Transformer Fusion and Pixel-Level Contrastive Learning for RGB-D Salient Object Detection

Jiesheng Wu, *Student Member, IEEE*, Fangwei Hao, Weiyun Liang, *Student Member, IEEE*, and Jing Xu, *Member, IEEE*

***Abstract*—Current RGB-D salient object detection (RGB-D SOD) methods mainly develop a generalizable model trained by binary cross-entropy (BCE) loss based on convolutional or Transformer backbones. However, they usually exploit convolutional modules to fuse multi-modality features, with little attention paid to capturing the long-range multi-modality interactions for feature fusion. Furthermore, BCE loss does not explicitly explore intra- and inter-pixel relationships in a joint embedding space. To address these issues, we propose a cross-modality interaction parallel-transformer (CIPT) module, which better captures the long-range multi-modality interactions, generating more comprehensive fusion features. Besides, we propose a pixel-level contrastive learning (PCL) method that improves inter-pixel discrimination and intra-pixel compactness, resulting in a well-structured embedding space and a better saliency detector. Specifically, we propose an asymmetric network (TPCL) for RGB-D SOD, which consists of a Swin V2 Transformer-based backbone and a designed lightweight backbone (LDNet). Moreover, an edge-guided module and a feature enhancement (FE) module are proposed to refine the learned fusion features. Extensive experiments demonstrate that our method achieves excellent performance against 15 state-of-the-art methods on seven public datasets. We expect our work to facilitate the exploration of applying Transformer and contrastive learning for RGB-D SOD tasks.**

***Index Terms*—Multi-modality fusion, pixel-level contrastive learning, RGB-D salient object detection, transformer.**

I. Introduction

HUMANS are usually attracted to the most prominent object in an image and recognize it. To simulate this behavior, a computer vision task, i.e., salient object detection (SOD), has been defined [1], [2], [3], [4]. Generally, SOD is often used as a pre-processing operation for various computer vision tasks such as object recognition [5], image retrieval [6], editing [7], and object tracking [8]. Recent studies on RGB SOD have achieved significant progress. However, with the complexity and change of various scenes, it is challenging to detect salient objects only by relying on RGB modality. Meanwhile, with the applications of depth sensors and cameras, depth cues are gradually being introduced to localize salient objects. Therefore, recent researchers utilize depth modality to assist RGB modality in detecting salient objects, commonly referred to as RGB-D SOD [9].

The first exploration based on deep learning uses convolutional neural networks (CNNs) for RGB-D SOD [10]. Since then, numerous CNN-based methods have been proposed, and significant improvements have been achieved [11], [12], [13], [14], [15]. In the past two years, with the development of Vision Transformers (ViTs) [16], [17], some researchers have replaced CNN-based backbones with Transformer-based ones, and the performance has been further improved [18], [19], [20]. While significant improvement has been achieved with these methods, there are still two issues worth investigating:

(i) For RGB-D SOD, most of the current methods ignore capturing long-range multi-modality interactions to generate more comprehensive fusion features. Specifically, RGB-D SOD usually focuses on multi-modality interaction and fusion to generate more saliency-related fusion features [21], which can be divided into early fusion [14], [22], [23], middle fusion [12], [24], [25], and late fusion [26]. Moreover, existing methods usually employ channel attention (CA) [27], spatial attention (SA) [27], element-wise multiplication, and concatenation operations to capture multi-modality interactions, which may not sufficiently capture long-range interactions and generate comprehensive fusion features for RGB-D SOD. In addition, these methods usually use Transformer-based or CNN-based backbones for feature extraction, and they do not explore Transformer for multi-modality interaction and fusion. In fact, the powerful long-range multi-modality interaction ability of Transformer has been successfully explored in multiple multi-modality fusion fields (e.g., vision-language fusion tasks [28], [29]). Hence, the first issue is motivated: *how to capture long-range multi-modality interactions by employing Transformer between two modalities to generate more comprehensive fusion features for RGB-D SOD?*

(ii) Generally, a typical RGB-D SOD model consists of a deep encoder-decoder trained with pixel-level BCE loss. The BCE loss aims to classify each pixel into a specific class and does not explicitly explore the relationships between pixels. However, a robust RGB-D SOD model should be able to obtain highly discriminative pixel-level feature representations [30]

Manuscript received 7 December 2022; revised 23 March 2023 and 26 April 2023; accepted 3 May 2023. Date of publication 11 May 2023; date of current version 18 January 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62002177, and in part by the Natural Science Foundation of Tianjin City, China under Grants 21JCYBJC00110 and 19JCQNJC00300. The Associate Editor coordinating the review of this manuscript and approving it for publication was Prof. Sanghoon Lee. *(Corresponding author: Jing Xu.)*

The authors are with the College of Artificial Intelligence, Nankai University, Tianjin 300071, China (e-mail: jasonwu@mail.nankai.edu.cn; haofangwei@mail.nankai.edu.cn; weiyunliang@mail.nankai.edu.cn; xujing@nankai.edu.cn).

Our codes and predicted saliency maps will be released on GitHubhttps://github.com/TomorrowJW/TPCL_RGBDSOD*.

Digital Object Identifier 10.1109/TMM.2023.3275308

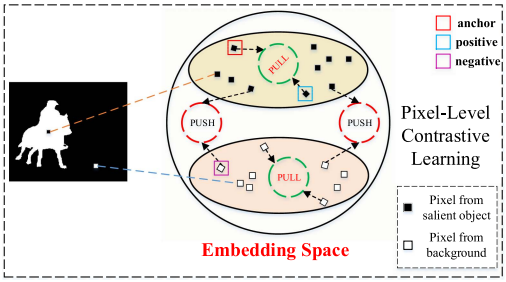
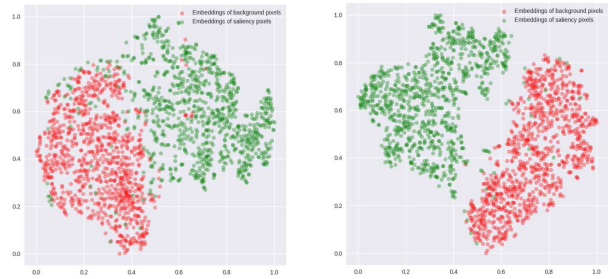


Fig. 1. Idea of the proposed pixel-level contrastive learning. Current RGB-D SOD models usually project salient or background pixels to an embedding space without accounting for the relationships between pixels (i.e., inter-pixel discrimination and intra-pixel compactness). Our proposed pixel-level contrastive loss solves the two shortcomings and enhances the performance of RGB-D SOD.



(a) without PCL (b) with PCL

Fig. 2. Representation distribution trained without/with the pixel-level contrastive loss on 929 labeled RGB-D images from the SIP dataset [14].

with the advantage of exploring pixel-to-pixel relationships. Specifically, an ideal RGB-D SOD model can obtain an embedding space that can effectively cluster the representations of pixels belonging to the same label while also increasing the representation distances between pixels belonging to different labels [31]. Such an embedding space can improve the performance of RGB-D SOD. Therefore, the second issue is motivated: *how to promote compactness within embeddings of saliency or background pixels while also encouraging discrimination between the saliency embeddings and background embeddings. In other words, how to encourage strong relationships between embeddings of intra-pixel and distinct relationships between embeddings of inter-pixel?*

To address the two issues, we propose two solutions. For the first issue, we propose a cross-modality interaction parallel-Transformer (CIPT) module consisting of two improved parallel Transformers. Different from the standard Transformer, long-range multi-modality interactions exist between the two Transformers of CIPT, aiming to capture the global dependencies between RGB and depth modalities to generate more comprehensive fusion features. Furthermore, we employ multiple CIPT modules at different scales and present a hierarchical multi-scale cross-modality fusion (HMCMF) module to learn contextual semantics. Overall, CIPT strives to integrate more multi-modality interactions into a holistic module to generate more comprehensive fusion features.

For the second issue, empirical studies confirm that contrastive learning (CL) [32], [33], [34] is a beneficial solution to match it. Specifically, we propose a novel label-based pixel-level contrastive learning (PCL) to explore intra- and inter-pixel relationships. Fig. 1 shows the workings of our proposed PCL. Our PCL loss aims to pull the embeddings between pixels of the same label closer while pushing away the embeddings between pixels of different labels in a joint embedding space. A PCL-trained model can learn a representation space that preserves intra-pixel compactness and inter-pixel discrimination. Here, we employ t-SNE [35] to show the distributions of learned fusion features without/with PCL loss in Fig. 2. Intuitively, the results demonstrate that PCL tackles the second issue. Furthermore, inspired by recent advances in SOD [18], [36], [37], we propose a lightweight LDNet and an edge-guided module in the stage of RGB-D SOD encoder (introduced later in Section III). In the decoding stage, we propose a simple progressive ladder decoder that fuses multi-stage features to output the final saliency maps. Based on the above insights and investigations, we propose an asymmetric network (TPCL) that fully explores Transformer and contrastive learning to generate well-generalized and discriminative fusion features for RGB-D SOD. The main contributions can be summarized as follows:

- We propose an asymmetric network named TPCL for RGB-D SOD by capturing comprehensive multi-modality interactions and pixel-to-pixel relationships to improve detection performance.
- We design a CIPT module to capture long-range multi-modality interactions and generate more comprehensive fusion features for RGB-D SOD.
- We present a PCL method for boosting RGB-D SOD. As far as we know, we are the first to explore pixel-level contrastive learning for RGB-D SOD. Our proposed PCL can learn an embedding space with inter-pixel discrimination and intra-pixel compactness.
- Extensive experiments demonstrate that our proposed TPCL achieves excellent performance against 15 state-of-the-art RGB-D methods on seven public datasets.

II. Related Work

A. RGB-D Salient Object Detection

Traditional RGB-D SOD methods usually utilize hand-crafted features for RGB-D SOD [38], [39]. However, RGB-D SOD often struggles to perform well in complex scenes due to the limited representation abilities of hand-crafted features. Recently, with the advancement of deep learning, CNN-based methods have dominated the entire RGB-D SOD field [4]. In general, RGB-D SOD aims to fuse features extracted from two modalities to generate fusion features that carry saliency cues. Thus, CNN-based RGB-D methods can be classified into three categories: early fusion [14], [22], [23], middle fusion [12], [24], [25], and late fusion [26]. For example, Qu et al. [10] used CNN to learn saliency cues from the generated hand-crafted features, which can be classified into the first category. Existing CNN-based methods usually focus on the middle fusion

strategy, which is attributed to the multi-level feature extraction capability of CNN. For instance, Chen et al. [12] developed a progressive fusion network to fuse multi-level features from each modality. Li et al. [40] proposed a novel hierarchical interactions network to aggregate multi-scale features. To break down the barriers between two modalities, Li et al. [41] proposed an attention-steered interweave fusion network for sufficiently capturing the cross-modality complementary features. Recently, with the development of unsupervised/self-supervised learning technology, some researchers exploited unlabeled information for RGB-D SOD [42], [43], [44], [45]. Furthermore, most of these methods usually adopted a symmetrical architecture to fuse multi-modality features, i.e., used two identical backbones for feature extraction. However, some researchers believed that the architecture might ignore the difference between modalities and proposed asymmetric networks for feature extraction. They primarily designed a lightweight network for depth feature extraction [36], [46]. In the past two years, with the wide application of ViTs, some researchers have chosen Transformer-based backbones for feature extraction and have achieved better performance [18], [19], [20]. Motivated by these investigations, we aim to take the Transformer one step further toward the goal of improving RGB-D SOD performance. Therefore, we follow the paradigm of middle fusion and propose an asymmetric network (TPCL) for RGB-D SOD.

B. Vision Transformer

Transformer is originally presented for machine translation tasks in natural language processing (NLP) [16]. The great success of Transformer mainly benefits from the multi-head self-attention (MHSA) mechanism, which can capture long-distance dependencies between tokens. Thus, inspired by this idea, Alexey et al. [17] naturally migrated the Transformer to computer vision and proposed Vision Transformer (ViT) for image classification. Nonetheless, since ViT has the disadvantages of low output resolution, high computational complexity, and the inability to provide multi-scale features, it is still challenging to employ ViT directly in dense prediction tasks. Based on this, Wang et al. [47], [48] proposed PVT as a pyramid architecture backbone for various dense prediction tasks. PVT can generate global receptive fields and generate multi-scale feature maps for dense prediction tasks. Moreover, to reduce computational complexity, a representative Swin Transformer [49], [50] is proposed. Swin Transformer employs window shift to reduce complexity and achieve global-relationship modeling, which also produces multi-scale features by stacking multiple Swin Transformer blocks. Recently, Wu et al. [51] plugged pyramid pooling effectively into Transformer and proposed P2T, which also achieved significant performance for scene understanding tasks. Recent work in terms of the innovation of ViT is constantly emerging, such as DeiT [52], T2T-ViT [53], CPVT [54], and CVT [55]. In our work, we apply different Transformer-based backbones to cooperate with our method to explore the effectiveness of varying backbones for RGB-D SOD and demonstrate the effectiveness of our method. More importantly, we successfully explore Transformers for multi-modality interaction and fusion, which generate more comprehensive fusion features for RGB-D SOD.

C. Contrastive Learning

Contrastive learning (CL) is a popular discriminative method for self-supervised representation learning [32], [33], [56], [57], [58], [59]. These methods usually regard an instance as an anchor, generate a data augmentation version of the instance as a positive, and view other instances of a mini-batch as negatives. In this case, the contrastive loss InfoNCE [58] is used for training. Such a paradigm of instance discrimination can learn a robust representation space for downstream tasks. However, this paradigm is mainly dependent on batch size and consistency, which requires a large computational resource. To this end, a representative work MoCo [32] is proposed, which builds a large and consistent dictionary to cope with these two shortcomings. Recently, Prannay et al. [34] proposed supervised contrastive learning (SCL) for image classification. They effectively combine contrastive learning with labels and extend the contrastive loss to N positive-negative sample pairs to learn a more discriminative representation space. Naturally, some researchers explored contrastive learning for other tasks and achieved excellent performance, e.g., semantic segmentation [31] and language pre-training [60]. For SOD tasks, improving inter-pixel discrimination and intra-pixel compactness is crucial. Therefore, we propose a novel pixel-level contrastive learning for RGB-D SOD. Specifically, considering the nature of the class-agnostic binary salient object detection task, we aim to design a more fine-grained disentangled framework for RGB-D SOD. Moreover, inspired by the differences between SOD and semantic segmentation, our proposed PCL approach avoids cross-image pixel-level contrast. Therefore, we extend SCL [34] to the RGB-D SOD task while reducing reliance on data augmentation compared to previous works [31], [34], which further reduces computational needs.

III. Methodology

A. Overview

As shown in Fig. 3, we propose an asymmetric network (TPCL) for RGB-D SOD. Firstly, inspired by the asymmetric structure [36] and Transformer-based backbones [18], we propose an asymmetric encoder to extract RGB and depth features. Specifically, the encoder consists of an RGB stream and a depth stream. RGB stream adopts the pre-trained Swin Transformer V2 [49], [50] to extract hierarchical RGB features, and the depth stream proposes a lightweight and efficient CNN-based network (LDNet) without any pre-training to extract depth features. In contrast, our encoder differs from the ones used in [36] and [18]. We use the Swin Transformer V2 as our RGB backbone, reducing computational complexity and GPU memory consumption compared with other Transformer-based backbones (Details will be elaborated in Section IV). Furthermore, the extracted multi-modality features are fed into two weight-shared HMCMF modules to capture multi-scale multi-modality context information in order to generate fusion features. Meanwhile, for

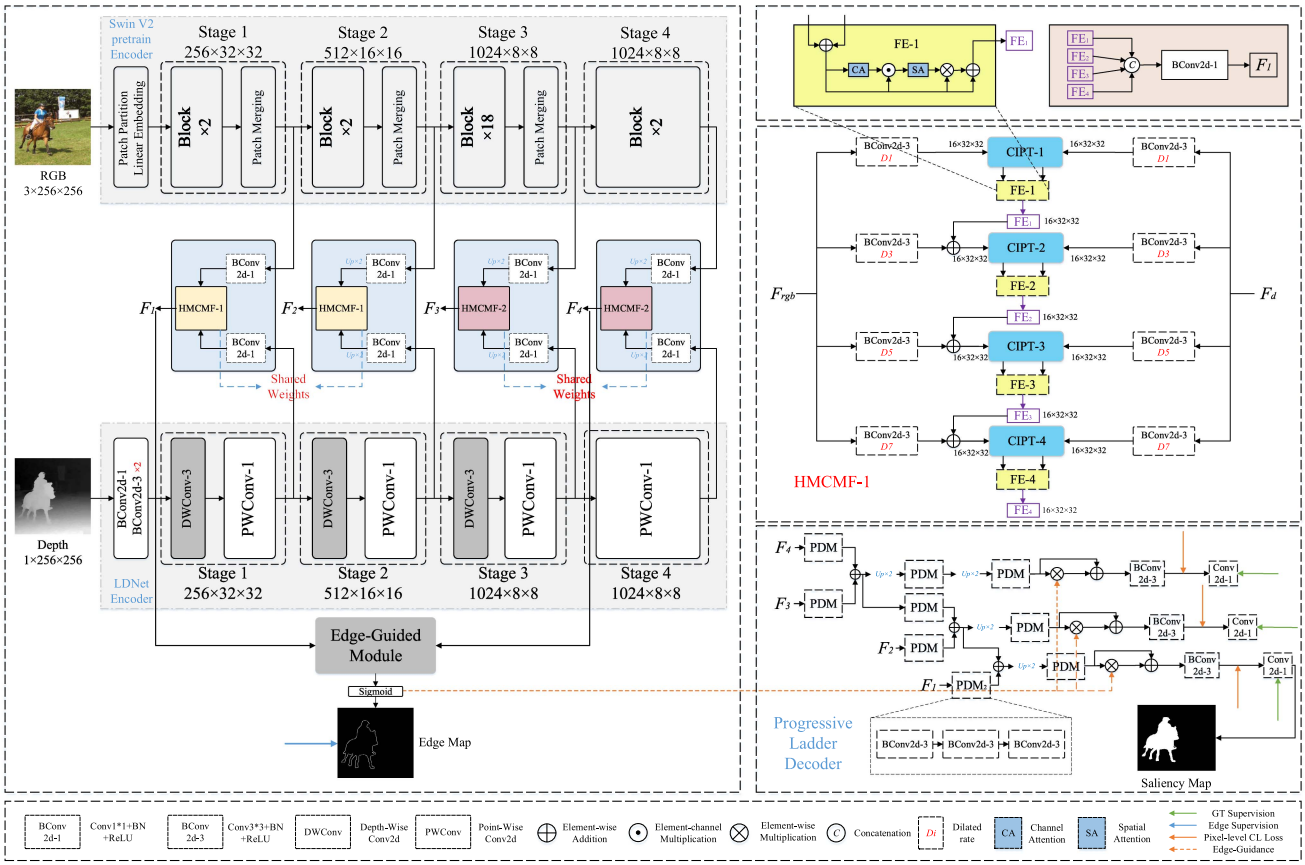


Fig. 3. Overall architecture of our proposed TPCL. TPCL consists of an asymmetric encoder, a hierarchical multi-scale cross-modality fusion (HMCMF) module, an edge-guided module, and a progressive ladder decoder. TPCL includes the Swin Transformer V2 stream and the proposed LDNet stream for the encoding process. Moreover, TPCL exploits two weight-shared HMCMF modules to fuse multi-modalities features. For the decoding process, the progressive ladder decoder fuses multi-stage features and edge features to enhance generated saliency maps. More importantly, TPCL proposes a PCL loss to boost the training process.

capturing more comprehensive multi-modality interactions, we propose a CIPT module to fuse the multi-modality features at each scale. Finally, the obtained fusion features of each stage and edge features generated by the edge-guided module are fed into the progressive ladder decoder to output saliency maps. More importantly, we are the first to explore pixel-level contrastive learning for RGB-D SOD. Unlike previous contrastive learning methods [32], [33] for representation learning, our PCL aims to combine labels to promote compactness within embeddings of saliency or background pixels while also encouraging discrimination between the saliency embeddings and background embeddings. which benefits RGB-D SOD. Therefore, we first introduce our proposed PCL in the next subsection.

B. Pixel-Level Contrastive Learning

Current methods usually use BCE loss to train an RGB-D SOD model. However, a BCE-trained model may not capture intra-pixel compactness and inter-pixel discrimination. Thus, we employ the proposed pixel-level contrastive learning (PCL) to solve the issue. Inspired by the differences between SOD and semantic segmentation, our proposed PCL approach avoids cross-image pixel-level contrast [31]. Meanwhile, motivated by the nature of the class-agnostic binary salient object detection task, PCL provides a more fine-grained disentangled framework for SOD. Specifically, we propose a novel PCL loss for RGB-D SOD. As shown in Fig. 1, PCL aims to reduce the distance between the embeddings of the foreground or background pixels while increasing the distance between foreground and background pixel embeddings. In other words, PCL encourages the model to better separate the foreground and background pixels in the embedding space.

Basically, let $R \in \mathbb{R}^{h\times w\times 3}$ denotes an RGB image, and $D \in \mathbb{R}^{h\times w\times 1}$ denotes a depth map. Let $\mathbf{F} \in \mathbb{R}^{h\times w\times d}$ denotes the fusion feature output by the decoder before the projection head, and the feature is ℓ_2 normalized. N_c is the number of pixels with class $c \in \{0, 1\}$ in ground truth (GT) $\mathbf{G}$; N^G is the number of all pixels in $\mathbf{G}$; $\mathbf{f}_p^F$ denotes a d-dimensional fusion feature vector extracted from $\mathbf{F}$ at pixel p. Let $\mathbb{1}_{pq}^{GG} = \mathbb{1}[y_p^G = y_q^G, p \neq q]$ and $e_{pq}^{\mathbf{FF}} = exp(\mathbf{f}_p^{\mathbf{F}} \cdot \mathbf{f}_{q/}^{\mathbf{F}} \tau)$, in which y_p^G and y_q^G are the class labels of pixel p and q in $\mathbf{G}$, respectively. τ denotes the temperature hyperparameter. Thus, our proposed PCL loss can be defined as follows

$$\mathcal{L}_{PCL} = -\frac{1}{N^G}\sum_{p=1}^{N^G}\frac{1}{N_{y_p^G}^G}\sum_{q=1}^{N_0+N_1}\mathbb{1}_{pq}^{GG}\mathbf{log}\left(\frac{e_{pq}^{\mathbf{FF}}}{\sum_{k=1}^{N_0+N_1}(e_{pk}^{\mathbf{FF}})}\right). \quad (1)$$

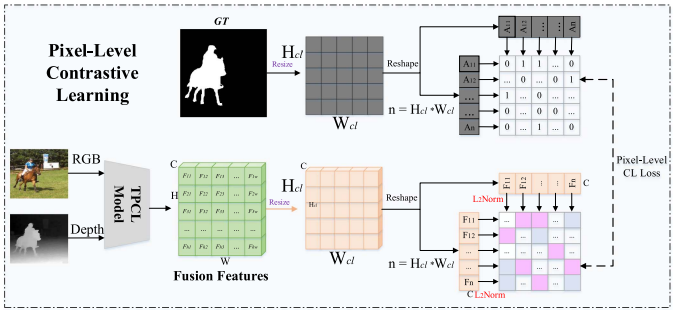


Fig. 4. Pipeline of PCL. The pink squares represent the similarity of positive pairs, and the gray squares denote the similarity of negative pairs. First, a pair of RGB and depth images are fed into the TPCL model to generate a fusion feature tensor. Second, the tensor is resized and reshaped. Third, PCL calculates pixel-pixel score maps (by dot product) between each anchor pixel and other positive or negative pixels. The corresponding GT is resized and reshaped to generate a mask for PCL, in which "1" denotes the positive pairs and "0" represents the negative pairs. Finally, PCL uses (1) to calculate the loss between score maps and masks.

Here, $\mathcal{L}_{PCL}$ represents our proposed PCL loss. $N^G_{y^G_p}$ is the number of pixels with label y^G_p in $\mathbf{G}$. In addition, since mini-batch data is fed into a model for training, the final loss is the average $\mathcal{L}_{PCL}$ of all images in the mini-batch. To illustrate our proposed PCL, Fig. 4 shows the entire pipeline.

Specifically, inspired by supervised contrastive learning [34], we select positive and negative pixel pairs based on their corresponding labels. To be precise, we consider a pixel as an anchor, and all the pixels belonging to the same label are treated as positive samples. Conversely, the pixels that do not belong to the same label are considered negative samples, forming numerous positive and negative pairs. Therefore, we can calculate the similarity between these pairs using dot products, where we aim to increase the similarity of positive pairs and decrease the similarity of negative pairs via our proposed PCL. In practice, due to the limited memory of our GPU, we first resize the learned fused features and GT before performing PCL.

C. Asymmetric Dual Stream Encoder

Recent methods usually choose a symmetrical dual-stream structure, CNN-based or Transformer-based, for RGB-D SOD [18], [40]. These backbones are pre-trained on ImageNet [61] for model initialization. However, inspired by the work of [36], [62], [63], we learn that there are intrinsic differences between RGB images and depth maps, and CNN can capture more depth and position information. Furthermore, saliency prediction is dominated by RGB cues. RGB cues are more suitable for transfer learning on the pre-trained backbones. Therefore, we adopt an asymmetric dual-stream encoder for RGB and depth feature extraction.

1) RGB Encoder Based on Swin Transformer V2: Swin Transformer outperforms pure ViT in tasks including image classification, object detection, and semantic segmentation, thanks to its advantages of multi-scale modeling and linear computing complexity [49]. Additionally, Swin Transformer V2 (Swin V2) [50], a new version of Swin Transformer, offers the merit of more reliable training while saving a significant amount of GPU memory consumption. Thus, we use a Swin V2-based backbone to extract multi-stage features for multi-modality fusion. Specifically, given an input RGB image R, multi-stage RGB features are extracted by Swin V2, which are denoted as $\{\mathbf{F}^i_{rgb} \in \mathbb{R}^{h_i \times w_i \times d_i} \mid i = 1, 2, 3, 4\}$.

2) Depth Encoder Based on LDNet: Depth features usually provide local spatial geometric details for salient object localization. Moreover, most depth-related tasks (e.g., depth estimation, optical flow) do not use ImageNet pre-trained backbones to fine-tune their models. Instead, they employ geometric pre-training [64] or train the entire network from scratch. In addition, CNN-based networks can capture more local spatial features compared with Transformer-based networks, for which we consider that it is not necessary to use a large Transformer-based backbone to process depth maps as in previous work [18]. Therefore, we design a CNN-based network to extract depth features as simply as possible. Shown as Fig. 3, we propose a randomly initialized network to extract multi-stage depth features named LDNet. To be specific, we first employ a stem part that consists of 1×1 convolutional layer (denoted as Conv_1) and two 3×3 convolutional layers (denoted as Conv_3) to extract depth features, and these layers are all equipped with a batch normalization (BN) and a *ReLU* activation. Then, the learned depth features are fed into three stages in turn. For each stage, it consists of a 3×3 depth-wise (DW) convolution and a 1×1 point-wise (PW) convolution with a BN and *ReLU* activation. Finally, for the fourth stage, we exploit a Conv_1 layer to obtain the final output. It is worth noting that these multi-stage depth features maintain the same dimensions as those multi-stage RGB features. Thus, multi-stage depth features are obtained, which are denoted as $\{\mathbf{F}^i_d \in \mathbb{R}^{h_i \times w_i \times d_i} \mid i = 1, 2, 3, 4\}$.

In the above design, two key factors are considered. Specifically, in previous works, the depth encoder of the asymmetric structure typically adopts a lightweight manner, which can be classified into two types: parallel [46] and serial structures [36], [65]. The parallel structure first uses a stem to extract basic depth features, followed by parallel convolution blocks to extract multi-level features. In contrast, the serial structure, similar to VGG [66], employs a serial structure to extract multi-level features but with reduced channel numbers at each level to achieve lightweight purposes. The former abandons pooling layers to preserve the deep geometric structure, while the latter can learn more semantic information from the deep structure. Moreover, other researchers employ a tailored lightweight backbone [67], [68] with a serial structure to extract depth features. Our LDNet is similar to theirs. To maximize both advantages, we adopt a serial structure while abandoning the pooling layer to preserve the deep structure and semantic information.

D. Hierarchical Multi-Scale Cross-Modality Fusion Module

In multi-modality fusion, capturing more comprehensive multi-modality interactions is beneficial for RGB-D SOD. In addition, multi-scale fusion features are essential for RGB-D SOD [40], [69], [70]. Thus, we propose an HMCMF module to capture more comprehensive multi-modality interactions and

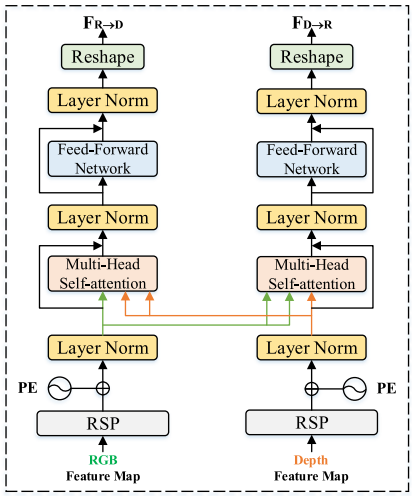


Fig. 5. Overview of the proposed CIPT module.

multi-scale features. HMCMF mainly includes multiple CIPT and FE modules, of which CIPT is the core. Here, we take the **HMCMF-1** as an example for the introduction. The top right corner of Fig. 3 presents the details of **HMCMF-1**.

Specifically, given the RGB features $\mathbf{F}_{rgb}^{i}$ and depth features $\mathbf{F}_{d}^{i}$ of a certain stage $i \in \{1,2,3,4\}$, we first reduce the number of channels of these feature maps to c_i by employing multiple Conv_1 layers. In addition, for the features in the last three stages, we perform a double up-sampling operation to benefit subsequent fusion operations. Then, the obtained RGB features are separately fed into four different Conv_3 layers with dilation rates $\{D_j \mid j = 1,3,5,7\}$ to generate four different scale features, which are denoted as

$$\mathbf{F_{rgb}}_{j}^{i} = \mathrm{Conv}_3\left(\mathbf{F}_{rgb}^{i}\right), j \in \{1,3,5,7\}. \tag{2}$$

Similarly, the obtained depth features can be denoted as $\mathbf{F_d}_{j}^{i}$. Next, $\mathbf{F_{rgb}}_{j}^{i} \in \mathbb{R}^{h_i \times w_i \times c_i/4}$ and $\mathbf{F_d}_{j}^{i} \in \mathbb{R}^{h_i \times w_i \times c_i/4}$ are fed into the CIPT and FE modules to generate fusion features. CIPT fuses RGB and depth features by capturing more comprehensive multi-modality interactions. Fig. 5 shows the details of the CIPT module. Specifically, given the j-th RGB feature maps $\mathbf{F_{rgb}}_{j}^{i} \in \mathbb{R}^{h_i \times w_i \times c_i/4}$ and the j-th depth feature maps $\mathbf{F_d}_{j}^{i} \in \mathbb{R}^{h_i \times w_i \times c_i/4}$, we first split them into fixed-size patches with the size of 4×4, and the patches are reshaped as 1-dimensional feature vectors for fitting the input form of CIPT. Note that the feature vectors are added to learnable positional embeddings. Moreover, unlike a standard multi-head self-attention (MSA) in Transformer, our MSA receives a query Q, a key K, and a value V, which are generated by two modalities. They are represented as two sets of matrices $\mathbf{Q_R}, \mathbf{K_R}, \mathbf{V_R}$ and $\mathbf{Q_D}, \mathbf{K_D}, \mathbf{V_D}$. Therefore, our MSA can be formulated as

$$\begin{aligned} \mathrm{MSA}_{\mathbf{R}\to\mathbf{D}} &= \mathrm{Cat}(\mathrm{head}_0, \ldots, \mathrm{head}_n)\mathbf{W}_1 + \mathbf{I}_{rgb}, \\ \mathrm{MSA}_{\mathbf{D}\to\mathbf{R}} &= \mathrm{Cat}(\mathrm{head}_0, \ldots, \mathrm{head}_m)\mathbf{W}_2 + \mathbf{I}_{d}, \end{aligned} \tag{3}$$

where head_n and head_m are denoted as

$$\mathrm{head}_n = \mathrm{CSA}\left(\mathbf{Q_R}, \mathbf{K_D}, \mathbf{V_D}\right),$$

$$\mathrm{head}_m = \mathrm{CSA}\left(\mathbf{Q_D}, \mathbf{K_R}, \mathbf{V_R}\right). \tag{4}$$

Thus, $\mathrm{CIPT}_{\mathbf{R}\to\mathbf{D}}$ and $\mathrm{CIPT}_{\mathbf{D}\to\mathbf{R}}$ are represented as

$$\begin{aligned} \mathrm{CIPT}_{\mathbf{R}\to\mathbf{D}} &= \mathrm{MSA}_{\mathbf{R}\to\mathbf{D}} + \mathrm{FFN}(\mathrm{MSA}_{\mathbf{R}\to\mathbf{D}}), \\ \mathrm{CIPT}_{\mathbf{D}\to\mathbf{R}} &= \mathrm{MSA}_{\mathbf{D}\to\mathbf{R}} + \mathrm{FFN}(\mathrm{MSA}_{\mathbf{D}\to\mathbf{R}}). \end{aligned} \tag{5}$$

Here, $\mathrm{MSA}(\cdot)$ denotes the MSA and $\mathrm{CSA}(\cdot)$ represents the cross-modality self-attention. $\mathbf{I}_{rgb}$ and $\mathbf{I}_d$ denote the input of two modalities. $\mathrm{FFN}(\cdot)$ denotes the feed-forward layer, and its hidden layer dimension is set to 64 in our study. $\mathbf{W}_1$ and $\mathbf{W}_2$ denote linear projections. $\mathrm{Cat}(\cdot)$ means the concatenation operation. Thus, for the input $\mathbf{F_{rgb}}_{j}^{i}$ and $\mathbf{F_d}_{j}^{i}$, the output can be obtained as follows:

$$\begin{aligned} \mathbf{F}_{\mathbf{R}\to\mathbf{D}_j^i} &= \mathrm{RSP}\left(\mathrm{LN}\left(\mathrm{CIPT}_{\mathbf{R}\to\mathbf{D}}\right)\right), \\ \mathbf{F}_{\mathbf{D}\to\mathbf{R}_j^i} &= \mathrm{RSP}\left(\mathrm{LN}\left(\mathrm{CIPT}_{\mathbf{D}\to\mathbf{R}}\right)\right), \end{aligned} \tag{6}$$

where $\mathrm{LN}(\cdot)$ and PE represent layer normalization and positional embeddings, respectively. $\mathrm{RSP}(\cdot)$ denotes patch partition or reshape operations. $\mathbf{F}_{\mathbf{R}\to\mathbf{D}_j^i}$ and $\mathbf{F}_{\mathbf{D}\to\mathbf{R}_j^i}$ denote the obtained fusion features.

In addition, since the obtained fusion features may contain a modality tendency, we further propose an FE module to better generate the final fusion feature at each scale. FE details are shown in the upper right of Fig. 3. Specifically, for the two fusion features $\mathbf{F}_{\mathbf{R}\to\mathbf{D}_j^i} \in \mathbb{R}^{h_i \times w_i \times c_i/4}$ and $\mathbf{F}_{\mathbf{D}\to\mathbf{R}_j^i} \in \mathbb{R}^{h_i \times w_i \times c_i/4}$, they are first added to obtain the feature $\mathbf{F}_j^i$. Then, we compute the generated feature $\{\mathbf{FE}_k \in \mathbb{R}^{h_i \times w_i \times c_i/4} \mid k \in \{1,2,3,4\}\}$ by FE module as

$$\mathbf{FE}_k = \mathbf{F}_j^i \oplus \left(\mathbf{F}_j^i \otimes \mathrm{SA}\left(\mathbf{F}_j^i \odot \mathrm{CA}\left(\mathbf{F}_j^i\right)\right)\right). \tag{7}$$

Here, $\mathrm{CA}(\cdot)$ and $\mathrm{SA}(\cdot)$ denote channel attention and spatial attention, respectively. $\oplus$, $\odot$, and $\otimes$ represent element-wise addition, channel-wise multiplication, and element-wise multiplication, respectively.

Furthermore, previous works [40], [69], [70] have demonstrated that some additional residual connections are applied to the multi-scale structure, enabling inter-scale interactions. Therefore, $\mathbf{FE}_k$ of the previous scale is added to the RGB feature of the current scale through residual connection to fuse context information.

Last, all features $\{\mathbf{FE}_k \mid k \in \{1,2,3,4\}\}$ are concatenated and then fed into a Conv_1 layer to generate the fusion feature $\mathbf{F}_i$. So far, the HMCMF modules have generated fusion features at the four stages. Meanwhile, to reduce the parameters, TPCL only uses two HMCMF modules with shared weights and the number of layers and the number of heads of Transformer in each CIPT module is 1 and 8 (i.e., $n = m = 8$), respectively. Note that the shared weight strategy should not be implemented among the four HMCMF modules. Specifically, there are two reasons: the first is that the low-level information and the high-level information are represented inconsistently; the second is that the resolution of the feature maps is different, which makes it difficult for us to implement this strategy among the four modules.

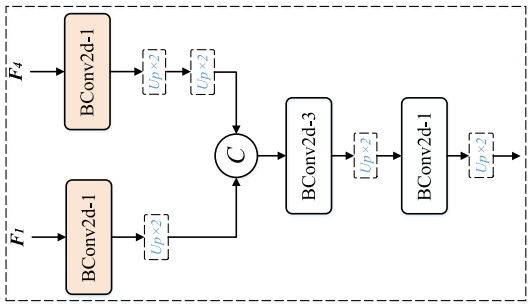


Fig. 6. Overview of the proposed edge-guided module.

E. Edge-Guided Module

Previous work [18], [37] has demonstrated that learning acceptable edge features can help the model to localize and refine salient objects in a scene. Moreover, since the low-level features contain rich details, they usually exploit low-level features with a tailored module to learn edge representations. Although low-level details contribute more information to learning edge features, they may carry non-edge coarse elements that affect the learning process [71]. Therefore, we consider integrating low-level and high-level semantic features for learning edge features of salient objects.

Fig. 6 shows the schematic of our edge-guided module. Specifically, the fusion features $\mathbf{F}_1$ and $\mathbf{F}_4$ are used to learn edge features. First, $\mathbf{F}_1$ and $\mathbf{F}_4$ are fed into two separate Conv_1 layers companies with up-sampling operations, respectively. Then, the obtained features are concatenated to generate a new feature. Next, the feature is fed into two successive Conv_3 and Conv_1 layers to learn an acceptable edge feature representation. The process can be formulated as

$$\begin{aligned}\mathbf{F}_{\mathrm{edge}_t} &= \mathrm{Cat}(\delta_{\uparrow}\left(\mathrm{Conv}_1\left(\mathbf{F}_1\right)\right), \delta_{\uparrow}\left(\mathrm{Conv}_1\left(\mathbf{F}_4\right)\right)),\\ \mathbf{F}_{\mathrm{edge}} &= \delta_{\uparrow}\left(\mathrm{Conv}_1\left(\delta_{\uparrow}\left(\mathrm{Conv}_3(\mathbf{F}_{\mathrm{edge}_t})\right)\right)\right),\end{aligned} \tag{8}$$

where $\delta_{\uparrow}(\cdot)$ denotes the up-sampling operation and $\mathbf{F}_{\mathrm{edge}}$ denotes the learned edge feature.

F. Progressive Ladder Decoder

Now, multi-level fusion features $\{\mathbf{F}_i \mid i \in \{1,2,3,4\}\}$ and edge feature $\mathbf{F}_{\mathrm{edge}}$ are obtained. To this end, these features should be fed into a tailed decoder to generate final saliency maps. Thus, we propose a progressive ladder decoder to output the desired saliency maps. The bottom right of Fig. 3 shows the details of our decoder.

Specifically, our decoder employs a progressive aggregation to integrate multi-level fusion features. Here, a progressive decoder module named PDM is first introduced. The module consists of three successive Conv_3 layers followed by a 2-fold up-sampling operation. Taking the $\mathbf{F}_4$ and $\mathbf{F}_3$ as an example to introduce the forward processes of our decoder. First, $\mathbf{F}_4$ and $\mathbf{F}_3$ are fed into two separate PDM modules, respectively. Then, the obtained two features are added together to generate a new feature. Next, the new feature is fed into two separate PDM modules, in which one is used to decode the current saliency map, and the other is used to aggregate the $\mathbf{F}_2$ to participate in decoding the next saliency map. In addition, the edge feature is aggregated in the multi-level decoder progress by an element-wise multiplication operation. In a word, given the multi-level fusion features $\{\mathbf{F}_i \mid i \in \{1,2,3,4\}\}$ and edge features $\mathbf{F}_{\mathrm{edge}}$, the decoder can be formulated as follows:

$$\begin{aligned}\mathbf{S}_f &= \delta_{\uparrow}\left(\mathrm{PDM}\left(\delta_{\uparrow}\left(\mathrm{PDM}\left(\mathbf{F}_i\right)\oplus \mathrm{PDM}\left(\mathbf{F}_{i-1}\right)\right)\right)\right),\\ \mathbf{C}_l &= \mathrm{Conv}_3\left(\mathbf{S}_f \oplus \left(\mathbf{S}_f \otimes \mathrm{Sig}\left(\mathbf{F}_{\mathrm{edge}}\right)\right)\right),\\ \mathbf{S}_o &= \mathrm{Conv}_1\left(\mathbf{C}_l\right),\end{aligned} \tag{9}$$

where $\mathrm{PDM}(\cdot)$ denotes the proposed PDM module. $\mathrm{Sig}(\cdot)$ represents the *Sigmoid* activation. $\{\mathbf{C}_l \mid l \in \{1,2,3\}\}$ is used to calculate the PCL loss. $\mathrm{Conv}_1(\cdot)$ is used as a projection head to output the final saliency map $\{\mathbf{S}_o \mid o \in \{1,2,3\}\}$. To this end, our model can output three saliency maps. The saliency map $\mathbf{S}_1$ is used for the final evaluation result.

G. Hybrid Loss Function

Our model TPCL is trained by a hybrid loss function: pixel-level contrastive loss $\mathcal{L}_{PCL}$, saliency loss $\mathcal{L}_S$, and edge loss $\mathcal{L}_E$. The total loss is formulated as

$$\mathcal{L} = 0.9 \times \mathcal{L}_{PCL} + 0.1 \times \mathcal{L}_S + \mathcal{L}_E. \tag{10}$$

$\mathcal{L}_S$ consists of $\mathcal{L}_{BCE}$ and $\mathcal{L}_{HEL}$. It can be defined as

$$\begin{aligned}\mathcal{L}_{S_1} &= \mathcal{L}_{BCE}\left(\mathbf{S}_1, \mathbf{G}\right) + \mathcal{L}_{HEL}\left(\mathbf{S}_1, \mathbf{G}\right),\\ \mathcal{L}_{S_2} &= \mathcal{L}_{BCE}\left(\mathbf{S}_2, \mathbf{G}\right) + \mathcal{L}_{HEL}\left(\mathbf{S}_2, \mathbf{G}\right),\\ \mathcal{L}_{S_3} &= \mathcal{L}_{BCE}\left(\mathbf{S}_3, \mathbf{G}\right) + \mathcal{L}_{HEL}\left(\mathbf{S}_3, \mathbf{G}\right),\\ \mathcal{L}_S &= 1 \times \mathcal{L}_{S_1} + 0.5 \times \mathcal{L}_{S_2} + 0.5 \times \mathcal{L}_{S_3}.\end{aligned} \tag{11}$$

Here, $\mathcal{L}_{HEL}$ represents the proposed enhanced loss in [72], which can be used to optimize the foreground and background independently. $\mathcal{L}_{BCE}$ denotes the BCE loss that is defined as

$$\mathcal{L}_{BCE} = \frac{\mathbf{G}\log \mathbf{S_o} + (\mathbf{1}-\mathbf{G})\log\left(\mathbf{1}-\mathbf{S_o}\right)}{N^G}, \tag{12}$$

where $\mathbf{G}$ is the corresponding GT.

$\mathcal{L}_E$ is defined as

$$\mathcal{L}_E = w\mathcal{L}_{BCE}\left(\mathbf{F}_{\mathrm{edge}}, \mathbf{G}_{\mathrm{edge}}\right) + \mathcal{L}_{Dice}\left(\mathbf{F}_{\mathrm{edge}}, \mathbf{G}_{\mathrm{edge}}\right). \tag{13}$$

Here, since the proportion of edge pixels in an edge map $\mathbf{G}_{\mathrm{edge}}$ is tiny, resulting in a distribution imbalance problem. Therefore, we use weighted $w\mathcal{L}_{BCE}$ to balance the distribution of pixels. Additionally, we adopt a popular $\mathcal{L}_{Dice}$ loss [73] for supervised training, which may more effectively optimize the unbalanced distribution for better edge supervision.

For PCL loss $\mathcal{L}_{PCL}$, it can be defined as

$$\begin{aligned}\mathcal{L}_{PCL} = {} & 1 \times \mathcal{L}_{PCL_1}\left(\mathbf{C}_1, \mathbf{G}\right)\\ & + 0.5 \times \mathcal{L}_{PCL_2}\left(\mathbf{C}_2, \mathbf{G}\right) + 0.5 \times \mathcal{L}_{PCL_3}\left(\mathbf{C}_3, \mathbf{G}\right).\end{aligned} \tag{14}$$

TABLE I
QUANTITATIVE RESULTS BETWEEN THE PROPOSED TPCL AND 15 STATE-OF-THE-ART METHODS OF STRUCTURE MEASURE ($S_\alpha \uparrow$), MAX F-MEASURE ($F_\beta \uparrow$), MAX E-MEASURE ($E_\xi \uparrow$), AND MAE $\downarrow$ ON SEVEN RGB-D DATASETS

	Method	A2dele [90]	HDFNet [72]	**TriTrans** [19]	DCF [84]	DSA^{2}F [65]	CDNet [91]	HAINet [40]	JL-DCF [15]	SPNet* [92]	**VST** [20]	RD3D* [93]	SSL [42]	DCMF [46]	C^{2}DFNet [94]	SPSN [95]	Ours	Ours*
	Year	CVPR 2020	ECCV 2020	MM 2021	CVPR 2021	CVPR 2021	TIP 2021	TIP 2021	TPAMI 2021	ICCV 2021	ICCV 2021	AAAI 2021	AAAI 2022	TIP 2022	TMM 2022	ECCV 2022		
	Input Size	256 × 256	320 × 320	256 × 256	352 × 352	256 × 256	224 × 224	352 × 352	320 × 320	352 × 352	224 × 224	352 × 352	256 × 256	256 × 256	256 × 256	352 × 352	256 × 256	256 × 256
	Backbones	VGG-16 [66]	VGG-16 [66]	Res-50 [96]+ViT-B [17]	Res-50 [96]	VGG-19 [66]	VGG-16 [66]	VGG-16 [66]	Res-101 [96]	Res2-50 [97]	T2T-ViT-14 [53]	I3DRes-50 [98]	VGG-16 [66]	VGG-16 [66]	Res-50 [96]	VGG-16 [66]	Swin V2-B [50]	Swin V2-B [50]
		VGG-16 [66]	VGG-16 [66]	Res-50 [96]+ViT-B [17]	Res-50 [96]	DepthNet [36]	VGG-16 [66]	VGG-16 [66]	Res-101 [96]	Res2-50 [97]	T2T-ViT-14 [53]	I3DRes-50 [98]	VGG-16 [66]	DepthNet [46]	Res-50 [96]	VGG-16 [66]	LDNet	LDNet
NJU2K [74]	$S_\alpha \uparrow$	0.871	0.908	0.920	0.903	0.904	0.918	0.909	0.902	0.925	0.922	0.916	0.909	0.913	0.908	0.918	0.926	0.925
	$F_\beta \uparrow$	0.874	0.911	0.926	0.905	0.917	0.918	0.909	0.904	0.935	0.920	0.914	0.923	0.915	0.909	0.921	0.930	0.930
	$E_\xi \uparrow$	0.916	0.944	0.955	0.943	0.937	0.950	0.941	0.944	0.954	0.951	0.947	0.939	0.948	0.942	0.952	0.959	0.959
	MAE $\downarrow$	0.051	0.039	0.030	0.038	0.039	0.036	0.038	0.041	0.028	0.035	0.036	0.038	0.043	0.038	0.033	0.028	0.028
NLPR [78]	$S_\alpha \uparrow$	0.899	0.923	0.928	0.922	0.918	0.929	0.921	0.925	0.927	0.932	0.930	0.922	0.922	0.928	0.923	0.936	0.935
	$F_\beta \uparrow$	0.882	0.917	0.924	0.910	0.906	0.919	0.908	0.918	0.925	0.920	0.919	0.923	0.906	0.917	0.912	0.930	0.930
	$E_\xi \uparrow$	0.944	0.963	0.966	0.957	0.952	0.960	0.954	0.963	0.959	0.962	0.965	0.960	0.954	0.961	0.960	0.970	0.970
	MAE $\downarrow$	0.029	0.023	0.020	0.023	0.024	0.023	0.025	0.022	0.021	0.024	0.022	0.025	0.029	0.021	0.023	0.017	0.017
DUT-R [76]	$S_\alpha \uparrow$	0.885	0.908	0.933	0.924	0.921	0.930	0.910	0.906	-	0.943	0.932	0.929	0.928	0.933	-	0.946	0.936
	$F_\beta \uparrow$	0.891	0.915	0.946	0.932	0.938	0.934	0.920	0.910	-	0.948	0.939	0.947	0.932	0.944	-	0.956	0.946
	$E_\xi \uparrow$	0.928	0.945	0.966	0.957	0.956	0.955	0.944	0.941	-	0.969	0.960	0.958	0.958	0.964	-	0.974	0.967
	MAE $\downarrow$	0.043	0.041	0.025	0.030	0.031	0.029	0.038	0.042	-	0.024	0.031	0.029	0.035	0.025	-	0.020	0.024
STERE [75]	$S_\alpha \uparrow$	0.879	0.900	0.908	0.905	0.897	0.912	0.909	0.903	0.907	0.913	0.911	0.904	0.910	0.902	0.906	0.920	0.916
	$F_\beta \uparrow$	0.880	0.900	0.911	0.904	0.910	0.907	0.909	0.904	0.915	0.907	0.906	0.914	0.906	0.897	0.902	0.922	0.917
	$E_\xi \uparrow$	0.928	0.943	0.953	0.948	0.942	0.947	0.947	0.947	0.944	0.951	0.947	0.939	0.946	0.943	0.945	0.960	0.956
	MAE $\downarrow$	0.045	0.042	0.033	0.037	0.039	0.037	0.038	0.040	0.037	0.038	0.037	0.039	0.043	0.038	0.036	0.029	0.031
LFSD [77]	$S_\alpha \uparrow$	0.825	0.846	0.866	0.856	0.883	0.877	-	0.853	-	0.882	0.858	-	0.878	0.863	-	0.892	0.885
	$F_\beta \uparrow$	0.828	0.858	0.870	0.860	0.889	0.879	-	0.863	-	0.889	0.854	-	0.875	0.867	-	0.888	0.887
	$E_\xi \uparrow$	0.866	0.889	0.905	0.903	0.924	0.911	-	0.894	-	0.921	0.891	-	0.909	0.903	-	0.926	0.918
	MAE $\downarrow$	0.084	0.085	0.066	0.071	0.055	0.061	-	0.077	-	0.061	0.073	-	0.068	0.065	-	0.049	0.059
SIP [14]	$S_\alpha \uparrow$	0.829	0.886	0.886	0.873	0.862	0.862	0.886	0.880	0.894	0.904	0.885	0.888	0.788	0.871	0.892	0.902	0.900
	$F_\beta \uparrow$	0.834	0.894	0.899	0.886	0.891	0.888	0.903	0.889	0.916	0.915	0.889	0.909	0.772	0.877	0.900	0.922	0.917
	$E_\xi \uparrow$	0.890	0.930	0.930	0.922	0.911	0.905	0.927	0.925	0.930	0.944	0.924	0.927	0.855	0.916	0.936	0.946	0.943
	MAE $\downarrow$	0.070	0.048	0.043	0.052	0.057	0.060	0.048	0.049	0.043	0.040	0.048	0.046	0.107	0.053	0.043	0.035	0.038
DES [79]	$S_\alpha \uparrow$	0.886	0.926	0.943	0.916	0.916	0.935	0.929	0.931	0.944	0.943	0.935	0.936	0.932	0.921	0.936	0.941	0.935
	$F_\beta \uparrow$	0.872	0.921	0.941	0.907	0.915	0.930	0.924	0.923	0.946	0.940	0.929	0.939	0.924	0.916	0.938	0.942	0.935
	$E_\xi \uparrow$	0.920	0.970	0.981	0.956	0.954	0.973	0.967	0.968	0.983	0.978	0.972	0.978	0.968	0.960	0.975	0.977	0.970
	MAE $\downarrow$	0.029	0.021	0.014	0.023	0.023	0.020	0.019	0.020	0.014	0.017	0.019	0.017	0.023	0.020	0.016	0.015	0.017

The best and second best are highlighted in red and blue in each column. "*"Denotes the model is trained on setting A. Transformer-based methods are highlighted in bold. "Res-50" and "Res2-50" denotes the ResNet-50 and Res2Net-50 backbones, respectively.

IV. EXPERIMENTS

A. Experimental Setup

1) Datasets: We perform our experiments on seven popular RGB-D SOD datasets to validate our proposed TPCL model: NJU2K [74], STERE [75], DUT-RGBD [76], SIP [14], LFSD [77], NLPR [78], and DES [79]. Existing work usually has two widely used training settings:

A: 1485 samples from NJU2K [74] and 700 samples from NLPR [78] for training [13], [80], [81];

B: 1485 samples from NJU2K [74], 700 samples from NLPR [78], and 800 samples from DUT-RGBD [76] for training [36], [65], [68], [76], [82], [83], [84].

For fair comparisons with previous work, we train our model on the two different settings respectively. We will report our results on the two settings.

2) Evaluation Metrics: We evaluate our model and other models by using five metrics: S-measure (S_α) [85], max F-measure (F_β) [86], max E-measure (E_ξ) [87], mean absolute error (MAE) [88], and precision-recall (PR) curve.

3) Implementation Details: Our model is trained on a single NVIDIA GTX 3080Ti GPU with 12 GB of memory and implemented with the PyTorch toolbox. During the training and testing processes, the size of the input RGB images and depth maps are resized to 256×256. We follow [18] to generate edge GT from saliency GT. Moreover, all images are implemented with data augmentation (e.g., random flipping, random cropping, random rotation, and color enhancement) to avoid overfitting for training. The backbone network of the RGB stream adopts the Swin V2 Base model [50] pre-trained on ImageNet. The parameters of the depth stream and other components are initialized with the default setting of PyTorch. We use the AdamW [89] optimizer with $\beta_s = (0.9, 0.999)$, $\epsilon = 1e-8$, and weight_decay $= 1e-4$ to optimize our model. For our proposed PCL, we set the hyperparameter temperature τ as 0.3. In addition, since the limited memory of our GPU and the quadratic complexity of PCL, the learned fusion feature and GT are first resized before computing the $\mathcal{L}_{PCL}$. We respectively use bilinear interpolation and nearest neighbor interpolation to resize the feature map and GT to 52×52. The initial learning rate is $5e-5$, and then follows the *poly* decay strategy that formulates $lr = init_{lr} \times (1 - (curr_{lr}/max_{iter})^{power}$ to adjust the current learning rate, and $power = 0.9$. We train our model for 300 epochs with a batch size of 3, and the total time taken is close to 24 hours when trained on Setting B.

B. Performance Comparison

We compare our proposed TPCL with 15 state-of-the-art RGB-D SOD models, including A2dele [90], HDFNet [72], TriTransNet [19], DCF [84], DSA^{2}F [65], CDNet [91], HAINet [40], JL-DCF [15], SP-Net [92], VST [20], RD3D [93], SSL [42], DCMF [46], C^{2}DFNet [94], and SPSN [95]. For fair comparisons, we recalculate the max F-measure and E-measure by the predicted saliency maps, pre-trained models, and codes they provided.

1) Quantitative Comparisons: Table I reports the quantitative comparisons in terms of S_α, F_β, E_ξ, and MAE on seven RGB-D datasets. From Table I, it can be clearly seen that our TPCL notably outperforms all the compared methods on these datasets, in which Setting B version of TPCL achieves the best F_β values of 93.0%, 95.6%, 92.2%, and 92.2% on NLPR, DUT-RGBD, STERE, and SIP datasets, respectively. Meanwhile, the F_β value of TPCL is slightly lower than DSA^{2}F and

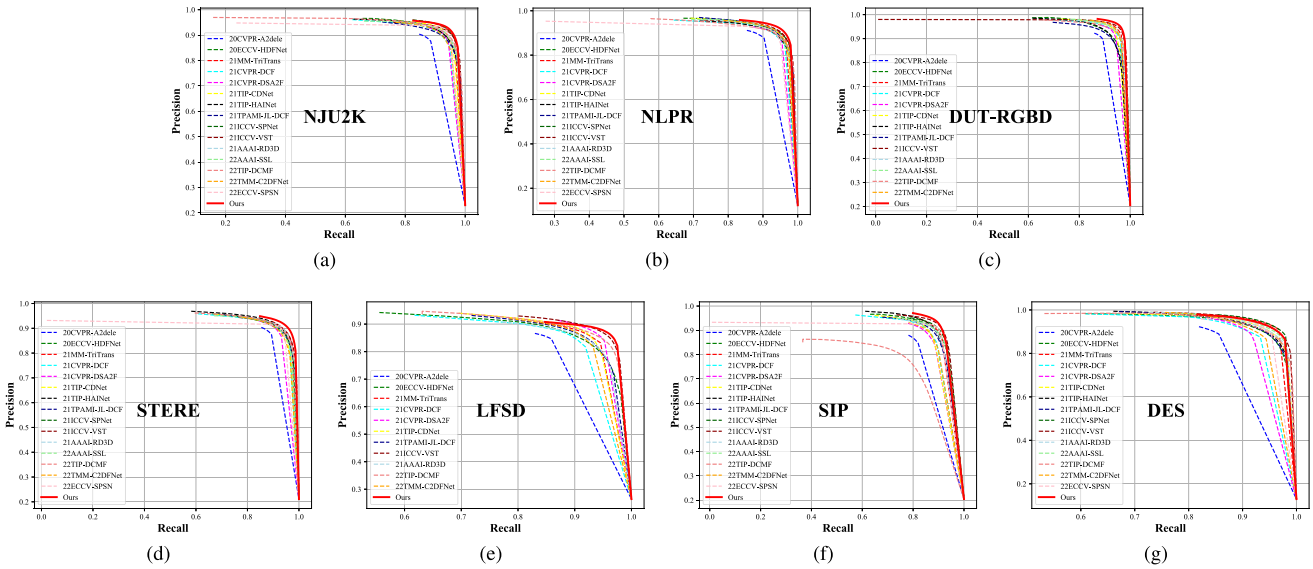


Fig. 7. PR Curves of TPCL and 15 state-of-the-art models on the seven RGB-D datasets.

VST by 0.1% on the LFSD dataset. On the NJU2K dataset, although it is considered harder to identify salient objects due to the inclusion of more low-contrast scenes, TPCL outperforms almost all other models. This result demonstrates that our model also generalizes well in complex scenes. In terms of the metrics E_ξ and MAE, both Setting B versions of TPCL achieve the best performance on all datasets. In addition, on the LFSD and SIP datasets, although TPCL performs slightly worse than DSA^{2}F and VST in terms of the metrics E_ξ and MAE when trained on Setting A, our model still has strong generalization capabilities since it outperforms SPNet and RD3D under the same Setting A. In terms of the metric S_α, both Setting A and B versions of TPCL attain the best performance in all cases except for Setting A version on the SIP dataset is slightly lower than VST by 0.2%. Furthermore, compared with the Transformer-based models TriTransNet [19] and VST [20], TPCL outperforms them in almost all metrics. Also, we observe that Transformer-based methods outperform CNN-based methods, which confirms that Transformer achieves better performance for salient feature learning. Therefore, we apply Transformer to learn rich fusion semantics between RGB and depth modalities, which benefits RGB-D SOD tasks.

2) PR Curves: The PR curves of the proposed TPCL and the other 15 state-of-the-art methods on the seven RGB-D datasets are displayed in Fig. 7. The highest curve indicates the best performance of the associated model. As we can see, TPCL beats its competitors.

3) Qualitative Comparisons: To further show the effectiveness of TPCL, Fig. 8 displays some representative saliency maps between TPCL and 15 state-of-the-art methods in various scenes. Compared with these state-of-the-art methods, our method can successfully detect salient objects in these challenging scenes. Generally, in the cases of fine edges and details (1st–2nd rows), our model can capture more fine-grained information, which indicates that edge-guided module and PCL can facilitate TPCL to focus on the edge and pixel-level information. Similarly, in the case of small objects and low contrast (3rd–4th rows), our model can localize the salient regions better. Moreover, in the cases of low-quality depth maps (5th–6th rows), large objects (7th–8th rows), multiple objects, complex scenes, smooth edges, and complicated shapes (9th–12th rows), our model exhibits robust generalization performance in these scenes, which indicates the effectiveness of our model. In particular, in the fifth row, the squirrel's tail was not labeled in GT, but our model still detected the unlabeled tail.

TABLE II
COMPARISONS OF FLOPS AND PARAMETERS

Method	Backbone (RGB)	Backbone (Depth)	FLOPs (G)	Params (M)
A2dele [90]	VGG-16 [66]	VGG-16 [66]	41.86	30.34
HDFNet [72]	VGG-16 [66]	VGG-16 [66]	58.72	44.15
TriTrans [19]	Res-50 [96]+ViT-B [17]	Res-50 [96]+ViT-B [17]	**282.58**	**139.55**
DCF [84]	Res-50 [96]	Res-50 [96]	107.82	108.49
DSA^{2}F [65]	VGG-19 [66]	DepthNet [36]	50.31	23.62
CDNet [91]	VGG-16 [66]	VGG-16 [66]	94.13	32.93
HAINet [40]	VGG-16 [66]	VGG-16 [66]	96.06	59.82
JL-DCF [15]	Res-101 [96]	Res-101 [96]	**470.81**	**143.52**
SPNet [92]	Res2-50 [97]	Res2-50 [97]	36.01	**175.29**
RD3D [93]	I3DResNet-50 [98]	I3DResNet-50 [98]	101.46	46.90
SSL [42]	VGG-16 [66]	VGG-16 [66]	143.97	74.17
DCMF [46]	VGG-16 [66]	DepthNet	102.33	58.94
C^{2}DFNet [94]	Res-50 [96]	Res-50 [96]	11.08	51.61
SPSN [95]	VGG-16 [66]	VGG-16 [66]	53.17	37.04
Ours	Swin V2-B [50]	LDNet	**212.02**	129.47

For fairness, all the input images are resized to 256 × 256 for comparison.
The highest three results are highlighted in bold.

4) Complexity Comparisons: We calculate FLOPs and parameters for all compared models, and the results are shown in Table II below. For fairness, all the input images are resized to 256×256 for comparisons. Specifically, the results show that our model is high in FLOPs and parameters. However, compared with TriTrans [19], JL-DCF [15], and SPNet [92], our model can achieve superior performance with fewer parameters and FLOPs. Moreover, the computation cost mainly exists in the Swin Transformer backbone (87.92 M Params).

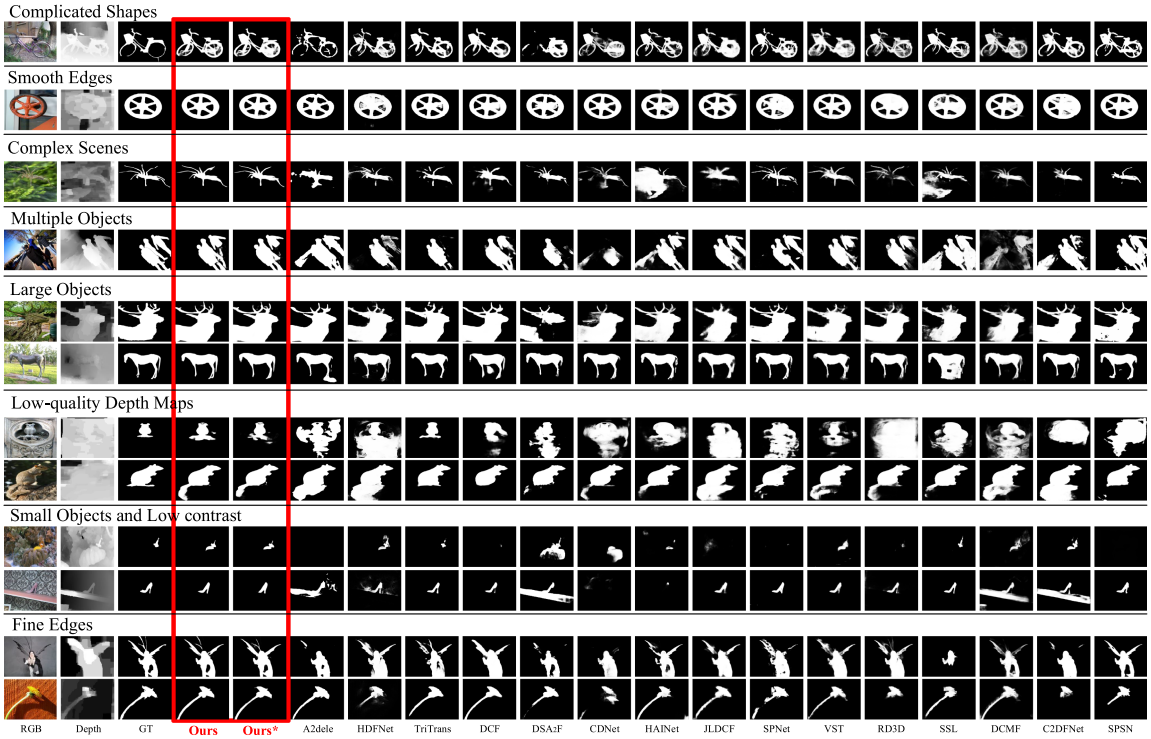


Fig. 8. Qualitative comparisons between TPCL and 15 recently state-of-the-art RGB-D SOD models. "*" denotes the visual results trained on Setting A. Our TPCL is remarkable in some challenging cases: fine edges and details (1st–2nd rows), small objects, low contrast (3rd–4th rows), low-quality depth maps (5th–6th rows), large objects (7th–8th rows), multiple objects, complex scenes, smooth edges, and complicated shapes (9th–12th rows).

TABLE III
ABLATION STUDIES ON THREE DATASETS, IN WHICH EG DENOTES THE PROPOSED EDGE-GUIDED MODULE

Model	$\mathcal{L}_S$	$\mathcal{L}_{PCL}$	$\mathcal{L}_E$+EG	CIPT	FE	**STERE** [75]				**LFSD** [77]				**DES** [79]			
						$S_\alpha\uparrow$	$F_\beta\uparrow$	$E_\xi\uparrow$	MAE $\downarrow$	$S_\alpha\uparrow$	$F_\beta\uparrow$	$E_\xi\uparrow$	MAE $\downarrow$	$S_\alpha\uparrow$	$F_\beta\uparrow$	$E_\xi\uparrow$	MAE $\downarrow$
V1	✓			✓	✓	0.915	0.916	0.956	0.033	0.880	0.874	0.916	0.058	0.932	0.931	0.965	0.018
V2	✓		✓	✓	✓	0.916	0.913	0.954	0.032	0.885	0.885	0.923	0.054	0.936	0.934	0.969	0.017
V3	✓	✓		✓	✓	0.917	0.919	0.959	0.031	0.892	**0.889**	**0.929**	0.049	0.941	0.938	0.976	0.016
V4	✓	✓	✓		✓	0.912	0.915	0.956	0.035	0.883	0.882	0.918	0.058	0.934	0.930	0.969	0.017
V5	✓	✓	✓	✓		0.919	0.920	0.959	0.030	0.888	0.885	0.924	0.051	0.938	0.935	0.974	0.016
Ours	✓	✓	✓	✓	✓	**0.920**	**0.922**	**0.960**	**0.029**	**0.892**	0.888	0.926	**0.049**	**0.941**	**0.942**	**0.977**	**0.015**

V1-V5 represents baseline, contrastive learning removed, edge guidance removed, CIPT module removed, and FE module removed, respectively.
The best results of each column are highlighted in bold.

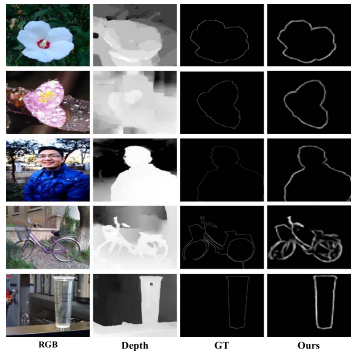


Fig. 9. Edge predictions of salient objects output by edge-guided module.

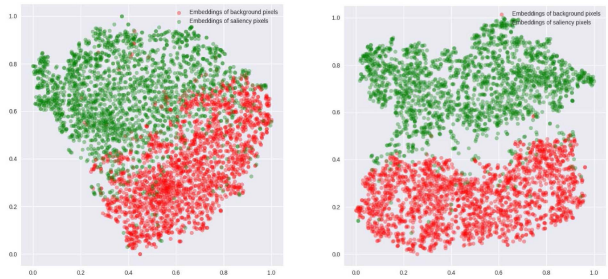
C. Ablation Studies

1) Effect of Each Component: To verify the effectiveness of our proposed components in TPCL, we conduct multiple ablation studies and generate different variants of TPCL to show the effect of the proposed components. All ablation studies are trained on Setting B.

Table III shows the effects of our proposed components tested on STERE [75], LFSD [77], and DES [79]. We build a baseline (the 1st row of Table III, denoted as V1), removing PCL and edge-guided module with $\mathcal{L}_E$. We note that the baseline model still achieves superior performance, attributed to the powerful feature representation and fusion capabilities of the Transformer. Next, we will conduct the following analysis:

a) Effect of the edge-guided module: To demonstrate the effectiveness of the edge-guided module, we further perform ablation studies on the edge-guided module. The ablation results of the edge-guided module (denoted as V2) are shown in the 2nd row of Table III. We can learn that with the help of the edge-guided module, the performance of TPCL is enhanced. We do not specifically tailor a complex module to learn edge features to enhance saliency prediction as in previous works. Here, we present several learned edge predictions in Fig. 9 to demonstrate the effectiveness of the edge-guided module. Additionally, it can be observed that our edge-guided module achieves superior performance in learning salient object-related edge information and refining salient object detection.

b) Effect of the proposed pixel-level contrastive learning: To demonstrate the effectiveness of the proposed PCL, we conduct ablation studies by adding this component base on the baseline. The ablation results (denoted as V3) are displayed in the 3rd row of Table III. We can see that although the performance of V3



(a) without $\mathcal{L}_{PCL}$ (b) with $\mathcal{L}_{PCL}$

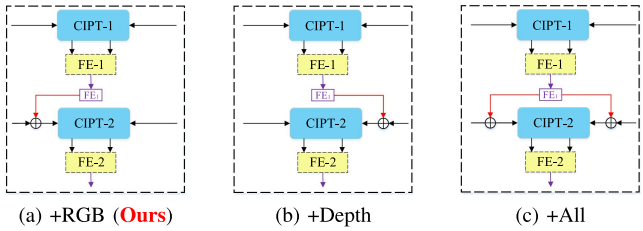
Fig. 10. T-SNE visualization of fusion features of the four datasets learned without (a) pixel-level contrastive loss [i.e., $\mathcal{L}_{PCL}$ in (1)] and with $\mathcal{L}_{PCL}$. (b) Embeddings are colored according to labels.

is slightly higher than that of the baseline, there is still a particular gap compared with the entire TPCL, indicating that PCL leads to performance improvements. Furthermore, to more intuitively show the role and effectiveness of PCL, we use t-SNE [35] to visualize the learned fused feature distributions (extracted before the final 1×1 convolutional layer, i.e., feature $\mathbf{C}_1$). In practice, since the number of pixels is large, we visualize the distributions by computing the mean of the embeddings of the saliency pixel set and the background pixel set for each sample.

Fig. 10 shows the distributions of saliency and background embeddings. We can clearly see that exploiting the proposed loss $\mathcal{L}_{PCL}$ to participate in the training of TPCL can learn discriminative fusion representations better. Specifically, the representations learned by $\mathcal{L}_{PCL}$ contain the intra- and inter-relationships between pixels. Moreover, it can be observed that the embeddings of the same label are clustered more compactly, and the boundary line between the saliency pixels and the background pixels is also more highlighted. Hence, PCL can improve the capability of discriminative fusion representation learning and explore the relationships between pixels, which significantly boosts RGB-D SOD.

c) Effects of the proposed CIPT and FE modules: We perform ablation studies to demonstrate the impact of CIPT and FE modules. The 4th and 5th rows (denoted as V4 and V5) of Table III display the quantitative results. The numerical metrics show that the performance will decrease when these two modules are removed. Significantly, when the CIPT module is removed, the performance drops dramatically. The results demonstrate the ability of CIPT to effectively capture the semantic interactions between the two modalities, thereby generating comprehensive fusion features. Additionally, the findings suggest that the Transformer can be utilized for fusing multi-modality features and achieving superior performance.

d) Effects of the residual connections in HMCMF Modules: Since previous works [40], [69], [70] have demonstrated that some additional residual connections are applied to the multi-scale structure, enabling inter-scale interactions. Therefore, we follow the residual connection manner in the HAINet [40], in which the fused features of the previous scale are added to the RGB features of the next scale, which can achieve depth feature purification and RGB feature enhancement. This manner can achieve robust detection in the presence



(a) +RGB (**Ours**) (b) +Depth (c) +All

Fig. 11. Structures of three residual connection manners. (a) "+RGB" means that $\mathbf{FE}_k + \mathbf{F}_{rgb}$. (b) "+Depth" means that $\mathbf{FE}_k + \mathbf{F}_d$. (c) "+All" means that $\mathbf{FE}_k + \mathbf{F}_{rgb}$ and $\mathbf{FE}_k + \mathbf{F}_d$.

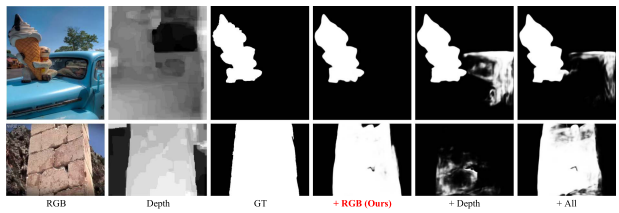


Fig. 12. Comparison of three different residual connections.

TABLE IV
EFFECTS OF DIFFERENT INPUTS ON EDGE-GUIDED MODULES

Methods	STERE [75]				LFSD [77]				SIP [14]			
	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	MAE$\downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	MAE$\downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	MAE$\downarrow$
$\mathbf{F_1} + \mathbf{F_2}$	0.918	0.919	0.957	0.031	0.887	0.887	0.922	0.053	0.903	0.921	0.945	0.037
$\mathbf{F_1} + \mathbf{F_3}$	0.921	0.922	0.960	0.029	0.885	0.880	0.920	0.053	0.903	0.922	0.946	0.037
$\mathbf{F_1} + \mathbf{F_2} + \mathbf{F_3} + \mathbf{F_4}$	**0.921**	**0.924**	**0.960**	**0.028**	0.890	**0.890**	0.925	0.051	**0.904**	0.921	0.946	0.036
$\mathbf{F_1} + \mathbf{F_4}$ (**Ours**)	0.920	0.922	0.960	0.029	**0.892**	0.888	**0.926**	**0.049**	0.902	**0.922**	**0.946**	**0.035**

The best results are highlighted in bold.

of low-quality depth maps. Here, to further investigate its effectiveness, we conduct three residual connection manners (i.e., $\mathbf{FE}_k + \mathbf{F}_{rgb}$, $\mathbf{FE}_k + \mathbf{F}_d$, and $\mathbf{FE}_k + \mathbf{F}_{rgb}$ and $\mathbf{FE}_k + \mathbf{F}_d$). The structures of three manners are shown in Fig. 11.

Therefore, we also conduct ablation studies based on the three manners. In the case of two low-quality depth maps, the detection effects of the three methods are compared as shown in Fig. 12. For the two scenes, the low-quality depth maps fail to provide valuable distance cues. However, it can be seen from the figure that using the first method, the detection effect will be better. Therefore, the obtained $\mathbf{FE}_k$ feature is added with $\mathbf{FE}_{rgb}$ feature, not with others.

e) Effects of the different inputs on edge-guided module: We consider learning edge features from the perspective of inputs. Previous works [18], [37] usually use low-level features to learn edge representations. However, low-level features may introduce many non-object edge features [71]. Therefore, we use low-level and high-level features to learn edge features, which can fuse pixel-level and semantic-level information simultaneously. Here, we perform multiple ablation studies to demonstrate the effectiveness of the inputs. The results are presented in Table IV below.

The above results show that the combination of low-level features and high-level features can achieve better results than the combination of lower-level features $\mathbf{F_1} + \mathbf{F_2}$, and the combination of $\mathbf{F_1} + \mathbf{F_2} + \mathbf{F_3} + \mathbf{F_4}$ and $\mathbf{F_1} + \mathbf{F_4}$ both achieve better performance. However, compared to the combination

TABLE V
EFFECTS OF THE WEIGHT-SHARED STRATEGY BETWEEN THE HMCMF MODULES

Methods	STERE [75]				LFSD [77]				SIP [14]			
	$S_\alpha\uparrow$	$F_\beta\uparrow$	$E_\xi\uparrow$	MAE↓	$S_\alpha\uparrow$	$F_\beta\uparrow$	$E_\xi\uparrow$	MAE↓	$S_\alpha\uparrow$	$F_\beta\uparrow$	$E_\xi\uparrow$	MAE↓
No-weight-shared	0.920	0.921	0.959	0.029	**0.894**	**0.896**	**0.928**	0.049	0.901	0.919	0.944	0.037
weight-shared (**Ours**)	**0.920**	**0.922**	**0.960**	**0.029**	0.892	0.888	0.926	**0.049**	**0.902**	**0.922**	**0.946**	**0.035**

The best results are highlighted in bold.

TABLE VI
ABLATION RESULTS OF DIFFERENT BACKBONES

Methods	RGB	Depth	Input size	STERE [75]				LFSD [77]				SIP [14]			
				$S_\alpha\uparrow$	$F_\beta\uparrow$	$E_\xi\uparrow$	MAE↓	$S_\alpha\uparrow$	$F_\beta\uparrow$	$E_\xi\uparrow$	MAE↓	$S_\alpha\uparrow$	$F_\beta\uparrow$	$E_\xi\uparrow$	MAE↓
Transformer-Based Methods															
TriTrans$_{MM21}$ [19]	Res-50 [96] ViT-B [17]	Res-50 [96] ViT-B [17]	256 × 256	0.908	0.911	0.953	0.033	0.866	0.870	0.905	0.066	0.886	0.899	0.930	0.043
VST$_{ICCV21}$ [20]	T2T-ViT-14 [53]	T2T-ViT-14 [53]	224 × 224	0.913	0.907	0.951	0.038	0.882	0.889	0.921	0.061	**0.904**	0.915	0.944	0.040
Ours	T2T-ViT-14 [53]	LDNet	224 × 224	0.908	0.907	0.952	0.035	0.880	0.883	0.921	0.058	0.904	0.910	**0.948**	0.036
	P2T-B [51]	LDNet	256 × 256	0.920	0.918	0.956	0.029	0.888	0.884	0.925	0.051	0.894	0.913	0.941	0.040
	PVT-V2-B [48]	LDNet	256 × 256	0.918	0.917	0.957	0.030	0.891	**0.893**	**0.930**	**0.047**	0.897	0.912	0.940	0.040
	Swin V2-B [50]	LDNet	256 × 256	**0.920**	**0.922**	**0.960**	**0.029**	**0.892**	0.888	0.926	0.049	0.902	**0.922**	0.946	**0.035**
CNN-Based Methods															
DSA^{2}F$_{CVPR21}$	VGG-19 [66]	DepthNet [36]	256 × 256	0.897	0.910	0.942	0.039	**0.883**	**0.889**	**0.924**	**0.055**	0.862	0.891	0.911	0.057
SPNet$_{ICCV21}$ [92]	Res2-50 [97]	Res2-50 [97]	**352 × 352**	0.907	**0.915**	0.944	0.037	-	-	-	-	0.894	0.916	0.930	0.043
DCMF$_{TIP22}$ [46]	VGG-16 [66]	DepthNet [46]	256 × 256	0.910	0.906	0.946	0.043	0.878	0.875	0.909	0.068	0.788	0.772	0.855	0.107
C^{2}DFNet$_{TMM22}$ [94]	Res-50 [96]	Res-50 [96]	256 × 256	0.902	0.897	0.943	0.038	0.863	0.867	0.903	0.065	0.871	0.877	0.916	0.053
SPSN$_{ECCV22}$ [95]	VGG-16 [66]	VGG-16 [66]	**352 × 352**	0.906	0.902	0.945	0.036	-	-	-	-	0.892	0.900	0.936	0.043
Ours	VGG-16 [66]	LDNet	256 × 256	0.902	0.898	0.941	0.039	0.856	0.856	0.895	0.071	0.884	0.889	0.926	0.048
	Res-50 [96]	LDNet	256 × 256	0.907	0.905	0.948	0.036	0.870	0.868	0.909	0.062	0.887	0.897	0.929	0.044
	VGG-19 [66]	LDNet	256 × 256	0.903	0.898	0.943	0.040	0.866	0.868	0.911	0.064	0.887	0.896	0.929	0.046
	Res2-50 [97]	LDNet	256 × 256	**0.913**	0.912	**0.952**	**0.035**	0.873	0.866	0.909	0.060	**0.896**	**0.913**	**0.941**	**0.041**
△ gains (Transformer *vs* CNN)				0.77%	1.10%	0.84%	17.14%	2.18%	2.54%	1.87%	18.33%	0.67%	0.99%	0.53%	14.63%

Swin V2-B, PVT-V2-B, and P2T-B denote the swin V2 base, PVT-V2-B2-linear, and P2T-base backbones, respectively. "Res-50" and "Res2-50" denote the ResNet-50 and Res2Net-50 backbones, respectively. "△ gains" represents the performance gains of the best result obtained with swin compared to the best result obtained with CNN-based backbones.
The best results of each column are highlighted in bold.

TABLE VII
ABLATION RESULTS OF ASYMMETRIC AND SYMMETRIC ARCHITECTURES

Architecture	RGB	Depth	STERE [75]				LFSD [77]				SIP [14]			
			$S_\alpha\uparrow$	$F_\beta\uparrow$	$E_\xi\uparrow$	MAE↓	$S_\alpha\uparrow$	$F_\beta\uparrow$	$E_\xi\uparrow$	MAE↓	$S_\alpha\uparrow$	$F_\beta\uparrow$	$E_\xi\uparrow$	MAE↓
Symmetric	Swin V2-B [50]	Swin V2-B [50]	0.918	0.921	0.959	0.030	0.886	0.884	0.923	0.053	**0.905**	**0.922**	0.946	0.036
Asymmetric (**Ours**)	Swin V2-B [50]	LDNet	**0.920**	**0.922**	**0.960**	**0.029**	**0.892**	**0.888**	**0.926**	**0.049**	0.902	0.922	**0.946**	**0.035**
Symmetric	Res-50 [96]	Res-50 [96]	0.897	0.901	0.947	0.039	**0.872**	**0.876**	**0.915**	**0.060**	**0.890**	**0.902**	**0.932**	0.044
Asymmetric (**Ours**)	Res-50 [96]	LDNet	**0.907**	**0.905**	**0.948**	**0.036**	0.870	0.868	0.909	0.062	0.887	0.897	0.929	**0.044**
Symmetric	VGG-16 [66]	VGG-16 [66]	0.894	0.890	0.937	0.044	0.847	0.844	0.886	0.075	**0.887**	**0.896**	**0.927**	0.049
Asymmetric (**Ours**)	VGG-16 [66]	LDNet	**0.902**	**0.898**	**0.941**	**0.039**	**0.856**	**0.856**	**0.895**	**0.071**	0.884	0.889	0.926	**0.048**

Swin V2-B denotes the swin V2 base backbone. "Res-50" denotes the ResSnet-50 backbones.
The best results of each column are highlighted in bold.

of $\mathbf{F_1}+\mathbf{F_2}+\mathbf{F_3}+\mathbf{F_4}$, the combination of $\mathbf{F_1}+\mathbf{F_4}$ consumes less computational cost, so we choose the combination of $\mathbf{F_1}+\mathbf{F_4}$.

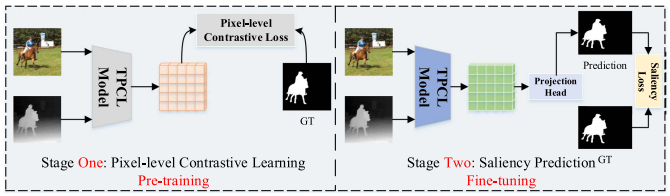
f) Effects of the weight-shared strategy between the HMCMF moudles: In our work, to reduce the parameters as much as possible, we implement a weight-shared strategy between the first and second HMCMF modules and another weight-shared strategy between the third and fourth HMCMF modules. To demonstrate the effectiveness of this strategy, we add an ablation study (with no weight shared between these modules), i.e., using four independent HMCMF modules. The results are shown in Table V below, which proves the effectiveness of the strategy and basically achieves consistent performance without increasing extra parameters.

2) Effects of Different Backbones: We conduct multiple ablation studies with different backbones to explore further the effects of our proposed LDNet and components, which demonstrate our methods can support different backbones. Specifically, we replace our backbones with some representative Transformer-based and CNN-based backbones for comparison with our results. For Transformer-based backbones, we choose PVT-V2-B2-Linear [48], P2T-Base [51]. For the CNN-based backbones, we choose the ResNet-50 [96], Res2Net-50 [97], VGG-16 [66], and VGG-19 [66] for comparison. Note that since the feature dimensions extracted by each backbone are different, we adjust the hyper-parameters (e.g., hidden layer dimension and patch size) and retrain all Transformer-based and CNN-based models to ensure optimal performance. The results are presented in Table VI below.

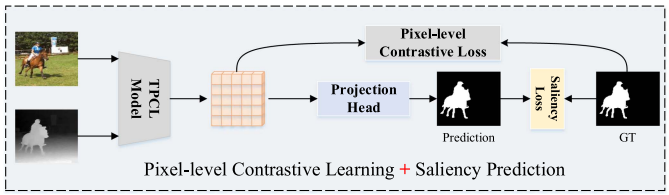
As can be observed from the quantitative results in Table VI, the Transformer-based backbones perform better than the CNN-based backbones. The results demonstrate that Transformer performs feature extraction and long-range modeling dependencies better than CNN. Specifically, in terms of comparison of different Transformer-based backbones (see the 7th, 8th, and 9th rows of Table VI), the Swin V2 version of our method slightly outperforms the PVT-V2 and P2T versions. The results indicate that Swin Transformer can capture more useful information compared with PVT-V2 and P2T in RGB-D SOD tasks. Furthermore, it is worth mentioning that the time consumption of P2T version is more than twice as much as the PVT-V2 and Swin V2 versions to train our proposed methods. The possible reason is that P2T uses a lighter *Hardwish* activation and training memory usage is an additional 52% compared with the typical *GELU* activation [51]. Moreover, to make a fair comparison, we follow the approach of VST [20] by using T2T-ViT-14 as the backbone network to extract the two scale features of the stem and the output of the last two layers and then incorporate our method to provide the experimental results. The results demonstrate that our method incorporated with T2T-ViT-14 achieves comparable performance with fewer parameters (61.45 M vs. 83.83 M) than VST. Notably, all of our versions based on the Transformer architecture exhibit superior performance compared to other state-of-the-art Transformer-based models, such as TriTrans [19], which demonstrate the effectiveness and flexibility of our method. Moreover, from the comparison of CNN-based methods (see the last four rows of Table VI), we can see that our method outperforms other state-of-the-art methods, except that the metrics on the LFSD dataset dropping slightly when compared to DSA^{2}F [65] but is higher than DCMF [46] and C^{2}DFNet [94]. The results demonstrate that our proposed components can cooperate with CNN-based backbones better and further verify the scalability and superiority of our method.

3) Effects of Asymmetric and Symmetric Architectures: We conduct several ablation experiments to investigate the effectiveness of our approach using both asymmetric and symmetric architectures and demonstrate the effectiveness of the proposed lightweight backbone LDNet. We selected Swin V2-B, ResNet-50, and VGG-16 as our backbones, and the results of these experiments are presented in the following Table VII. Note that since the feature dimensions extracted by each backbone are different, we adjust the hyper-parameters (e.g., hidden layer dimension and patch size) and retrain all CNN-based models to ensure optimal performance.

In terms of comparison of asymmetric and symmetric architectures, both Transformer-based and CNN-based asymmetric and symmetric architectures achieve comparable performance on the datasets, which demonstrates that our method works for



(a) Two-stage Training



(b) One-stage Training

Fig. 13. **Two kinds of training strategies.** (a) Two-stage training is divided into PCL pre-training and saliency prediction fine-tuning. (b) One-stage training synchronizes PCL and saliency prediction (**our method in this study**). The projection head denotes 1×1 convolutional layer.

TABLE VIII
COMPARISON RESULTS OF TWO DIFFERENT TRAINING STRATEGIES

Training strategy	**STERE** [75]				**LFSD** [77]				**SIP** [14]			
	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	MAE $\downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	MAE $\downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	MAE $\downarrow$
One-stage (ours)	**0.920**	**0.922**	**0.960**	**0.029**	**0.892**	**0.888**	0.926	**0.049**	0.902	**0.922**	0.946	**0.035**
Two-stage	0.918	0.919	0.957	0.031	0.891	0.887	**0.927**	0.051	**0.903**	0.919	**0.947**	0.036

both architectures. Moreover, The comparisons demonstrate the effectiveness of our proposed LDNet. It further indicates that exploring a lightweight random initialized backbone can extract depth features to incorporate with RGB features for a better saliency prediction. Based on the above analysis, we consider that it is unnecessary to utilize a large network for depth feature extraction, and it can be replaced with a lightweight network.

4) Effects of Different Training Strategies: In general, contrastive learning is used for visual representation learning, resulting in a robust pre-trained model for various downstream tasks (e.g., image classification [32], [33]). Therefore, we investigate the impacts of two different training strategies on RGB-D SOD, which is shown in Fig. 13.

The first strategy we proposed is the two-stage training strategy, which first pre-trains the entire model with our PCL loss $\mathcal{L}_{PCL}$, and then fine-tunes the pre-trained model with the saliency loss $\mathcal{L}_S$ (see Fig. 13(a)).

The second strategy is the training method used in our work, which employs a one-stage strategy for training. To be specific, the strategy uses the contrastive loss $\mathcal{L}_{PCL}$ and saliency loss $\mathcal{L}_S$ simultaneously to train the entire model (see Fig. 13(b)).

We report the comparison results of two kinds of training strategies of our proposed model in Table VIII. From Table VIII, it can be observed that one-stage training achieves a slight performance improvement than two-stage training. We consider the possible reason for this is that the two-stage strategy may pull away the pixel embedding distribution (which contains inter-pixel discrimination and intra-pixel compactness) obtained in the first stage, while the one-stage strategy does not. Similar one-stage training methods have identical applications and conclusions in other contrastive learning tasks [31], [60]. Additionally, we show four visual saliency examples of two-stage training to demonstrate the effectiveness of PCL and two-stage training. Specifically, after the first pre-training stage, we add an **untrained projection head** (1×1 convolutional layer) to predict the saliency map.

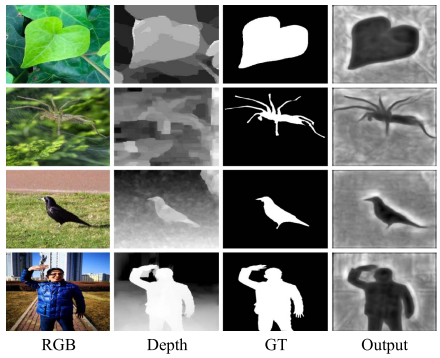


Fig. 14. Visualization of saliency maps output by an untrained projection head (1×1 convolutional layer).

TABLE IX
COMPARISONS WITH SOTAS RGB-T SOD METHODS ON THREE DATASETS

Dataset	Metric	MIDD$_{21}$	[100]ECFFNet$_{21}$	[101]CSRNet$_{21}$	[102]TNet$_{22}$	[103] Ours
VT821 [106]	$S_\alpha \uparrow$	0.871	0.877	0.885	**0.899**	0.896
	$F_\beta \uparrow$	0.851	0.834	0.858	**0.888**	0.884
	$E_\xi \uparrow$	0.918	0.910	0.923	0.938	**0.939**
	MAE$\downarrow$	0.045	0.034	0.038	**0.030**	0.032
VT1000 [104]	$S_\alpha \uparrow$	0.915	0.923	0.918	0.929	**0.936**
	$F_\beta \uparrow$	0.913	0.917	0.908	0.930	**0.940**
	$E_\xi \uparrow$	0.957	0.959	0.953	0.966	**0.974**
	MAE$\downarrow$	0.027	0.021	0.024	0.021	**0.016**
VT5000 [105]	$S_\alpha \uparrow$	0.868	0.874	0.868	0.895	**0.905**
	$F_\beta \uparrow$	0.849	0.848	0.837	0.881	**0.895**
	$E_\xi \uparrow$	0.920	0.921	0.914	0.937	**0.951**
	MAE$\downarrow$	0.043	0.038	0.042	0.033	**0.026**

The best results are highlighted in bold.

Fig. 14 displays four examples of saliency maps output by an untrained projection head (the 4th column is the obtained output). It can be clearly seen that the pre-trained model trained using only the PCL loss $\mathcal{L}_{PCL}$ is able to detect salient objects well, even including some details, which further demonstrates the effectiveness of our proposed pixel-level contrastive learning.

D. Application to RGB-T SOD Task

Recently, different Transformer fusion methods for RGB-X semantic segmentation tasks have emerged and achieved superior performance [99]. To this end, to demonstrate the generalization performance of our method, we also apply TPCL for RGB-Thermal (RGB-T) SOD. The comparison results with recent SOTAs [100], [101], [102], [103] prove that our method can also be applied to the RGB-X SOD tasks. The performance comparisons are shown in Table IX below, and the results confirm the effectiveness of our method. Our method performs the best on the VT1000 [104] and VT5000-test [105] datasets, which demonstrates the robustness and generalizability of TPCL for RGB-T SOD tasks.

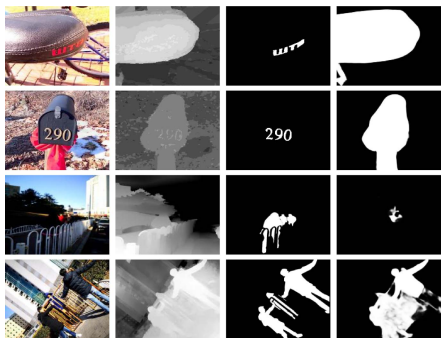


Fig. 15. Visual examples of failure cases.

E. Discussion

1) Failure Cases: we show several representative examples in Fig. 15 below. It is difficult to locate salient objects in the following aspects perfectly: 1) Shows sharp contrast but not salient objects. In the first scene (See the top two rows of the figure), although the foreground objects have sharp contrast with the background and have high-quality depth maps, they are not really salient objects. Therefore, this ambiguity makes it difficult for our model to detect red signs and yellow number symbols. 2) Salient objects are occluded, especially in low-light conditions (See the bottom two rows of the figure). Our model fails to accurately detect multiple salient objects when the objects are occluded by the other objects (e.g., the fence). In particular, the third example shows how difficult it is for the model to recognize salient objects when the objects are in low-light conditions. However, our model is biased toward detecting a person in brighter regions. Our model is also unfriendly to salient objects in low-light regions.

2) Limitations and Future Works: we have identified three limitations that warrant further investigation. Firstly, while our proposed Pixel-level Contrastive Learning (PCL) has demonstrated promising results, it currently relies on pixel-level labels and is not applicable to RGB-D SOD for semi-supervised or unsupervised learning. Recent studies have shown that there is a growing interest in unsupervised or semi-supervised RGB-D SOD, and therefore, it is imperative to explore how our PCL approach can be adapted to accommodate these scenarios. Secondly, our approach has only utilized contrastive learning to discriminate pixel-level feature representations and has not taken into account the semantic-level feature representations. In other words, for the high-level semantic features extracted from backbones, we believe that the two modalities should be consistent at the semantic level. Therefore, contrastive learning can be used to pull the semantic-level feature representations of the two modalities closer. We argue that a shared backbone can be used for feature extraction when pulling the semantic-level feature representations closer. Thirdly, our proposed model is data-driven, meaning that its generalization ability is enhanced with larger amounts of data. Specifically, our results show that the model's performance in Setting B is superior to that of Setting A. In short, using contrastive learning and Transformer for RGB-D SOD is still worth exploring.

V. Conclusion

In this article, we propose a TPCL network to boost RGB-D SOD via Transformer fusion and pixel-level contrastive learning. Different from previous fusion methods, we exploit Transformer to capture multi-modality interactions better, which generates more comprehensive fusion features. Additionally, in terms of intra- and inter-pixel relationships, although an RGB-D SOD model trained by cross-entropy loss can achieve superior performance, it cannot obtain a well-structured and robust embedding space. Therefore, we propose a novel PCL algorithm that preserves intra-pixel compactness and inter-pixel discrimination in an embedding space. Experimental results on seven public datasets demonstrate the effectiveness of our method. In the future, we plan to explore a contrastive learning method for RGB-D representation learning and improve our PCL for RGB SOD.

References

[1] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 569–582, Mar. 2015.
[2] H. Jiang et al., "Salient object detection: A discriminative regional feature integration approach," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, 2013, pp. 2083–2090.
[3] W. Zhu, S. Liang, Y. Wei, and J. Sun, "Saliency optimization from robust background detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 2814–2821.
[4] W. Wang et al., "Salient object detection in the deep learning era: An in-depth survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 6, pp. 3239–3259, Jun. 2022.
[5] Z. Ren, S. Gao, L.-T. Chia, and I. W.-H. Tsang, "Region-based saliency detection and its application in object recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 5, pp. 769–779, May 2014.
[6] Y. Gao et al., "Visual-textual joint relevance learning for tag-based social image search," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 363–376, Jan. 2013.
[7] M.-M. Cheng, F.-L. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu, "Repfinder: Finding approximately repeated scene elements for image editing," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 1–8, 2010.
[8] V. Mahadevan and N. Vasconcelos, "Saliency-based discriminant tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1007–1013.
[9] J. Ren, X. Gong, L. Yu, W. Zhou, and M. Y. Yang, "Exploiting global priors for RGB-D saliency detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2015, pp. 25–32.
[10] L. Qu et al., "RGBD salient object detection via deep fusion," *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2274–2285, May 2017.
[11] H. Chen, Y.-F. Li, and D. Su, "M3net: Multi-scale multi-path multimodal fusion network and example application to RGB-D salient object detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 4911–4916.
[12] H. Chen and Y. Li, "Progressively complementarity-aware fusion network for RGB-D salient object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3051–3060.
[13] J.-X. Zhao et al., "Contrast prior and fluid pyramid integration for RGBD salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3927–3936.
[14] 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 Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 5, pp. 2075–2089, May 2021.
[15] K. Fu et al., "Siamese network for RGB-D salient object detection and beyond," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5541–5559, Sep. 2022.
[16] A. Vaswani et al., "Attention is all you need," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30, pp. 6000–6010.
[17] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[18] Z. Liu, Y. Tan, Q. He, and Y. Xiao, "SwinNet: Swin transformer drives edge-aware RGB-D and RGB-T salient object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 7, pp. 4486–4497, Jul. 2022.

[19] Z. Liu, Y. Wang, Z. Tu, Y. Xiao, and B. Tang, "TritransNet: RGB-D salient object detection with a triplet transformer embedding network," in *Proc. 29th ACM Int. Conf. Multimedia*, 2021, pp. 4481–4490.

[20] N. Liu, N. Zhang, K. Wan, L. Shao, and J. Han, "Visual saliency transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 4722–4732.

[21] T. Zhou, D.-P. Fan, M.-M. Cheng, J. Shen, and L. Shao, "RGB-D salient object detection: A survey," *Comput. Vis. Media*, vol. 7, no. 1, pp. 37–69, 2021.

[22] H. Song et al., "Depth-aware salient object detection and segmentation via multiscale discriminative saliency fusion and bootstrap learning," *IEEE Trans. Image Process.*, vol. 26, no. 9, pp. 4204–4216, Sep. 2017.

[23] Z. Zhang et al., "Bilateral attention network for RGB-D salient object detection," *IEEE Trans. Image Process.*, vol. 30, pp. 1949–1961, 2021.

[24] C. Zhu, X. Cai, K. Huang, T. H. Li, and G. Li, "PDNet: Prior-model guided depth-enhanced network for salient object detection," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2019, pp. 199–204.

[25] H. Chen and Y. Li, "Three-stream attention-aware network for RGB-D salient object detection," *IEEE Trans. Image Process.*, vol. 28, no. 6, pp. 2825–2835, Jun. 2019.

[26] J. Han, H. Chen, N. Liu, C. Yan, and X. Li, "CNNs-based RGB-D saliency detection via cross-view transfer and multiview fusion," *IEEE Trans. Cybern.*, vol. 48, no. 11, pp. 3171–3183, Nov. 2018.

[27] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7132–7141.

[28] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.

[29] Z.-Y. Dou et al., "An empirical study of training end-to-end vision-and-language transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 18166–18176.

[30] R. Cong et al., "CIR-Net: Cross-modality interaction and refinement for RGB-D salient object detection," *IEEE Trans. Image Process.*, vol. 31, pp. 6800–6815, 2022.

[31] W. Wang et al., "Exploring cross-image pixel contrast for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 7303–7313.

[32] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9729–9738.

[33] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[34] P. Khosla et al., "Supervised contrastive learning," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 18661–18673.

[35] L. V. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, 2008.

[36] M. Zhang et al., "Asymmetric two-stream architecture for accurate RGB-D saliency detection," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 374–390.

[37] J.-X. Zhao et al., "EGNet: Edge guidance network for salient object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 8779–8788.

[38] K. Desingh, K. M. Krishna, D. Rajan, and C. Jawahar, "Depth really matters: Improving visual salient region detection with depth," in *Proc. Brit. Mach. Vis. Conf.*, 2013, pp. 1–11.

[39] R. Cong et al., "Saliency detection for stereoscopic images based on depth confidence analysis and multiple cues fusion," *IEEE Signal Process. Lett.*, vol. 23, no. 6, pp. 819–823, Jun. 2016.

[40] G. Li et al., "Hierarchical alternate interaction network for RGB-D salient object detection," *IEEE Trans. Image Process.*, vol. 30, pp. 3528–3542, 2021.

[41] C. Li et al., "ASIF-Net: Attention steered interweave fusion network for RGB-D salient object detection," *IEEE Trans. Cybern.*, vol. 51, no. 1, pp. 88–100, Jan. 2021.

[42] X. Zhao, Y. Pang, L. Zhang, H. Lu, and X. Ruan, "Self-supervised pretraining for RGB-D salient object detection," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, pp. 3463–3471.

[43] X. Wang et al., "Boosting RGB-D saliency detection by leveraging unlabeled RGB images," *IEEE Trans. Image Process.*, vol. 31, pp. 1107–1119, 2022.

[44] T. Yang, Y. Wang, L. Zhang, J. Qi, and H. Lu, "Depth-inspired label mining for unsupervised RGB-D salient object detection," in *Proc. 30th ACM Int. Conf. Multimedia*, 2022, pp. 5669–5677.

[45] W. Ji et al., "Promoting saliency from depth: Deep unsupervised RGB-D saliency detection," in *Proc. Int. Conf. Learn. Representations*, 2022.

[46] F. Wang, J. Pan, S. Xu, and J. Tang, "Learning discriminative cross-modality features for RGB-D saliency detection," *IEEE Trans. Image Process.*, vol. 31, pp. 1285–1297, 2022.

[47] W. Wang et al., "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 568–578.

[48] W. Wang et al., "PVT V2: Improved baselines with pyramid vision transformer," *Comput. Vis. Media*, vol. 8, no. 3, pp. 415–424, 2022.

[49] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10012–10022.

[50] Z. Liu et al., "Swin transformer V2: Scaling up capacity and resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 12009–12019.

[51] Y.-H. Wu, Y. Liu, X. Zhan, and M.-M. Cheng, "P2T: Pyramid pooling transformer for scene understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Aug. 30, 2022, doi: 10.1109/TPAMI.2022.3202765.

[52] H. Touvron et al., "Training data-efficient image transformers & distillation through attention," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 10347–10357.

[53] L. Yuan et al., "Tokens-to-token vit: Training vision transformers from scratch on imagenet," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 558–567.

[54] X. Chu et al., "Conditional positional encodings for vision transformers," 2021, *arXiv:2102.10882*.

[55] H. Wu et al., "CVT: Introducing convolutions to vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 22–31.

[56] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3733–3742.

[57] M. Ye, X. Zhang, P. C. Yuen, and S.-F. Chang, "Unsupervised embedding learning via invariant and spreading instance feature," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6210–6219.

[58] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.

[59] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 776–794.

[60] B. Gunel, J. Du, A. Conneau, and V. Stoyanov, "Supervised contrastive learning for pre-trained language model fine-tuning," 2020, *arXiv:2011.01403*.

[61] J. Deng et al., "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[62] M. A. Islam, S. Jia, and N. D. Bruce, "How much position information do convolutional neural networks encode?," 2020, *arXiv:2001.08248*.

[63] T. V. Dijk and G. D. Croon, "How do neural networks see depth in single images?," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 2183–2191.

[64] K. Wang, Y. Chen, H. Guo, L. Wen, and S. Shen, "Geometric pretraining for monocular depth estimation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 4782–4788.

[65] P. Sun, W. Zhang, H. Wang, S. Li, and X. Li, "Deep RGB-D saliency detection with depth-sensitive attention and automatic multi-modal fusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 1407–1417.

[66] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[67] W. Zhang, G.-P. Ji, Z. Wang, K. Fu, and Q. Zhao, "Depth quality-inspired feature manipulation for efficient RGB-D salient object detection," in *Proc. 29th ACM Int. Conf. Multimedia*, 2021, pp. 731–740.

[68] S. Chen and Y. Fu, "Progressively guided alternate refinement network for RGB-D salient object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 520–538.

[69] 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 *Proc. 16th Eur. Conf. Comput. Vis.*, Glasgow, U.K., 2020 pp. 275–292.

[70] Y. Yang et al., "Bi-directional progressive guidance network for RGB-D salient object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 8, pp. 5346–5360, Aug. 2022.

[71] Y. Sun, S. Wang, C. Chen, and T.-Z. Xiang, "Boundary-guided camouflaged object detection," 2022, *arXiv:2207.00794*.
[72] Y. Pang, L. Zhang, X. Zhao, and H. Lu, "Hierarchical dynamic filtering network for RGB-D salient object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 235–252.
[73] E. Xie et al., "Segmenting transparent objects in the wild," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 696–711.
[74] R. Ju, L. Ge, W. Geng, T. Ren, and G. Wu, "Depth saliency based on anisotropic center-surround difference," in *Proc. IEEE Int. Conf. Image Process.*, 2014, pp. 1115–1119.
[75] Y. Niu, Y. Geng, X. Li, and F. Liu, "Leveraging stereopsis for saliency analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 454–461.
[76] Y. Piao, W. Ji, J. Li, M. Zhang, and H. Lu, "Depth-induced multi-scale recurrent attention network for saliency detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 7254–7263.
[77] N. Li, J. Ye, Y. Ji, H. Ling, and J. Yu, "Saliency detection on light field," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 2806–2813.
[78] H. Peng, B. Li, W. Xiong, W. Hu, and R. Ji, "RGBD salient object detection: A benchmark and algorithms," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 92–109.
[79] Y. Cheng, H. Fu, X. Wei, J. Xiao, and X. Cao, "Depth enhanced saliency detection method," in *Proc. Int. Conf. Internet Multimedia Comput. Serv.*, 2014, pp. 23–27.
[80] G. Li, Z. Liu, L. Ye, Y. Wang, and H. Ling, "Cross-modal weighting network for RGB-D salient object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 665–681.
[81] A. Luo et al., "Cascade graph neural networks for RGB-D salient object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 346–364.
[82] M. Zhang, W. Ren, Y. Piao, Z. Rong, and H. Lu, "Select, supplement and focus for RGB-D saliency detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3472–3481.
[83] W. Ji, J. Li, M. Zhang, Y. Piao, and H. Lu, "Accurate RGB-D salient object detection via collaborative learning," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 52–69.
[84] W. Ji et al., "Calibrated RGB-D salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 9471–9481.
[85] D.-P. Fan, M.-M. Cheng, Y. Liu, T. Li, and A. Borji, "Structure-measure: A new way to evaluate foreground maps," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4548–4557.
[86] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1597–1604.
[87] D.-P. Fan et al., "Enhanced-alignment measure for binary foreground map evaluation," 2018, *arXiv:1805.10421*.
[88] A. Borji, M.-M. Cheng, H. Jiang, and J. Li, "Salient object detection: A benchmark," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5706–5722, Dec. 2015.
[89] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2017, *arXiv:1711.05101*.
[90] Y. Piao, Z. Rong, M. Zhang, W. Ren, and H. Lu, "A2Dele: Adaptive and attentive depth distiller for efficient RGB-D salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9060–9069.
[91] W.-D. Jin, J. Xu, Q. Han, Y. Zhang, and M.-M. Cheng, "CDNet: Complementary depth network for RGB-D salient object detection," *IEEE Trans. Image Process.*, vol. 30, pp. 3376–3390, 2021.
[92] T. Zhou et al., "Specificity-preserving RGB-D saliency detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 4681–4691.
[93] Q. Chen et al., "RGB-D salient object detection via 3D convolutional neural networks," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, pp. 1063–1071.
[94] M. Zhang, S. Yao, B. Hu, Y. Piao, and W. Ji, "C^2DFNet: Criss-cross dynamic filter network for RGB-D salient object detection," *IEEE Trans. Multimedia*, early access, Jul. 01, 2022, doi: 10.1109/TMM.2022.3187856.
[95] M. Lee, C. Park, S. Cho, and S. Lee, "SPSN: Superpixel prototype sampling network for RGB-D salient object detection," 2022, *arXiv:2207.07898*.
[96] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[97] S.-H. Gao et al., "Res2Net: A new multi-scale backbone architecture," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 2, pp. 652–662, Feb. 2021.
[98] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6299–6308.
[99] H. Liu, J. Zhang, K. Yang, X. Hu, and R. Stiefelhagen, "CMX: Cross-modal fusion for RGB-X semantic segmentation with transformers," 2022, *arXiv:2203.04838*.
[100] Z. Tu, Z. Li, C. Li, Y. Lang, and J. Tang, "Multi-interactive dual-decoder for RGB-thermal salient object detection," *IEEE Trans. Image Process.*, vol. 30, pp. 5678–5691, 2021.
[101] W. Zhou, Q. Guo, J. Lei, L. Yu, and J.-N. Hwang, "ECFFNet: Effective and consistent feature fusion network for RGB-T salient object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 3, pp. 1224–1235, Mar. 2022.
[102] F. Huo, X. Zhu, L. Zhang, Q. Liu, and Y. Shu, "Efficient context-guided stacked refinement network for RGB-T salient object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 5, pp. 3111–3124, May 2021.
[103] R. Cong et al., "Does thermal really always matter for RGB-T salient object detection?," *IEEE Trans. Multimedia*, early access, Oct. 21, 2022, doi: 10.1109/TMM.2022.3216476.
[104] Z. Tu et al., "RGB-T image saliency detection via collaborative graph learning," *IEEE Trans. Multimedia*, vol. 22, no. 1, pp. 160–173, Jan. 2020.
[105] Z. Tu et al., "RGBT salient object detection: A large-scale dataset and benchmark," *IEEE Trans. Multimedia*, early access, May 03, 2022, doi: 10.1109/TMM.2022.3171688.
[106] G. Wang et al., "RGB-T saliency detection benchmark: Dataset, baselines, analysis and a novel approach," in *Proc. Image Graph. Technol. Appl.: 13th Conf. Image Graph. Technol. Appl.*, 2018, pp. 359–369.

Jiesheng Wu (Student Member, IEEE) is currently working toward the Ph.D. degree with the College of Artificial Intelligence, Nankai University, Tianjin, China. His research interests include computer vision, salient object detection, camouflaged object detection, and multi-modal computing.

Fangwei Hao is currently working toward the Ph.D. degree with the College of Artificial Intelligence, Nankai University, Tianjin, China. His main research focuses on image processing based on deep learning.

Weiyun Liang (Student Member, IEEE) is currently working toward the Ph.D. degree with the College of Artificial Intelligence, Nankai University, Tianjin, China. His research interests include computer vision and multimedia computing.

Jing Xu (Member, IEEE) received the Ph.D. degree from Nankai University, Tianjin, China, in 2003. She is currently a Professor with the College of Artificial Intelligence, Nankai University. She has authored or coauthored more than 100 papers in software engineering, software security, and Big Data analytics. She was the recipient of the Second Prize of the Tianjin Science and Technology Progress Award twice in 2017 and 2018, respectively.