Krähenbühl and Koltun, 2011] is adopted to enhance the initial pseudo-labels, and we employ the proposed Self-rectify Pseudo-label Refinement mechanism to improve pseudo-labels quality during the training process gradually.

Initially, we train our base model on Natural Still Image (NSI) SOD and subsequently transfer the model to non-NSI SOD tasks. Throughout the training of the base model, we combine all the NSI data for training. However, during the transfer process, we only employ task-specific data for training. For example, when migrating to video SOD, we solely utilize video frames and optical flow as input. The transfer process also follows a two-stage training approach, while we applied the proposed fine-tuning method to optimize the SCE instead of training it from scratch. Besides, ResNet-50 [He et al., 2016] pre-trained by MoCo-v2 [Chen et al., 2020], A2S [Zhou et al., 2023a] and MIDD [Tu et al., 2021] are employed as the pre-trained deep network, SCE, and SD, respectively. A more detailed description and explanation can be found in supplementary materials.
3.1 Progressive Curriculum Learning-based Saliency Distilling

The problem of obtaining saliency cues or extracting salient information from scratch has always been a challenge for unsupervised salient object detection methods. Earlier deep learning-based methods [Zhang et al., 2017] relied on noisy saliency cues generated by traditional SOD methods, while approaches like A2S [Zhou et al., 2023a] employ the activation maps produced by a pre-trained network as saliency cues. This method effectively extracts the saliency information embedded in the pre-trained network. However, at the early stages of training, hard samples in complex scenes may corrupt the fragile saliency patterns in the network, leading to irreparable accumulation errors and the risk of pattern collapse. To address this issue, we introduce the concept of curriculum learning into saliency distilling and propose Progressive Curriculum Learning-based Saliency Distilling (PCL-SD). As can be seen in Figure 3, the proposed PCL-SD rigidly excludes hard samples at the early stages of training and gradually incorporates them as training progresses. As a result, the model progressively extracts saliency knowledge from easy to hard samples, and the entire training process becomes more robust and stable.

The process of saliency distilling can be formulated as:

\[
\mathcal{L}_{\text{sali}} = 0.5 - \frac{1}{N} \sum_{i} |S(i) - 0.5| \tag{1}
\]

Here, \(N\) represents the number of pixels, and \(S(i)\) denotes pixel \(i\) in the saliency prediction \(S\) output by the saliency cue extractor (SCE). To be intuitively described, \(\mathcal{L}_{\text{sali}}\) pulls the predicted values of each pixel towards either 0 or 1. However, during the early stages of training, \(\mathcal{L}_{\text{sali}}\) may pull hard samples with values close to 0.5 in the wrong direction, which we refer to as the problem of error accumulation. The proposed PCL-SD strategy focuses on two essential aspects: (1) how to define hard samples, and (2) how to gradually incorporate them. Firstly, the determination of a pixel in the saliency prediction \(S\) as a hard sample is based on its prediction value. Specifically, a pixel \(S(i)\) is classified as a hard sample if

\[
|S(i) - 0.5| < p. \tag{2}
\]

Here, \(p\) is the threshold for dividing hard samples, with a larger \(p\) indicating more hard samples are divided. Secondly, the value of \(p\) is initially set as 0.2 and progressively decreased during the training process until all samples are included. This decrease is governed by the formula:

\[
p = \text{Max}(0, 0.2 - 0.6 \times \frac{E_c}{E_t}), \tag{3}
\]

where \(E_c\) and \(E_t\) denote current epoch and total epoch, respectively. Finally, we define PCL-SD as:

\[
M(i) = \begin{cases} 
0 & \text{if } |S(i) - 0.5| < p, \\
1 & \text{otherwise},
\end{cases}
\]

\[
\mathcal{L}_{\text{pcl-sali}} = 0.5 - \frac{1}{N} \sum_{i} |M(i) \odot S(i) - 0.5| \tag{4}
\]

where \(\odot\) denotes the Hadamard product for matrices.

3.2 Self-rectify Pseudo-label Refinement

Obtaining high-quality pseudo-labels is crucial for training a saliency detector (SD). As shown in Figure 4, the saliency prediction \(S\) output by SD can partially rectify errors within the pseudo-labels. We define saliency prediction as posterior rectification: \(R_{\text{post}} = S\). However, while this posterior rectification can rectify errors in initial pseudo-labels, it also introduces the risk of the model becoming overly confident and stagnant. To overcome this, we introduce prior information from the input image to optimize saliency prediction, in order to avoid the model falling into a self-complacent trap.

Previous approaches primarily rely on CRF for prior rectification, which entails significant computational costs. Inspired by [Ru et al., 2022], we employ a real-time pixel refiner to provide efficient prior rectification based on the input image. To start, let \(I\) and \(P\) represent the input image and position information, while \(\sigma_I\) and \(\sigma_P\) denote the standard deviation of feature values and position differences, respectively.
The parameters $\omega_1$ and $\omega_2$ control the smoothness. We define the feature distance $d_{f}^{ij}$ and position distance $d_{p}^{ij}$ between pixels as follows:

$$d_{f}^{ij} = -\frac{||I(i) - I(j)||}{\omega_1 \sigma_f}, d_{p}^{ij} = -\frac{||P(i) - P(j)||}{\omega_2 \sigma_p} \quad (5)$$

Then, the refiner $R(\cdot)$ is then defined as:

$$R(I) = \sum_{j \in \mathcal{N}(i)} \left( -\frac{\exp(d_{f}^{ij})}{\sum_{k \in \mathcal{N}(i)} \exp(d_{f}^{ik})} + \omega_3 \frac{\exp(d_{p}^{ij})}{\sum_{k \in \mathcal{N}(i)} \exp(d_{p}^{ik})} \right) \quad (6)$$

Here, $\mathcal{N}(\cdot)$ represents the set of neighboring pixels in an 8-way manner. Finally, the prior rectification can be defined as:

$$R_{pri} = R(I) \odot S \quad (7)$$

where $\odot$ denotes the Hadamard product for matrices. At last, the refined pseudo-label is defined as:

$$G_{ref} = \lambda_1 R_{ref} + \lambda_2 R_{post} + \lambda_3 G_{pre} \quad (8)$$

Here, $G_{ref}$ refers to the pseudo-labels after refinement, and $G_{pre}$ refers to the previous pseudo-labels. The introduction of $G_{ref}$ aims to improve the stability of the refinement process. $\lambda_1, \lambda_2,$ and $\lambda_3$ are empirically assigned as 0.2, 0.6, and 0.2, respectively. As shown in Figure 4, prior rectification has effectively compensated for the considerable loss of local details. The proposed SPR mechanism combines prior and posterior rectification, gradually improving the quality of pseudo labels during the training process, demonstrating strong self-supervised performance.

### 3.3 Knowledge Transfer to Non-NSI SOD Tasks

We investigate the transferability of the proposed method to video SOD and Remote Sensing Image (RSI) SOD. Figure 5 illustrates the varying degrees of relevance between different SOD tasks. The tasks within NSI SOD benefit from a greater amount of shared knowledge, allowing for the joint training of multiple tasks to achieve a better generalization performance. However, as we broaden our focus to generic SOD tasks, the inherent gap between tasks becomes the primary influencing factor. Joint training becomes more challenging and poses risks of model degradation. More discussions on this topic can be found in supplementary materials.

We posit that identifying an appropriate fine-tuning method can effectively address the issue of model degradation. Inspired by recent studies on Adapter-tuning [Houlsby et al., 2019], we design a simple but effective fine-tuning method for knowledge transfer from NSI SOD to non-NSI SOD tasks. Specifically, for end-to-end tasks in SOD, the prevailing methods and models employ the U-net [Ronneberger et al., 2015] structure and utilize multi-scale feature aggregation to achieve accurate saliency predictions. We contend that shallow features primarily contribute to local details and possess a degree of cross-task generality, while deep features play a pivotal role in salient object localization and exhibit task-specific characteristics. Hence, we suppose that fine-tuning solely the network layers or modules responsible for deep feature handling allows the model to adapt to the target task while circumventing degradation. Technically, we define the deep feature handling process in the model as

$$F = T(F) \quad (9)$$

Here, $F$ represents the deep features extracted by the backbone, $\hat{F}$ denotes the processed features, and $T$ signifies the network layer or module performing the processing. In specific end-to-end SOD models, $T$ can comprise a convolutional layer that modifies the number of feature channels or a network module that enhances the features. Our adapter-tuning approach can be defined as:

$$\hat{F} = T(F) + T_a(F) \quad (10)$$

In this equation, $T_a$ refers to the adapter, which possesses a structure consistent with $T$. Following the processing, $T_a$ is connected to the original network through a residual connection. During fine-tuning, we exclusively optimize the weights of $T_a$ while keeping the remaining weights of the model frozen. The detailed description can be found in supplementary materials. It is worth mentioning that this fine-tuning method is universal for any kind of SOD method or task.

### 3.4 Supervision Strategy

We initially train the saliency cue extractor (SCE) in the first stage, followed by training the saliency detector (SD) in the second stage. In the training of the first stage, we also incorporate Boundary-aware Texture Matching (BTM) [Zhou et al., 2023b] to introduce extra structural cues, and is defined
as $L_{sc}$. Moreover, a structural consistency loss is employed to achieve transformation-invariant predictions, and is formulated as:

$$L_{sc} = \sum_i ||S(i) - \hat{S}(i)||.$$  

(11)

Here, $\hat{S}$ denotes saliency prediction after transformation. To ensure training stability, only random scaling is adopted. The total loss for training SCE can be defined as:

$$L_{sce} = L_{pcl-sd} + \gamma L_{btm} + L_{sc}. $$  

(12)

$\gamma$ is empirically assigned as 0.05. We train the saliency detector (SD) with IoU loss, which is defined as:

$$L_{IoU} = 1 - \frac{\sum_i (S(i)G(i))}{\sum_i (S(i)+G(i)-S(i)G(i))}. $$  

(13)

$G$ refers to the pseudo-labels, and the total loss for training SD can be defined as:

$$L_{sd} = L_{IoU} + L_{sc}. $$  

(14)

4 Experiments

4.1 Implementation Details

Training Settings

The batch size is set to 8 and input images are resized to $320 \times 320$. Horizontal flipping is employed as our data augmentation. We train the saliency cue extractor for 20 epochs using the SGD optimizer with an initial learning rate of 0.1, which is decayed linearly. We train the saliency detector for 10 epochs using the SGD optimizer with a learning rate of 0.005. All experiments were implemented on a single RTX 3090 GPU.

Datasets

We follow the prevalent settings of SOD and relevant tasks. Here are some details about the datasets we used. (1) RGB SOD: We use the training subsets of DUTS [Wang et al., 2017] to train our method. ECSSD [Yan et al., 2013], PASCALS [Li et al., 2014], HKU-IS [Li and Yu, 2015], DUTSTE [Wang et al., 2017] and DUT-O [Yang et al., 2013] are employed for evaluation. (2) RGB-D SOD: We choose 2185 samples from the training subsets of NLPR [Peng et al., 2014] and NJUD [Ju et al., 2014] as the training set. RGBD135 [Cheng et al., 2014], SIP [Fan et al., 2020] and the testing subsets of NJUD and NLPR are employed for evaluation. (3) RGB-T SOD: 2500 images in VT5000 [Tu et al., 2022a] are for training, while VT1000 [Tu et al., 2019], VT821 [Wang et al., 2018] and the rest 2500 images in VT5000 are for testing. (4) Video SOD: We choose the training splits of DAVIS [Perazzi et al., 2016] and DAVSOD [Fan et al., 2019] to train our method. SegV2 [Li et al., 2013], FBMS [Brox and Malik, 2010] and the testing splits of DAVIS and DAVSOD are employed for evaluation. (5) Remote Sensing Image SOD: We choose the training splits of ORSSD [Li et al., 2019] and EORSSD [Zhang et al., 2020b] to train our method. The testing splits of ORSSD and EORSSD are employed for evaluation.

Metrics

We employ three metrics to evaluate our model and the existing state-of-the-art methods, including Mean Absolute Error $\bar{M}$, average F-measure ($F_{\beta}$) [Achanta et al., 2009] and E-measure ($E_\xi$) [Fan et al., 2018]. Specifically, $\bar{M}$ measures the average pixel-wise difference between the prediction $P$ and the ground truth $G$, and is calculated as $\bar{M} = \frac{1}{N} \sum_i |P(i) - G(i)|$. $F_{\beta}$ considers both precision and recall values of the prediction map, and can be computed as $F_{\beta} = (1+\beta^2) \times \frac{P \times \text{Recall}}{\beta^2 \times P + \text{Recall}}$, with $\beta^2$ set to 0.3. $E_\xi$ takes account the local pixel values along with the image-level mean value, and is defined as $E_\xi = \frac{1}{N} \sum_i \phi_\xi(i, j)$, where $\phi_\xi$ represents the enhanced alignment matrix.

4.2 Comparisons With State-of-the-Art

We report the performance of our method on five representative SOD tasks, and more qualitative results are provided in supplementary materials.

Results on RGB SOD

Table 1 presents a quantitative comparison between the proposed method and recent fully-supervised, weakly-supervised, and unsupervised methods. The fully-supervised methods include MINet [Pang et al., 2020] and VST [Liu et al., 2021], the weakly-supervised methods contain WSSA [Zhang et al., 2020a] and MFNet [Piao et al., 2021], and the unsupervised methods comprise SBF [Zhang et al., 2017], TSD [Zhou et al., 2023] and STC [Song et al., 2023]. Our results are presented under different settings: (1) Training our method using task-specific data, denoted as “Ours_t.s.”, for a fair comparison; (2) Training our method using NSI data, including RGB, RGB-D, and RGB-T datasets, referred to as “Ours”.

The results presented in Table 1 clearly indicate that the proposed method outperforms existing USOD methods, leading to significant improvements in performance. Additionally, our unsupervised approach demonstrates competitive performance in comparison to recent weakly-supervised and fully-supervised methods. Notably, our method, referred to as “Ours”, exhibits a slight superiority over “Ours_t.s.”. We suppose that this advantage stems from the utilization of a more extensive training dataset, which enhances the model’s generalization ability and leads to improved performance when applied to unseen images.

A qualitative comparison is presented in Figure 6. As can be seen, our method has achieved more accurate and complete saliency prediction. Moreover, our approach exhibits remarkable performance in dealing with multiple objects (row 2).

Results on RGB-D and RGB-T SOD

Table 2 and 3 present a comparison between the proposed method and recent methods on RGB-D and RGB-T benchmarks, respectively. For a fair comparison, we also train our method using task-specific data, denoted as “Ours_t.s.”, VST [Liu et al., 2021], CCFE [Liao et al., 2022], DSU [Li et al., 2022], TSD, MIDD [Tu et al., 2021] and SRS [Liu et al., 2023] are employed for comparison. Our method has achieved state-of-the-art performance on both RGB-D and
Table 1: Quantitative comparison on RGB SOD benchmarks. “Sup.” indicates the supervised signals used to train SOD methods. “F”, “W” and “U” mean fully-supervised, weakly-supervised and unsupervised, respectively. The best results are shown in **bold**.

<table>
<thead>
<tr>
<th>Method</th>
<th>Year</th>
<th>Sup.</th>
<th>M ↓</th>
<th>F</th>
<th>β</th>
<th>E</th>
<th>ξ</th>
<th>M ↓</th>
<th>F</th>
<th>β</th>
<th>E</th>
<th>ξ</th>
<th>M ↓</th>
<th>F</th>
<th>β</th>
<th>E</th>
<th>ξ</th>
<th>M ↓</th>
<th>F</th>
<th>β</th>
<th>E</th>
<th>ξ</th>
</tr>
</thead>
<tbody>
<tr>
<td>MINet</td>
<td>2020</td>
<td>F</td>
<td>.055</td>
<td>.756</td>
<td>.873</td>
<td>.037</td>
<td>.828</td>
<td>.917</td>
<td>.033</td>
<td>.924</td>
<td>.933</td>
<td>.028</td>
<td>.908</td>
<td>.961</td>
<td>.064</td>
<td>.842</td>
<td>.899</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VST</td>
<td>2021</td>
<td>F</td>
<td>.058</td>
<td>.756</td>
<td>.872</td>
<td>.037</td>
<td>.818</td>
<td>.916</td>
<td>.033</td>
<td>.92</td>
<td>.957</td>
<td>.029</td>
<td>.9</td>
<td>.96</td>
<td>.061</td>
<td>.829</td>
<td>.902</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WSSA</td>
<td>2020</td>
<td>W</td>
<td>.068</td>
<td>.703</td>
<td>.845</td>
<td>.062</td>
<td>.742</td>
<td>.869</td>
<td>.047</td>
<td>.86</td>
<td>.932</td>
<td>.059</td>
<td>.87</td>
<td>.917</td>
<td>.096</td>
<td>.785</td>
<td>.855</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MFNet</td>
<td>2021</td>
<td>W</td>
<td>.098</td>
<td>.621</td>
<td>.784</td>
<td>.079</td>
<td>.693</td>
<td>.891</td>
<td>.058</td>
<td>.839</td>
<td>.919</td>
<td>.084</td>
<td>.844</td>
<td>.889</td>
<td>.115</td>
<td>.756</td>
<td>.824</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSD</td>
<td>2023</td>
<td>U</td>
<td>.061</td>
<td>.745</td>
<td>.863</td>
<td>.047</td>
<td>.810</td>
<td>.901</td>
<td>.044</td>
<td>.916</td>
<td>.938</td>
<td>.037</td>
<td>.902</td>
<td>.947</td>
<td>.074</td>
<td>.830</td>
<td>.882</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STC</td>
<td>2023</td>
<td>U</td>
<td>.068</td>
<td>.753</td>
<td>.852</td>
<td>.052</td>
<td>.809</td>
<td>.891</td>
<td>.050</td>
<td>.903</td>
<td>.935</td>
<td>.041</td>
<td>.891</td>
<td>.942</td>
<td>.076</td>
<td>.827</td>
<td>.881</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>-</td>
<td>U</td>
<td>.063</td>
<td>.749</td>
<td>.864</td>
<td>.046</td>
<td>.814</td>
<td>.906</td>
<td>.038</td>
<td>.922</td>
<td>.95</td>
<td>.034</td>
<td>.905</td>
<td>.953</td>
<td>.068</td>
<td>.841</td>
<td>.898</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>-</td>
<td>U</td>
<td>.062</td>
<td>.759</td>
<td>.868</td>
<td>.047</td>
<td>.816</td>
<td>.906</td>
<td>.038</td>
<td>.923</td>
<td>.951</td>
<td>.033</td>
<td>.908</td>
<td>.954</td>
<td>.069</td>
<td>.844</td>
<td>.899</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Quantitative comparison on RGB-D SOD benchmarks.

<table>
<thead>
<tr>
<th>Method</th>
<th>Year</th>
<th>Sup.</th>
<th>M ↓</th>
<th>F</th>
<th>β</th>
<th>E</th>
<th>ξ</th>
<th>M ↓</th>
<th>F</th>
<th>β</th>
<th>E</th>
<th>ξ</th>
<th>M ↓</th>
<th>F</th>
<th>β</th>
<th>E</th>
<th>ξ</th>
<th>M ↓</th>
<th>F</th>
<th>β</th>
<th>E</th>
<th>ξ</th>
</tr>
</thead>
<tbody>
<tr>
<td>VST</td>
<td>2021</td>
<td>F</td>
<td>.017</td>
<td>.917</td>
<td>.979</td>
<td>.034</td>
<td>.899</td>
<td>.943</td>
<td>.023</td>
<td>.886</td>
<td>.956</td>
<td>.04</td>
<td>.895</td>
<td>.941</td>
<td>.047</td>
<td>.889</td>
<td>.923</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCFE</td>
<td>2022</td>
<td>F</td>
<td>.020</td>
<td>.911</td>
<td>.964</td>
<td>.032</td>
<td>.914</td>
<td>.953</td>
<td>.021</td>
<td>.907</td>
<td>.962</td>
<td>.047</td>
<td>.889</td>
<td>.923</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSU</td>
<td>2022</td>
<td>U</td>
<td>.061</td>
<td>.761</td>
<td>.895</td>
<td>.135</td>
<td>.719</td>
<td>.797</td>
<td>.065</td>
<td>.745</td>
<td>.879</td>
<td>.156</td>
<td>.619</td>
<td>.774</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSD</td>
<td>2023</td>
<td>U</td>
<td>.068</td>
<td>.753</td>
<td>.852</td>
<td>.052</td>
<td>.809</td>
<td>.891</td>
<td>.050</td>
<td>.903</td>
<td>.935</td>
<td>.041</td>
<td>.891</td>
<td>.942</td>
<td>.076</td>
<td>.827</td>
<td>.881</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>-</td>
<td>U</td>
<td>.063</td>
<td>.749</td>
<td>.864</td>
<td>.046</td>
<td>.814</td>
<td>.906</td>
<td>.038</td>
<td>.922</td>
<td>.95</td>
<td>.034</td>
<td>.905</td>
<td>.953</td>
<td>.068</td>
<td>.841</td>
<td>.898</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>-</td>
<td>U</td>
<td>.062</td>
<td>.759</td>
<td>.868</td>
<td>.047</td>
<td>.816</td>
<td>.906</td>
<td>.038</td>
<td>.923</td>
<td>.951</td>
<td>.033</td>
<td>.908</td>
<td>.954</td>
<td>.069</td>
<td>.844</td>
<td>.899</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Quantitative comparison on RGB-T SOD benchmarks.

<table>
<thead>
<tr>
<th>Method</th>
<th>Year</th>
<th>Sup.</th>
<th>M ↓</th>
<th>F</th>
<th>β</th>
<th>E</th>
<th>ξ</th>
<th>M ↓</th>
<th>F</th>
<th>β</th>
<th>E</th>
<th>ξ</th>
<th>M ↓</th>
<th>F</th>
<th>β</th>
<th>E</th>
<th>ξ</th>
<th>M ↓</th>
<th>F</th>
<th>β</th>
<th>E</th>
<th>ξ</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCFE</td>
<td>2022</td>
<td>F</td>
<td>.030</td>
<td>.859</td>
<td>.937</td>
<td>.018</td>
<td>.906</td>
<td>.963</td>
<td>.027</td>
<td>.857</td>
<td>.934</td>
<td>.04</td>
<td>.895</td>
<td>.941</td>
<td>.047</td>
<td>.889</td>
<td>.923</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SRS</td>
<td>2023</td>
<td>W</td>
<td>.042</td>
<td>.911</td>
<td>.964</td>
<td>.032</td>
<td>.914</td>
<td>.953</td>
<td>.021</td>
<td>.907</td>
<td>.962</td>
<td>.047</td>
<td>.889</td>
<td>.923</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSD</td>
<td>2023</td>
<td>U</td>
<td>.061</td>
<td>.761</td>
<td>.895</td>
<td>.135</td>
<td>.719</td>
<td>.797</td>
<td>.065</td>
<td>.745</td>
<td>.879</td>
<td>.156</td>
<td>.619</td>
<td>.774</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>-</td>
<td>U</td>
<td>.063</td>
<td>.749</td>
<td>.864</td>
<td>.046</td>
<td>.814</td>
<td>.906</td>
<td>.038</td>
<td>.922</td>
<td>.95</td>
<td>.034</td>
<td>.905</td>
<td>.953</td>
<td>.068</td>
<td>.841</td>
<td>.898</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>-</td>
<td>U</td>
<td>.062</td>
<td>.759</td>
<td>.868</td>
<td>.047</td>
<td>.816</td>
<td>.906</td>
<td>.038</td>
<td>.923</td>
<td>.951</td>
<td>.033</td>
<td>.908</td>
<td>.954</td>
<td>.069</td>
<td>.844</td>
<td>.899</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Quantitative comparison on video SOD benchmarks.
Table 5: Quantitative comparison on RSI SOD benchmarks.

<table>
<thead>
<tr>
<th>Method</th>
<th>RGB</th>
<th>RGB-D</th>
<th>RGB-T</th>
<th>video</th>
<th>RSI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>M ↓ F↑</td>
<td>M ↓ F↑</td>
<td>M ↓ F↑</td>
<td>M ↓ F↑</td>
<td>M ↓ F↑</td>
</tr>
<tr>
<td>Ours</td>
<td>.933 .928</td>
<td>.052 .854</td>
<td>.019 .949</td>
<td>- - -</td>
<td>- - -</td>
</tr>
<tr>
<td>Ours_f</td>
<td>.933 .927</td>
<td>.047 .87</td>
<td>.020 .953</td>
<td>.068 .696</td>
<td>.074 .634</td>
</tr>
</tbody>
</table>

Table 6: Evaluation on Pseudo-label Quality.

Table 8: Evaluation on Supervision Strategy. “RGB” denotes the training set of RGB SOD.

Table 7: Evaluation on Self-rectify Pseudo-label Refinement.

### Results on Video SOD and RSI SOD

Table 4, 5 present a comparison between the proposed method and recent methods on video SOD and RSI SOD benchmarks, respectively. STVS [Chen et al., 2021], WSVSD [Zhao et al., 2021], LVNet [Li et al., 2019], MJRB [Tu et al., 2022b] and TSD are employed for comparison. We consider video SOD and RSI SOD as two types of target transfer tasks. Thus, in the table, “Ours” represents zero-shot transfer results, while “Ours_f” refers to the outcomes obtained by fine-tuning the transferred model on the target task using our proposed knowledge transfer approach. Note that the transfer for video SOD and RSI SOD is conducted separately. Our model exhibits excellent adaptability to the target task following fine-tuning, and exhibits remarkable performance.

### 4.3 Ablation Study

**Evaluation on Pseudo-label Quality**

We assessed the quality of the pseudo-labels generated by models trained on different datasets. As previously mentioned, “Ours_s” denotes the model trained using task-specific data, whereas “Ours_f” refers to the model transferred to the target task. The results are presented in Table 6. In comparison to “Ours_s”, “Ours” exhibits slightly inferior performance on the RGB training set, but displays a notable improvement on the RGB-D and RGB-T training sets, which possess a comparatively limited amount of training data. This indicates that a larger training dataset yields superior model performance and enhanced generalization ability. Furthermore, “Ours” shows a slight improvement in video SOD, whereas it exhibits a substantial enhancement in RSI SOD. This indicates that video SOD and RSI SOD share more common knowledge, while RSI SOD requires greater fine-tuning and adaptation. More analysis on the adaptation to target tasks is presented in supplementary materials.

**Evaluation on SPR**

We evaluated the influence of various rectifications on the pseudo-labels, and the results are presented in Table 7. The posterior rectification \( R_{post} \) effectively corrects the erroneous predictions present in pseudo-labels, while the prior rectification \( R_{pri} \) adequately compensates for the lack of local details in pseudo-labels. As a result, the proposed SPR gradually enhances the quality of pseudo-labels, thereby improving the model’s performance.

**Evaluation on Supervision Strategy**

We evaluated the effectiveness of the proposed supervision strategy, as shown in Table 8. We treat all samples as easy samples to examine the effectiveness of PCL-SD. Upon applying PCL-SD, the model exhibits a slight improvement on the training set. Nonetheless, an impressive enhancement in performance can be observed on the test set. This improvement substantiates the model’s heightened generalization capability. Additionally, we explored the training of the saliency detector using different loss functions. The results indicate that the commonly employed binary cross-entropy (bce) in supervised SOD did not lead to effective performance enhancement. We hypothesize that this ineffectiveness may be attributed to the errors and interference stemming from incorrect predictions in pseudo-labels. In contrast, the self-supervised loss \( L_{ms} \) delivered a noteworthy improvement.

### 5 Conclusion

In this paper, we propose a two-stage unified unsupervised SOD framework for generic SOD tasks, with knowledge transfer as the foundation. Specifically, we introduce two innovative mechanisms: Progressive Curriculum Learning-based Saliency Distilling (PCL-SD) and Self-rectify Pseudo-label Refinement (SPR), which aim to extract saliency cues and optimize pseudo-labels. Additionally, we present a universal fine-tuning method to transfer the acquired saliency knowledge to generic SOD tasks. Extensive experiments on five representative SOD tasks validate the effectiveness and feasibility of our proposed method.
Acknowledgements
This work was partially supported by the National Natural Science Foundation of China (No. 62272227 & No. 62276129), and the Natural Science Foundation of Jiangsu Province (No. BK20220890).

References


