This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

VSCode: General Visual Salient and Camouflaged Object Detection with 2D Prompt Learning

Ziyang Luo[1] Nian Liu[2,*] Wangbo Zhao[3] Xuguang Yang[1] Dingwen Zhang[1]
Deng-Ping Fan[5,6] Fahad Khan[2,4] Junwei Han[1,7,*]
[1]Northwestern Polytechnical University [2]Mohamed bin Zayed University of Artificial Intelligence
[3]National University of Singapore [4] CVL, Linköping University [5] TBI Center, CS, Nankai University
[6] Nankai International Advanced Research Institute (SHENZHEN FUTIAN)
[7] Institute of Artificial Intelligence, Hefei Comprehensive National Science Center

Abstract

Salient object detection (SOD) and camouflaged object detection (COD) are related yet distinct binary mapping tasks. These tasks involve multiple modalities, sharing commonalities and unique cues. Existing research often employs intricate task-specific specialist models, potentially leading to redundancy and suboptimal results. We introduce VSCode, a generalist model with novel 2D prompt learning, to jointly address four SOD tasks and three COD tasks. We utilize VST as the foundation model and introduce 2D prompts within the encoder-decoder architecture to learn domain and task-specific knowledge on two separate dimensions. A prompt discrimination loss helps disentangle peculiarities to benefit model optimization. VSCode outperforms state-of-the-art methods across six tasks on 26 datasets and exhibits zero-shot generalization to unseen tasks by combining 2D prompts, such as RGB-D COD. Source code has been available at https://github.com/Sssssuperior/VSCode.

1. Introduction

Visual salient object detection (SOD) and camouflaged object detection (COD) are two interconnected yet unique tasks. The goal of SOD is to identify prominent objects within an image that significantly contrast with their surroundings [5], which can be used to promote segmentation [46, 111], detection [94], and Part-Object Relational visual saliency [57, 58]. While COD focuses on identifying objects concealed within their environment. These objects intentionally blend in by sharing structural or textural similarities with their surroundings [15]. Despite the seemingly different definitions of SOD and COD, they both belong to the realm of binary segmentation and share some vital fundamental similarities, such as objectness and structuredness.

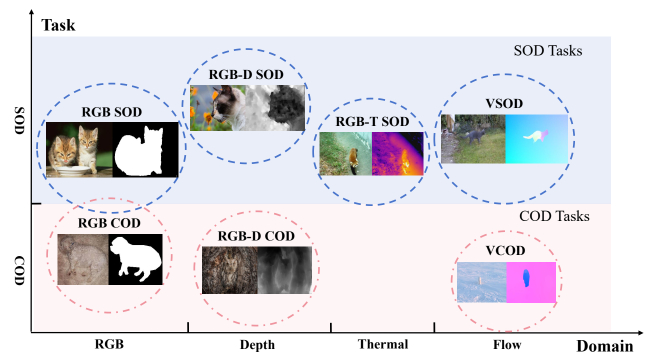

Figure 1. **Relationship of SOD, COD, and multimodal tasks.** Each specific task is seen as a combination of two dimensions, i.e. domain (RGB/Depth/Thermal/Flow) and task (SOD/COD).

To cater to various scenarios, both SOD and COD have given rise to several sub-tasks with different modalities, including RGB SOD [83, 108], RGB COD [27, 66, 101], RGB-D SOD [42, 71], RGB-D COD [92], and RGB-T SOD [80, 107]. By leveraging optical flow maps, Video SOD (VSOD) [44, 91] and VCOD [10, 36] tasks can also be seen as a combination of two modalities. The relationship of SOD, COD, and multimodal tasks is shown in Figure 1, where each specific task can be considered as a combination of two dimensions, i.e. domain and task. Although these multimodal tasks differ in the complementary cues they employ, these modalities share some key commonalities. For instance, depth, thermal, and optical flow maps often show obvious objectness as in RGB images.

Although previous CNN-based [10, 61, 71, 77, 86, 109] and transformer-based [54, 116] approaches have effectively addressed these tasks and achieved favorable results, they usually rely on meticulously designed models to tackle each task individually. Designing models specifically for individual tasks can be problematic since the training data of one task is typically limited. Task-specific specialist models

*Corresponding author: Nian Liu (liunian228@gmail.com) and Junwei Han (junweihan2010@gmail.com)

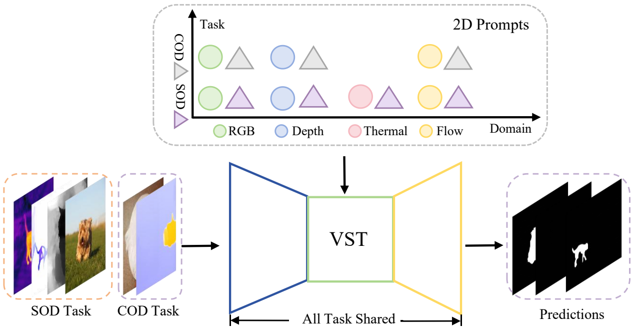

Figure 2. **Overall architecture of our VSCode model.** We use VST [54] as the foundation model to acquire commonalities among multimodal SOD and COD tasks. For each task, we integrate 2D prompts to aggregate peculiarities along the domain dimension and the task dimension, including four domain-specific prompts and two task-specific prompts.

may be overly adapted to a particular task and overfitted to specific training data distribution, which ultimately sacrifices generalization ability and results in suboptimal performance. One solution may be using more data, however, being costly and time-consuming for data annotation. To this end, joint learning a generalist model emerges as a more promising option, as it allows for the maximum use of all data and the effective learning of the commonalities of all tasks, hence significantly reducing the risk of overfitting and enhancing the generalization capability [34, 55]. However, joint learning multiple tasks is not straightforward. On one hand, simultaneously handling both commonalities and peculiarities of all tasks poses a significant challenge as the incompatibility among different tasks easily leads to a decline in performance with simple joint training [40]. On the other hand, it usually introduces additional complexity, computational costs, and parameters.

In this paper, we present a general **V**isual **S**alient and **C**amouflaged **o**bject **de**tection (VSCode) model which encapsulates both commonalities and peculiarities of different tasks with a simple but effective design, as illustrated in Figure 2. On one hand, we adopt VST [54] as the shared segmentation foundation model to assimilate commonalities of different tasks by leveraging its simple and pure-transformer-based architecture. On the other hand, inspired by the recent emergence of the parameter-efficient prompting technique [31, 63, 112], we propose 2D prompts to capture task peculiarities. Specifically, we decompose these peculiarities along the domain dimension and the task dimension, and consequently design domain-specific prompts and task-specific prompts to comprehend the differences among diverse domains and tasks, respectively. These 2D prompts can effectively disentangle domain and task peculiarities, making our model easily adaptable by combining them to tackle specific tasks and even unseen ones. Furthermore, we present a prompt discrimination loss to encourage the 2D prompts to focus on acquiring adequate peculiarities and enable the foundational model to concentrate on commonality learning.

Finally, we train our VSCode model on four SOD tasks and two COD tasks, demonstrating its effectiveness against state-of-the-art methods. What's more, we carry out evaluations on a reserved task and reveal remarkable zero-shot generalization ability of our model, which has never been explored in previous works. The main contributions in this work can be summarized as follows:

- We present VSCode, the first generalist model for multimodal SOD and COD tasks.
- We propose to use a foundation segmentation model to aggregate commonalities and introduce 2D prompts to learn peculiarities along the domain and task dimensions, respectively.
- A prompt discrimination loss is proposed to effectively enhance the learning of peculiarities and commonalities for 2D prompts and the foundation model, respectively.
- Our VSCode model surpasses all existing state-of-the-art models across all tasks on 26 datasets and showcases its ability to generalize to unseen tasks, further emphasizing the superiority of our approach.

2. Related Work

2.1. Deep Learning Based SOD and COD

SOD. Previous RGB SOD works delved into attention-based [9, 19, 49, 71, 108], multi-level fusion-based [20, 24, 67, 83, 102, 113], recurrent-model-based [8, 12, 48, 60, 82], and multi-task-based methods [29, 72, 84, 90, 104, 106, 110]. In the case of RGB-D SOD, some models [9, 42, 43, 52, 53, 71, 109] leveraged various attention mechanisms to incorporate depth cues into RGB features. With regard to RGB-T SOD, recent studies also introduced attention-based methods [77, 80] and multi-level fusion [76, 107] to excavate the relationship between RGB and thermal features. Regarding the VSOD task, some works [18, 21, 30, 74, 86] mined spatial-temporal and appearance cues. More recently, there was a growing trend where various research [28, 44, 51, 73] endeavored to incorporate optical flow for combining motion cues with appearance details. Consistent with recent studies, we treat optical flow as a form of modality information and view VSOD as a multimodal SOD task.

COD. Currently, COD has RGB COD, RGB-D COD, and VCOD tasks. RGB COD methods can be broadly categorized as multi-task-based approaches [61, 101], multi-input-based approaches [66, 114], and refinement-based approaches [15, 32]. The RGB-D COD task was initially introduced in [92], where depth inference models are adapted for object segmentation. For VCOD, prior studies segmented the moving camouflaged objects via dense optical flow [3, 4] or well-designed models [10, 36]. For a more comprehensive literature review, please refer to [16].

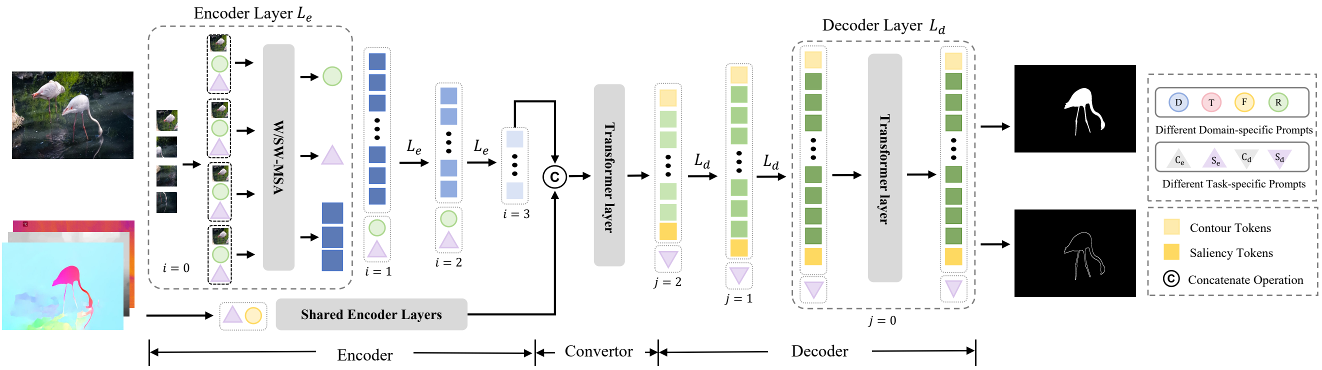

Figure 3. **Overall framework of our proposed VSCode model with 2D prompt learning.** Based on the VST [54] foundation model, we insert the respective domain-specific prompts and task-specific prompts in the attention windows in the Swin transformer [59] encoder layers to learn domain and task-specific encoder features. The convertor is used for multimodal feature fusion. Within the transformer decoder layers, task-specific prompts are appended to image feature tokens to perform task-specific decoding. We also provide detailed structures of an encoder layer ($i = 0$) and a decoder layer ($j = 0$).

2.2. Prompt in Computer Vision

Prompt was initially introduced in the field of NLP [6] and has been successfully integrated into computer vision tasks [25]. VPT [31] introduced a small amount of trainable parameters as prompts in the input space. ViPT [115] put forth the idea of modality-complementary prompts for task-oriented multi-modal tracking. Prior research has primarily focused on specific tasks, such as classification or tracking. In this paper, we propose to use 2D prompts for assembling different multimodal tasks and enable zero-shot generalization on unseen tasks, which has not been explored before.

2.3. Generalist Segmentation Architecture

Recently, several generalist frameworks have emerged for a range of segmentation tasks using a variety of prompts [1]. On one hand, X-Decoder [117] utilized generic non-semantic queries and semantic queries to decode different pixel-level and token-level outputs. UNINEXT [95] introduced three types of prompts, namely category names, language expressions, and reference annotations. On the other hand, Painter [87] and SegGPT [88] leveraged image-mask pairs from the same task as prompts. Unlike the approaches mentioned above, which mainly concentrate on task differences, our VSCode dissects unique characteristics based on both domain and task dimensions, leading to a more versatile design.

In the field of SOD and COD, EVP [56] introduced adaptors into the encoder and trained each task individually for various foreground segmentation tasks. Different from them, we consider not only multiple tasks but also multiple modalities and we train all tasks simultaneously.

3. Methodology

In this work, we propose VSCode with the aim of jointly training SOD and COD tasks in an efficient and effective way. We allow VST [54] to incorporate commonalities (Section 3.1), and utilize 2D prompts, which comprise domain-specific (Section 3.2) and task-specific prompts (Section 3.3), to encapsulate peculiarities. To accurately disentangle domain and task peculiarities in 2D prompts and encourage commonality learning in VST, we introduce a prompt discrimination loss (Section 3.5). Figure 3 shows the overall architecture of our proposed VSCode.

3.1. Foundation Model

To achieve a more comprehensive integration of commonalities from SOD and COD tasks, we select VST [54] as our fundamental model. VST was originally proposed for RGB and RGB-D SOD and comprises three primary components, i.e. a transformer encoder, a transformer convertor, and a multi-task transformer decoder. It initially employs the transformer encoder to capture long-range dependencies within the image features $\boldsymbol{f}_i^E \in \mathbb{R}^{l_i \times c_i}$, where $i \in [0, 1, 2, 3]$ indicates the index of blocks in the encoder, l_i and c_i mean the length of the patch sequence and the channel number of $\boldsymbol{f}_i^E$. Subsequently, the transformer convertor integrates the complement between RGB and depth features via cross-attention for RGB-D SOD or uses self-attention for RGB SOD. In the decoder, which is composed of a sequence of self-attention layers, VST predicts saliency maps and boundary maps simultaneously via a saliency token, a boundary token, and decoder features $\boldsymbol{f}_j^D \in \mathbb{R}^{l_j \times d}$, where j corresponds the index of blocks in the decoder. Here $j \in [2, 1, 0]$ for descending order and d =384. Due to the simple and pure-transformer-based architecture, VST can be easily used for other multimodal tasks and COD tasks without the need for model redesign. As a result, it emerges as a superior choice for constructing a generalist model for general multimodal SOD and COD.

In pursuit of improved outcomes and a more suitable structure, we introduce modifications to VST. First, we select Swin transformer [59] as our backbone due to its efficiency and high performance. Second, to maintain a unified structure for both RGB tasks and other multimodal tasks, we utilize the RGB convertor in VST, which comprises standard transformer layers. For multimodal tasks, we simply concatenate the supplementary modality's features with the RGB features along the channel dimension and employ a multilayer perceptron (MLP) to project them from $2d$ channels to d channels. For RGB tasks, no alterations are made. Third, we incorporate certain extensions from VST++ [50], specifically including the token-supervised prediction loss.

3.2. Domain-specific Prompt

Within the encoder, lower layers are dedicated to extracting low-level features, encompassing edges, colors, and textures, which exhibit distinct characteristics in various domains [100]. For instance, depth maps are typically rendered in grayscale, while thermal maps present a broader color spectrum. Higher layers, on the other hand, capture semantic information from modality features, which is crucial for all tasks. Consequently, we introduce domain-specific prompts $\boldsymbol{p}_i^d$ at each block i in the encoder and design four kinds of domain-specific prompts for RGB, depth, thermal, and optical flow, respectively, to highlight the disparities among domains, as shown in Figure 3.

Given the image features $\boldsymbol{f}_i^E$ from a specific block in the Swin transformer encoder, we use window-attention [59] and partition the feature $\boldsymbol{f}_i^E$ into window features $\boldsymbol{f}_{i\_w}^E \in \mathbb{R}^{l_i/M^2 \times M^2 \times c_i}$, where M represents the window size and l_i/M^2 is the number of windows. Then, we replicate the prompts $\boldsymbol{p}_i^d \in \mathbb{R}^{N_i \times c_i}$ for each window and obtain $\boldsymbol{p}_i^{d'} \in \mathbb{R}^{l_i/M^2 \times N_i \times c_i}$, where N_i represents the number of learnable prompt tokens. Next, we append them to the patch feature tokens in each window and perform self-attention within each window, which can be defined as

$$\begin{bmatrix} \boldsymbol{p}_{i+1}^{d'} \\ \boldsymbol{f}_{i\_w}^{E} \end{bmatrix} \leftarrow \mathrm{MLP}(\mathrm{SW/W\text{-}MSA}(\begin{bmatrix} \boldsymbol{p}_{i}^{d'} \\ \boldsymbol{f}_{i\_w}^{E} \end{bmatrix})), \quad (1)$$

where W-MSA and SW-MSA are multi-head self-attention modules with regular and shifted windowing configurations, respectively. Here we omit the residual connection [23], and layer normalization [2]. Next, we segment $\boldsymbol{p}_{i+1}^{d'}$ from each window and calculate the average of them to obtain $\boldsymbol{p}_{i+1}^{d}$, and then reassemble the output window feature $\boldsymbol{f}_{i\_w}^E$ to $\boldsymbol{f}_{i+1}^E$ for the next block.

3.3. Task-specific Prompt

Prior research [39] has traditionally regarded SOD and COD as opposing tasks, emphasizing the disparities between the features extracted by the SOD encoder and the COD encoder as much as possible. However, we believe that SOD and COD share significant commonalities in their features, such as low-level cues, high-level objectness, and spatial structuredness. As a result, we introduce task-specific prompts to learn the peculiarities while retaining the primary stream parameters shared to capture commonalities. We add the task-specific prompts in both VST encoder and decoder, and the overall impact of adding these prompts is illustrated in Figure 4.

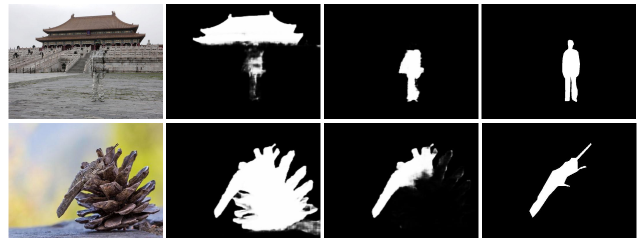

Figure 4. **Illustration of the influence of using different task prompts.**

Encoder. Although the encoder primarily focuses on domain-specific features with domain prompts, semantic features still play a pivotal role in distinguishing SOD and COD tasks. Semantic features from the encoder typically emphasize the most relevant region for a particular task and allocate more attention accordingly. In the case of the SOD task, the foreground region receives greater attention, whereas for the COD task, the background usually gains large importance since objects are typically concealed within it. Hence, it is essential to incorporate task-specific prompts to encourage learning task-related features in the encoder. Otherwise, we risk initially activating the wrong objects before the decoding process. Following the pattern of domain-specific prompts, we introduce task-specific prompts $\boldsymbol{p}_i^{te} \in \mathbb{R}^{N_i \times c_i}$ in each encoder block and use them in the same way as how domain-specific prompts are used.

Decoder. Camouflaged objects typically exhibit more intricate and detailed boundaries compared to salient objects. This complexity arises because concealed objects often share color or textual similarities with their surroundings, resulting in imperceptible boundaries. Therefore, solely introducing task-specific prompts in the encoder may not be adequate, as camouflaged objects require a more refined process within the decoder. We incorporate task-specific prompts in the decoder to allocate distinct attention for reconstructing both the boundary and object regions based on the features extracted by the encoder. In contrast, previous research [39] has not adequately explored the differences between these two tasks in the decoder, as they typically use a single decoder to handle both.

Regarding task-specific prompts in the decoder, we simply append learnable prompts $\boldsymbol{p}_{j+1}^{td} \in \mathbb{R}^{N \times d}$ to the decoder feature tokens $\boldsymbol{f}_{j+1}^D$ from a specific block $j+1$ in the decoder. Then, we apply the self-attention as follows:

$$\begin{bmatrix} \boldsymbol{p}_{j}^{td} \\ \boldsymbol{f}_{j}^{D} \end{bmatrix} \leftarrow \mathrm{MLP}(\mathrm{MSA}(\begin{bmatrix} \boldsymbol{p}_{j+1}^{td} \\ \boldsymbol{f}_{j+1}^{D} \end{bmatrix})), \quad (2)$$

where MSA denotes the multi-head self-attention. Here we omit the saliency and boundary tokens in the VST decoder

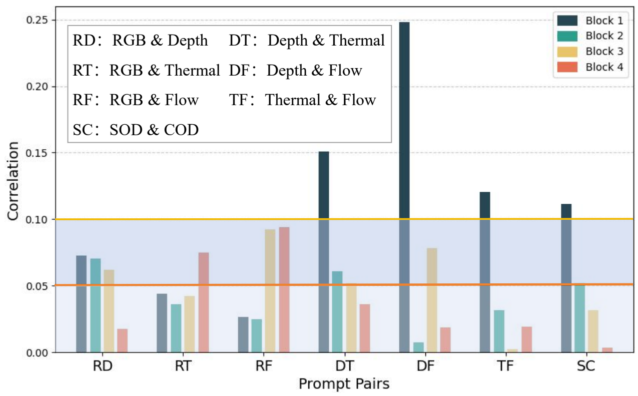

Figure 5. **Correlation of prompt pairs at each encoder block.**

for conciseness. Please note that our task-specific prompts differ from saliency and boundary tokens since we do not introduce any supervision for them.

3.4. Prompts Layout and Discussion

To incorporate the aforementioned prompts within the encoder-decoder architecture, inspired by VPT [31], we offer two prompt inserting versions. In the deep version, new prompts are introduced at the start of each transformer block, whereas the shallow version involves proposing prompts at first and updating them across all blocks. To unveil the specific relationship among different domains and tasks at varying network depths, we employ the deep version for both domain and task-specific prompts within the encoder. Based on VST's design, which introduces a saliency and a boundary token at the beginning of the decoder, we use the shallow version for task-specific prompts in the decoder.

We calculate the correlations of different domain and task prompt pairs at different blocks in Figure 5. It is evident that depth, thermal, and optical flow exhibit relatively strong correlations in low-level features, as all of them usually show obvious low-level contrast between target objects and backgrounds in terms of color or luminance. However, at higher levels, most domains exhibit lower correlations, highlighting the distinctions among them. Additionally, as for task-specific prompts, it is clear that SOD prompts and COD prompts exhibit more shared knowledge in the lower layers. As we progress to higher layers, the correlation decreases, indicating that high-level features gradually learn unrelated information. This observation urges us to implement the deep version of domain-specific prompts and task-specific prompts in the encoder in our final design, as different blocks acquire distinct knowledge. Moreover, the gradually decreased correlation values along with the increase of the network depth encourage us to use a progressively larger number of prompt tokens, as lower correlation means larger peculiarities and hence requires more parameters to learn.

3.5. Loss Function

The design principle of our model is to use 2D prompts for encompassing peculiarities while integrating commonalities into the foundation model. However, this is not straightforward for freely learned prompts. As shown in Figure 5, they still suggest certain correlations. This indicates that the learned prompts are entangled, risking the model's capacity to differentiate among various domains and tasks and resulting in suboptimal optimization. Hence, we propose a prompt discrimination loss to minimize the correlation among the prompts of the same type, guaranteeing that each prompt acquires unique domain or task knowledge. Specifically, we aggregate prompts of the same domain/task into a single embedding and then perform discrimination. First, we average the input prompt tokens of each same prompt type at each block and use linear projections to align the channel numbers to d. Subsequently, for each type of prompt, we concatenate the averaged prompts of different blocks, and use MLP to obtain the overall domain-specific prompt $p_l^{d_{all}}$ and task-specific encoder prompt $p_k^{te_{all}}$:

$$\begin{aligned} p_l^{d_{all}} &= \text{MLP}[\text{LA}(\boldsymbol{p}_0^d); \text{LA}(\boldsymbol{p}_1^d); \text{LA}(\boldsymbol{p}_2^d); \text{LA}(\boldsymbol{p}_3^d)], \\ p_k^{te_{all}} &= \text{MLP}[\text{LA}(\boldsymbol{p}_0^{te}); \text{LA}(\boldsymbol{p}_1^{te}); \text{LA}(\boldsymbol{p}_2^{te}); \text{LA}(\boldsymbol{p}_3^{te})], \end{aligned} \tag{3}$$

where L and A represent the linear and average operation, respectively, with $l \in \{depth, thermal, flow, rgb\}$, and $k \in \{SOD, COD\}$. Since the task-specific prompts in the decoder are shallow, we simply average them.

Afterward, we calculate the cosine similarity between prompt pairs, resulting in eight types of cosine similarity results $\mathcal{CS}_m$. Here m means the combination of domains/tasks, namely $\{RD, RT, RF, DF, DT, TF\}$ for domain-aggregated prompts and $\{SC_{EN}, SC_{DE}\}$ for task-aggregated prompts in the encoder and decoder, respectively. Finally, we minimize the correlation within these prompt pairs and define our prompt discrimination loss as

$$\mathcal{L}_{dis} = \sum_m \ln(1 + |\mathcal{CS}_m|), \tag{4}$$

which is further combined with the segmentation losses and boundary losses [54] to train our model.

4. Experiment

4.1. Datasets and Evaluation Metrics

For RGB SOD, we evaluate our proposed model using six commonly used benchmark datasets, i.e. **DUTS** [81], **ECSSD** [97], **HKU-IS** [45], **PASCAL-S** [47], **DUT-O** [99], and **SOD** [62]. For RGB-D SOD, we use six large benchmark datasets, including **STERE** [64], **NJUD** [33], **NLPR** [69], **DUTLF-Depth** [71], **SIP** [17], and **ReDWeb-S** [53]. In terms of RGB-T SOD, we consider three public datasets: **VT821** [79], **VT1000** [78], and **VT5000** [77]. For VSOD, we employ six widely used benchmark datasets: **DAVIS** [70], **FBMS** [65], **ViSal** [85], **SegV2** [41], **DAVSOD-Easy**, and **DAVSOD-Normal** [18]. Regarding RGB COD, three extensive benchmark datasets are considered, including

Settings	Params (M)	RGB SOD DUTS[81] $S_m \uparrow$	$F_m \uparrow$	$E_m \uparrow$	RGB-D SOD NJUD[33] $S_m \uparrow$	$F_m \uparrow$	$E_m \uparrow$	RGB-T SOD VT5000[77] $S_m \uparrow$	$F_m \uparrow$	$E_m \uparrow$	VSOD SegV2[41] $S_m \uparrow$	$F_m \uparrow$	$E_m \uparrow$	RGB COD CAMO[37] $S_m \uparrow$	$F_m \uparrow$	$E_m \uparrow$	VCOD CAD[3] $S_m \uparrow$	$F_m \uparrow$	$E_m \uparrow$
ST	323.46*	.900	.885	.940	.927	.928	.958	.900	.863	.938	.896	.870	.952	.793	.751	.871	.686	.522	.787
GT	54.06	.898	.884	.941	.924	.922	.954	.903	.886	.942	.930	.911	.972	-	-	-	-	-	-
GT+p^d	54.06	.902	.890	.945	.931	.932	.962	.909	.877	.947	.931	.917	.975	-	-	-	-	-	-
GT+p^d+p^t	54.09	.904	.892	.945	.931	.931	.961	.906	**.892**	.946	.925	.910	.970	.804	.776	.876	**.759**	**.639**	.808
GT+p^d+p^t+$\mathcal{L}_{dis}$	54.09	**.909**	**.899**	**.948**	**.935**	**.938**	**.965**	**.912**	.882	**.950**	**.943**	**.930**	**.984**	**.811**	**.782**	**.884**	.736	.614	.797
w/o p^{te}	54.07	.908	.896	.947	.932	.932	.960	.909	.878	.947	.933	.907	.966	.800	.770	.872	.743	.611	.798
w/o p^{td}	54.08	.902	.889	.943	.929	.929	.959	.904	.872	.941	.940	.919	.975	.799	.770	.875	.740	.599	**.814**

Table 1. **Ablation studies of our VSCode on the Swin-T [59] backbone with** 224×224 **image size.** We conduct evaluations on one representative dataset for each task. "ST" indicates special training, "GT" means general training, p^d represents domain-specific prompts, and p^t is task-specific prompts, which consists of p^{te} in the encoder and p^{td} in the decoder. $\mathcal{L}_{dis}$ is our prompt discrimination loss. The best results under each setting are labeled in **bold**.

Settings	Params (M)	RGB SOD DUTS[81] $S_m \uparrow$	$F_m \uparrow$	$E_m \uparrow$	RGB-D SOD NJUD[33] $S_m \uparrow$	$F_m \uparrow$	$E_m \uparrow$	RGB-T SOD VT5000[77] $S_m \uparrow$	$F_m \uparrow$	$E_m \uparrow$	VSOD SegV2[41] $S_m \uparrow$	$F_m \uparrow$	$E_m \uparrow$	RGB COD CAMO[37] $S_m \uparrow$	$F_m \uparrow$	$E_m \uparrow$	VCOD CAD[3] $S_m \uparrow$	$F_m \uparrow$	$E_m \uparrow$
shallow or deep version for domain-specific prompts																			
shallow	54.83	.900	.887	.943	.926	.926	.955	.905	.873	.944	**.931**	**.917**	.972	-	-	-	-	-	-
deep	54.06	**.902**	**.890**	**.945**	**.931**	**.932**	**.962**	**.909**	**.877**	**.947**	**.931**	**.917**	**.975**	-	-	-	-	-	-
shallow or deep version for task-specific prompts in the encoder																			
shallow	54.45	.902	.888	.943	.928	.928	.958	.902	.870	.940	**.927**	**.905**	**.964**	.793	.763	.866	.747	.616	.798
deep	54.08	**.903**	**.890**	**.944**	**.934**	**.934**	**.962**	**.905**	**.874**	**.943**	.924	.903	.960	**.804**	**.772**	**.881**	**.759**	**.651**	**.831**
shallow or deep version for task-specific prompts in the decoder																			
deep	54.10	.903	.891	**.945**	**.930**	**.932**	.960	.905	.888	.943	.922	.903	.966	.802	.774	**.881**	.738	.605	.801
shallow	54.09	**.904**	**.892**	**.945**	**.930**	.931	**.961**	**.906**	**.892**	**.946**	**.925**	**.910**	**.970**	**.804**	**.776**	.876	**.759**	**.639**	**.808**
number of domain-specific prompts at four blocks																			
1,1,1,1	54.06	.902	**.890**	**.945**	**.931**	.932	**.962**	**.909**	**.877**	**.947**	**.931**	**.917**	**.975**	-	-	-	-	-	-
5,5,5,5	54.08	**.903**	.890	**.945**	.931	**.935**	.961	.902	.869	.940	.918	.887	.954	-	-	-	-	-	-
number of task-specific prompts in the encoder at four blocks																			
5,5,5,5	54.07	**.903**	**.893**	**.947**	.928	.930	.959	.903	.870	.940	**.931**	**.918**	**.975**	.795	.766	.866	.739	.600	.799
1,1,5,10	54.08	**.903**	.890	.944	**.934**	**.934**	**.962**	**.905**	**.874**	**.943**	.924	.903	.960	**.804**	**.772**	**.881**	**.759**	**.651**	**.831**
number of task-specific prompts in the decoder																			
5	54.08	**.904**	.890	**.946**	.929	.931	.957	.904	.890	.943	.931	.911	.969	**.807**	**.782**	**.881**	.746	.626	.805
10	54.09	**.904**	**.892**	.945	**.930**	.931	**.961**	**.906**	**.892**	**.946**	.925	.910	.970	.804	.776	.876	**.759**	**.639**	**.808**
15	54.09	.903	.889	.944	.929	**.933**	.956	.904	.890	.942	**.932**	**.913**	**.974**	.798	.771	.875	.743	.621	.791

Table 2. **Ablation studies of different designs of prompt layout.**

COD10K [15], **CAMO** [37], and **NC4K** [61]. For VCOD, we utilize two widely accepted benchmark datasets: **CAD** [3] and **MoCA-Mask** [10]. To ensure a consistent evaluation across all SOD and COD tasks, we employ three commonly used evaluation metrics to assess model performance: structure-measure S_m [13], maximum enhanced-alignment measure E_m [14], and maximum F-measure F_m.

4.2. Implementation Details

Building on prior research [10, 22, 51, 54, 77], we employ the following datasets to train our model concurrently: the training set of **DUTS** for RGB SOD, the training sets of **NJUD**, **NLPR**, and **DUTLF-Depth** for RGB-D SOD, the training set of **VT5000** for RGB-T SOD, the training sets of **DAVIS** and **DAVSOD** for VSOD, the training sets of **COD10K** and **CAMO** for RGB COD, and the training set of **MoCA-Mask** for VCOD. To ensure a fair comparison with previous works [22, 38, 56, 76, 116], we resize each image to 384×384 pixels and then randomly crop them to 352×352 image regions for training. Our training process employs the Adam optimizer [35] with an initial learning rate of 0.0001, which is reduced by a factor of 10 at half and three-quarters of the total training steps. We conduct a total of 150,000 training steps using a 3090 GPU. We mix the above six tasks in each training iteration with two samples for each task, leading to a total batch size of 12.

4.3. Ablation Study

Architecture Design. To demonstrate the efficacy of various components in our VSCode model, we report the quantitative results in Table 1. We start by performing special training (ST) on each task individually and then conduct general training (GT) on all SOD tasks. Please note that here we do not consider COD tasks since no task prompt is used. We observe improved performance on RGB-T SOD and VSOD, demonstrating the significant benefit of shared knowledge in different tasks, especially for those with limited training data diversity. However, the results of RGB SOD and RGB-D SOD do not show a significant increase. Our hypothesis is that amalgamating the training of multi-

*The parameters for our specialized training methods amount to 53.61M for the RGB task and 54.06M for the multimodal task, resulting in a total of 323.46M parameters for all six tasks.

Method	Params (M)	DUTS[81]			ECSSD[97]			HKU-IS[45]			PASCAL-S[47]			DUT-O[99]			SOD[62]		
		$S_m \uparrow$	$F_m \uparrow$	$E_m \uparrow$	$S_m \uparrow$	$F_m \uparrow$	$E_m \uparrow$	$S_m \uparrow$	$F_m \uparrow$	$E_m \uparrow$	$S_m \uparrow$	$F_m \uparrow$	$E_m \uparrow$	$S_m \uparrow$	$F_m \uparrow$	$E_m \uparrow$	$S_m \uparrow$	$F_m \uparrow$	$E_m \uparrow$
VST[54]	44.48	.896	.877	.939	.932	.944	.964	.928	.937	.968	.873	.850	.900	.850	.800	.888	.854	.866	.902
ICON-R[116]	33.09	.890	.876	.931	.928	.943	.960	.920	.931	.960	.862	.844	.888	.845	.799	.884	.848	.861	.899
VST-T++ [50]	53.60	.901	.887	.943	.937	.949	.968	.930	.939	.968	.878	**.855**	**.901**	.853	.804	.892	.853	.866	.899
MENet[89]	27.83	.905	.895	.943	.927	.938	.956	.927	.939	.965	.871	.848	.892	.850	.792	.879	.841	.847	.884
VSCode-T	54.09	**.917**	**.910**	**.954**	**.945**	**.957**	**.971**	**.935**	**.946**	**.970**	**.878**	.852	.900	**.869**	**.830**	**.910**	**.863**	**.879**	**.908**
EVP[56]	64.52†	.917	.910	.956	.936	.949	.965	.935	.945	.971	.880	.859	.902	.864	.822	.902	.854	.873	.901
VSCode-S	74.72†	**.926**	**.922**	**.960**	**.949**	**.959**	**.974**	**.940**	**.951**	**.974**	**.887**	**.864**	**.904**	**.877**	**.840**	**.912**	**.870**	**.882**	**.910**

Table 3. **Quantitative comparison of our VSCode with other 5 SOTA RGB SOD methods on six benchmark datasets.** "-R", "-T" and "-S" mean the ResNet50 [23], Swin-T, and Swin-S[59] backbones, respectively. '-' indicates the code is not available. The best performance under all settings is **bolded**, and the best results under each setting are labeled in **bold**.

Method	Params (M)	NJUD [33]			NLPR[69]			DUTLF-Depth[71]			ReDWeb-S[53]			STERE[64]			SIP[17]		
		$S_m \uparrow$	$F_m \uparrow$	$E_m \uparrow$	$S_m \uparrow$	$F_m \uparrow$	$E_m \uparrow$	$S_m \uparrow$	$F_m \uparrow$	$E_m \uparrow$	$S_m \uparrow$	$F_m \uparrow$	$E_m \uparrow$	$S_m \uparrow$	$F_m \uparrow$	$E_m \uparrow$	$S_m \uparrow$	$F_m \uparrow$	$E_m \uparrow$
CMINet[103]	188.12	.929	.934	.957	.932	.922	.963	.912	.913	.938	.725	.726	.800	.918	.916	.951	.899	.910	.939
VST[54]	53.83	.922	.920	.951	.932	.920	.962	.943	.948	.969	.759	.763	.826	.913	.907	.951	.904	.915	.944
VST-T++ [50]	100.51	.928	.929	.958	.933	.921	.964	.944	.948	.969	.756	.757	.819	.916	.911	.950	.903	.914	.944
SPSN[38]	-	-	-	-	.923	.912	.960	-	-	-	-	-	-	.907	.902	.945	.892	.900	.936
CAVER[68]	55.79	.920	.924	.953	.929	.921	.964	.931	.939	.962	.730	.724	.802	.914	.911	.951	.893	.906	.934
VSCode-T	54.09	**.941**	**.945**	**.967**	**.938**	**.930**	**.966**	**.952**	**.959**	**.974**	**.766**	**.771**	**.831**	**.928**	**.926**	**.957**	**.917**	**.936**	**.955**
VSCode-S	74.72	**.944**	**.949**	**.970**	**.941**	**.932**	**.968**	**.960**	**.967**	**.980**	**.777**	**.776**	**.829**	**.931**	**.928**	**.958**	**.924**	**.942**	**.958**

Table 4. **Quantitative comparison of our VSCode with other 5 SOTA RGB-D SOD methods on six benchmark datasets.**

Method	Params (M)	VT821[79]			VT1000[78]			VT5000[77]		
		S_m	F_m	E_m	S_m	F_m	E_m	S_m	F_m	E_m
MIDD[76]	52.43	.871	.847	.916	.916	.904	.956	.868	.834	.919
TNet[11]	87.41	.899	.885	.936	.929	.921	.965	.895	.864	.936
CGMDR[7]	-	.894	.872	.932	.931	.927	.966	.896	.877	.939
VST-T++ [50]	100.51	.894	.861	.923	.941	.931	.972	.895	.854	.933
CAVER[68]	55.79	.891	.874	.933	.936	.927	.970	.892	.857	.935
VSCode-T	54.09	**.921**	**.906**	**.951**	**.949**	**.944**	**.981**	**.918**	**.892**	**.954**
VSCode-S	74.72	**.926**	**.910**	**.954**	**.952**	**.947**	**.981**	**.925**	**.900**	**.959**

Table 5. **Quantitative comparison of our VSCode with other 5 SOTA RGB-T SOD methods on three benchmark datasets.**

modal images within a shared model might prevent further optimization on those well-learned tasks. Based on this, we introduce domain-specific prompts p^d, resulting in substantial improvements across all datasets, which demonstrates the efficacy of domain-specific prompts in consolidating peculiarities within their respective domains. Subsequently, we introduce task-specific prompts p^t in the encoder-decoder architecture, enabling the capability to handle COD tasks. This brings slightly improved performance on some SOD tasks, however, significantly improves the performance on all COD tasks compared with the ST baseline, which probably owes to the well-learned commonalities from different tasks. Moreover, the incorporation of the prompt discrimination loss $\mathcal{L}_{dis}$ leads to improved performance on most tasks, reaffirming its effectiveness in disentangling peculiarities.

To further evaluate the effectiveness of the task-specific prompts in the encoder and decoder, we remove them individually, resulting in performance decrease. This indicates that using task prompts in both encoder and decoder is necessary. We also observe our 2D prompts only bring around 0.03M parameters, which makes our model much more parameter-efficient than the traditional special training scheme.

Prompt Location. Following VPT [31], we design other forms of prompt layout based on Section 3.4. Table 2 reveals that employing the shallow version of task-specific prompts in the decoder, the deep version of domain-specific prompts and task-specific prompts in the encoder yields the best results. One plausible rationale is that each block aggregates distinct-level features within the encoder, thus it is better to propose unique prompts for each block. In our decoder, we follow VST and used skip connection to fuse decoder features with encoder features, which have already utilized deep task prompts for distinction. Hence, using more task prompts in the decoder may not be essential, and the shallow version seems to be a more fitting choice.

Prompt Length. We perform experiments with varying lengths for three kinds of prompts. As shown in Table 2, for domain-specific prompts, using one prompt token at each block achieves better performance than using more tokens. This suggests that it's possible to effectively capture domain distinctions using only a small number of prompts, which matches the observed relatively large correlation within domain prompts in Figure 5. Regarding task-specific prompts within the encoder, a prompt layout of 1,1,5,10 tokens at four blocks is found to be optimal on COD tasks, highlighting the importance of high-level semantic features over low-level features in distinguishing between SOD and COD tasks. This observation matches Figure 5 as well in which the correlations of SC in deep blocks are smaller than those in shallow blocks. Regarding the number of task-specific prompts in the decoder, performance starts to decline when it exceeds

†Please note that our model shares parameters across six tasks, in contrast to EVP, which uses task-specific training. Therefore, comparing the parameters of our model with EVP may not be completely fair owing to the differences in training strategies and backbone utilization.

Method	Params (M)	DAVIS [70]			FBMS[65]			ViSal[85]			SegV2[41]			DAVSOD-Easy[18]			DAVSOD-Normal[18]		
		$S_m\uparrow$	$F_m\uparrow$	$E_m\uparrow$	$S_m\uparrow$	$F_m\uparrow$	$E_m\uparrow$	$S_m\uparrow$	$F_m\uparrow$	$E_m\uparrow$	$S_m\uparrow$	$F_m\uparrow$	$E_m\uparrow$	$S_m\uparrow$	$F_m\uparrow$	$E_m\uparrow$	$S_m\uparrow$	$F_m\uparrow$	$E_m\uparrow$
DCFNet[105]	71.66	.914	.899	.970	.883	.853	.910	**.952**	.953	**.990**	.903	.870	.953	.729	.612	.781	.708	.601	.791
FSNet[28]	102.30	.922	.909	.972	.875	.867	.918	-	-	-	.849	.773	.920	.760	.637	.796	.732	.623	.789
CoSTFormer[51]	-	.923	.906	**.978**	-	-	-	-	-	-	.874	.813	.943	.779	.667	.819	.730	.614	.777
UFO[75]	55.92	.918	.906	**.978**	.858	.868	.911	.926	.917	.969	.888	.850	.951	.747	.626	.799	.711	.605	.773
VSCode-T	54.09	**.930**	**.913**	.970	**.891**	**.880**	**.923**	**.952**	**.954**	.989	**.943**	**.937**	**.984**	**.792**	**.696**	**.831**	**.738**	**.631**	**.797**
VSCode-S	74.72	**.936**	**.922**	**.973**	**.905**	**.902**	**.939**	**.955**	**.957**	**.991**	**.946**	**.937**	**.984**	**.800**	**.710**	**.835**	**.758**	**.666**	**.815**

Table 6. **Quantitative comparison of our VSCode with other 4 SOTA VSOD methods on six benchmark datasets.**

Method	Params (M)	COD10K[15]			NC4K[61]			CAMO[37]		
		S_m	F_m	E_m	S_m	F_m	E_m	S_m	F_m	E_m
MGL[101]	63.60	.814	.738	.890	-	-	-	.776	.741	.842
UJSC[39]	121.63	.817	.750	.902	.856	.835	.920	.803	.775	.867
SegMar[32]	56.21	.833	.755	.907	.841	.827	.907	.816	.803	.884
FEDER[22]	44.13	.822	.768	.905	.847	.833	.915	.802	.789	.873
VSCode-T	54.09	**.847**	**.795**	**.925**	**.874**	**.853**	**.930**	**.838**	**.821**	**.909**
EVP[56]	64.52	.845	.794	.926	.874	.855	.933	.849	.833	.918
VSCode-S	74.72	**.869**	**.827**	**.942**	**.891**	**.878**	**.944**	**.873**	**.861**	**.938**

Table 7. **Quantitative comparison of our VSCode with other 5 SOTA RGB COD methods on three benchmark datasets.**

Method	Params (M)	CAD [3]			MoCA-Mask[10]		
		$S_m\uparrow$	$F_m\uparrow$	$E_m\uparrow$	$S_m\uparrow$	$F_m\uparrow$	$E_m\uparrow$
PNS-Net[26]	26.87	.671	.473	.787	.514	.068	.599
RCRNet[96]	53.79	.664	.405	.786	.559	.170	.593
MG[98]	-	.608	.378	.673	.500	.138	.514
SLT-Net[10]	164.68	.715	.542	**.823**	.624	.327	.768
VSCode-T	54.09	**.757**	**.659**	.808	**.650**	**.339**	**.787**
VSCode-S	74.72	**.790**	**.680**	**.853**	**.665**	**.386**	**.796**

Table 8. **Quantitative comparison of our VSCode with other 4 SOTA VCOD methods on two benchmark datasets.**

10. This emphasizes that blindly increasing the number of prompts doesn't guarantee improved performance.

4.4. Comparison with State-of-the-Art Methods

Due to space limitation, we only report the performance comparison of our methods against other most highly-performed state-of-the-art methods, including 4 specialist RGB SOD models [50, 54, 89, 116], 5 specialist RGB-D SOD models [38, 50, 54, 68, 103], 5 specialist RGB-T SOD models [7, 11, 50, 68, 76], 3 specialist VSOD models [28, 51, 105], 4 specialist RGB COD models [22, 32, 39, 101], and 4 specialist VCOD models [10, 26, 96, 98]. Two generalist models [56, 75] are also reported. To ensure a relatively fair comparison with EVP [56], which utilizes SegFormer-B4 [93] as their backbone (64.1M parameters), we switch our backbone to Swin-S [59] as it has a similar number of parameters (50M). As shown in Table 3, Table 4, Table 5, Table 6, Table 7, and Table 8, our VSCode significantly outperforms all specialist methods and two generalist models across all six tasks, underscoring the effectiveness of our specially designed 2D prompts and prompt discrimination loss. The supplementary material displays visual comparison results among the top-performing models.

4.5. Analysis of Generalization Ability

Previous generalist research [56, 75] primarily concentrated on assessing the effectiveness of models in training tasks, neglecting their capacity for generalizing to novel tasks. Therefore, we employ the RGB-D COD task, which is not used in training, to further investigate the zero-shot generalization capabilities of our model. Specifically, we utilize our well-trained model and combine depth and COD prompts to tackle the RGB-D COD task. As shown in Table 9, our VSCode model significantly outperforms the state-of-the-art specialist model PopNet [92], although ours works in a pure zero-shot way. This demonstrates the superior zero-shot generalization ability of our proposed method. We also present the results of our model using only RGB information, which yields considerably lower performance compared to zero-shot RGB-D results. This validates that our zero-shot performance is not reliant on the utilization of seen RGB COD information but on the effectiveness of consolidating domain- and task-specific knowledge, which allows for the straightforward combination of various domain- and task-specific prompts for unseen tasks.

Summary		COD10K[15]			NC4K[61]			CAMO[37]		
Method	Task	S_m	F_m	E_m	S_m	F_m	E_m	S_m	F_m	E_m
PopNet[92]	RGB-D	.851	.802	.919	.861	.843	.919	.808	.792	.874
VSCode-T	**ZS RGB-D**	**.882**	**.849**	**.945**	**.902**	**.894**	**.950**	**.885**	**.885**	**.945**
VSCode-T	RGB	.847	.795	.925	.874	.853	.930	.838	.821	.909

Table 9. **Comparison with the SOTA RGB-D COD method on three benchmark datasets.** "ZS" indicates zero-shot.

5. Conclusion

In this paper, we present VSCode, a novel generalist and parameter-efficient model that tackles general multimodal SOD and COD tasks. Concretely, we use a foundation model to assimilate commonalities and 2D prompts to learn domain and task peculiarities. Furthermore, a prompt discrimination loss is introduced to help effectively disentangle specific knowledge and learn better shared knowledge. Our experiments demonstrate the effectiveness of VSCode on six training tasks and one unseen task.

Acknowledgments

This work was supported in part by the Key R&D Program of Shaanxi Province under Grant 2021ZDLGY01-08; the National Natural Science Foundation of China under Grants 62136007, U20B2065, 62036005, 62322605; the Key Research and Development Program of Jiangsu Province under Grant BE2021093; the Institute of Artificial Intelligence, Hefei Comprehensive National Science Center Project under Grant 21KT008; and by the MBZUAI-WIS Joint Program for AI Research under Grants WIS P008 and P009.

References

[1] Muhammad Awais, Muzammal Naseer, Salman Khan, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Foundational models defining a new era in vision: A survey and outlook. *arXiv preprint arXiv:2307.13721*, 2023. 3

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4

[3] Pia Bideau and Erik Learned-Miller. It's moving! a probabilistic model for causal motion segmentation in moving camera videos. In *ECCV*, pages 433–449. Springer, 2016. 2, 6, 8

[4] Pia Bideau, Rakesh R Menon, and Erik Learned-Miller. Moa-net: self-supervised motion segmentation. In *ECCV*, pages 0–0, 2018. 2

[5] Ali Borji, Ming-Ming Cheng, Qibin Hou, Huaizu Jiang, and Jia Li. Salient object detection: A survey. *CVMJ*, 5:117–150, 2019. 1

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 33:1877–1901, 2020. 3

[7] Gang Chen, Feng Shao, Xiongli Chai, Hangwei Chen, Qiuping Jiang, Xiangchao Meng, and Yo-Sung Ho. Cgmdrnet: Cross-guided modality difference reduction network for rgb-t salient object detection. *IEEE TCSVT*, 32(9):6308–6323, 2022. 7, 8

[8] Shuhan Chen and Yun Fu. Progressively guided alternate refinement network for rgb-d salient object detection. In *ECCV*, pages 520–538, 2020. 2

[9] Zuyao Chen, Runmin Cong, Qianqian Xu, and Qingming Huang. Dpanet: Depth potentiality-aware gated attention network for rgb-d salient object detection. *IEEE TIP*, 2020. 2

[10] Xuelian Cheng, Huan Xiong, Deng-Ping Fan, Yiran Zhong, Mehrtash Harandi, Tom Drummond, and Zongyuan Ge. Implicit motion handling for video camouflaged object detection. In *CVPR*, pages 13864–13873, 2022. 1, 2, 6, 8

[11] Runmin Cong, Kepu Zhang, Chen Zhang, Feng Zheng, Yao Zhao, Qingming Huang, and Sam Kwong. Does thermal really always matter for rgb-t salient object detection? *IEEE TMM*, 2022. 7, 8

[12] Zijun Deng, Xiaowei Hu, Lei Zhu, Xuemiao Xu, Jing Qin, Guoqiang Han, and Pheng-Ann Heng. R3net: Recurrent residual refinement network for saliency detection. In *IJCAI*, pages 684–690, 2018. 2

[13] Deng-Ping Fan, Ming-Ming Cheng, Yun Liu, Tao Li, and Ali Borji. Structure-measure: A new way to evaluate foreground maps. In *ICCV*, pages 4548–4557, 2017. 6

[14] Deng-Ping Fan, Cheng Gong, Yang Cao, Bo Ren, Ming-Ming Cheng, and Ali Borji. Enhanced-alignment Measure for Binary Foreground Map Evaluation. In *IJCAI*, pages 698–704, 2018. 6

[15] Deng-Ping Fan, Ge-Peng Ji, Guolei Sun, Ming-Ming Cheng, Jianbing Shen, and Ling Shao. Camouflaged object detection. In *CVPR*, pages 2777–2787, 2020. 1, 2, 6, 8

[16] Deng-Ping Fan, Ge-Peng Ji, Peng Xu, Ming-Ming Cheng, Christos Sakaridis, and Luc Van Gool. Advances in deep concealed scene understanding. *VI*, 1(1):16, 2023. 2

[17] Deng-Ping Fan, Zheng Lin, Zhao Zhang, Menglong Zhu, and Ming-Ming Cheng. Rethinking rgb-d salient object detection: Models, data sets, and large-scale benchmarks. *IEEE TNNLS*, 32(5):2075–2089, 2020. 5, 7

[18] Deng-Ping Fan, Wenguan Wang, Ming-Ming Cheng, and Jianbing Shen. Shifting more attention to video salient object detection. In *CVPR*, pages 8554–8564, 2019. 2, 5, 8

[19] Deng-Ping Fan, Yingjie Zhai, Ali Borji, Jufeng Yang, and Ling Shao. Bbs-net: Rgb-d salient object detection with a bifurcated backbone strategy network. In *ECCV*, pages 275–292, 2020. 2

[20] Chaowei Fang, Haibin Tian, Dingwen Zhang, Qiang Zhang, Jungong Han, and Junwei Han. Densely nested top-down flows for salient object detection. *SCIS*, 65(8):182103, 2022. 2

[21] Yuchao Gu, Lijuan Wang, Ziqin Wang, Yun Liu, Ming-Ming Cheng, and Shao-Ping Lu. Pyramid constrained self-attention network for fast video salient object detection. In *AAAI*, volume 34, pages 10869–10876, 2020. 2

[22] Chunming He, Kai Li, Yachao Zhang, Longxiang Tang, Yulun Zhang, Zhenhua Guo, and Xiu Li. Camouflaged object detection with feature decomposition and edge reconstruction. In *CVPR*, pages 22046–22055, 2023. 6, 8

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 4, 7

[24] Q Hou, MM Cheng, X Hu, A Borji, Z Tu, and PHS Torr. Deeply supervised salient object detection with short connections. *IEEE TPAMI*, 41(4):815–828, 2018. 2

[25] Guyue Hu, Bin He, and Hanwang Zhang. Compositional prompting video-language models to understand procedure in instructional videos. *MIR*, 20(2):249–262, 2023. 3

[26] Ge-Peng Ji, Yu-Cheng Chou, Deng-Ping Fan, Geng Chen, Huazhu Fu, Debesh Jha, and Ling Shao. Progressively normalized self-attention network for video polyp segmentation. In *MICCAI*, pages 142–152. Springer, 2021. 8

[27] Ge-Peng Ji, Deng-Ping Fan, Yu-Cheng Chou, Dengxin Dai, Alexander Liniger, and Luc Van Gool. Deep gradient learning for efficient camouflaged object detection. *MIR*, 20(1):92–108, 2023. 1

[28] Ge-Peng Ji, Deng-Ping Fan, Keren Fu, Zhe Wu, Jianbing Shen, and Ling Shao. Full-duplex strategy for video object segmentation. In *CVMJ*, pages 155–175, 2023. 2, 8

[29] Wei Ji, Jingjing Li, Miao Zhang, Yongri Piao, and Huchuan Lu. Accurate rgb-d salient object detection via collaborative learning. In *ECCV*, pages 52–69, 2020. 2

[30] Yuzhu Ji, Haijun Zhang, Zequn Jie, Lin Ma, and QM Jonathan Wu. Casnet: A cross-attention siamese network for video salient object detection. *IEEE TNNLS*, 32(6):2676–2690, 2020. 2

[31] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *ECCV*, pages 709–727. Springer, 2022. 2, 3, 5, 7

[32] Qi Jia, Shuilian Yao, Yu Liu, Xin Fan, Risheng Liu, and Zhongxuan Luo. Segment, magnify and reiterate: Detecting camouflaged objects the hard way. In *CVPR*, pages 4713–

4722, 2022. 2, 8
[33] Ran Ju, Ling Ge, Wenjing Geng, Tongwei Ren, and Gangshan Wu. Depth saliency based on anisotropic center-surround difference. In *ICIP*, pages 1115–1119, 2014. 5, 6, 7
[34] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, pages 7482–7491, 2018. 2
[35] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6
[36] Hala Lamdouar, Charig Yang, Weidi Xie, and Andrew Zisserman. Betrayed by motion: Camouflaged object discovery via motion segmentation. In *ACCV*, 2020. 1, 2
[37] Trung-Nghia Le, Tam V Nguyen, Zhongliang Nie, Minh-Triet Tran, and Akihiro Sugimoto. Anabranch network for camouflaged object segmentation. *CVIU*, 184:45–56, 2019. 6, 8
[38] Minhyeok Lee, Chaewon Park, Suhwan Cho, and Sangyoun Lee. Spsn: Superpixel prototype sampling network for rgb-d salient object detection. In *ECCV*, pages 630–647. Springer, 2022. 6, 7, 8
[39] Aixuan Li, Jing Zhang, Yunqiu Lv, Bowen Liu, Tong Zhang, and Yuchao Dai. Uncertainty-aware joint salient object and camouflaged object detection. In *CVPR*, pages 10071–10081, 2021. 4, 8
[40] Aixuan Li, Jing Zhang, Yunqiu Lv, Tong Zhang, Yiran Zhong, Mingyi He, and Yuchao Dai. Joint salient object detection and camouflaged object detection via uncertainty-aware learning. *arXiv preprint arXiv:2307.04651*, 2023. 2
[41] Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, pages 2192–2199, 2013. 5, 6, 8
[42] Gongyang Li, Zhi Liu, and Haibin Ling. Icnet: Information conversion network for rgb-d based salient object detection. *IEEE TIP*, 29:4873–4884, 2020. 1, 2
[43] Gongyang Li, Zhi Liu, Linwei Ye, Yang Wang, and Haibin Ling. Cross-modal weighting network for rgb-d salient object detection. In *ECCV*, pages 665–681, 2020. 2
[44] Guanbin Li, Yuan Xie, Tianhao Wei, Keze Wang, and Liang Lin. Flow guided recurrent neural encoder for video salient object detection. In *CVPR*, pages 3243–3252, 2018. 1, 2
[45] Guanbin Li and Yizhou Yu. Visual saliency based on multi-scale deep features. In *CVPR*, pages 5455–5463, 2015. 5, 7
[46] Hao Li, Dingwen Zhang, Nian Liu, Lechao Cheng, Yalun Dai, Chao Zhang, Xinggang Wang, and Junwei Han. Boosting low-data instance segmentation by unsupervised pre-training with saliency prompt. In *CVPR*, pages 15485–15494, 2023. 1
[47] Yin Li, Xiaodi Hou, Christof Koch, James M Rehg, and Alan L Yuille. The secrets of salient object segmentation. In *CVPR*, pages 280–287, 2014. 5, 7
[48] Nian Liu and Junwei Han. Dhsnet: Deep hierarchical saliency network for salient object detection. In *CVPR*, pages 678–686, 2016. 2
[49] Nian Liu, Junwei Han, and Ming-Hsuan Yang. Picanet: Learning pixel-wise contextual attention for saliency detection. In *CVPR*, pages 3089–3098, 2018. 2
[50] Nian Liu, Ziyang Luo, Ni Zhang, and Junwei Han. Vst++: Efficient and stronger visual saliency transformer. *arXiv preprint arXiv:2310.11725*, 2023. 4, 7, 8
[51] Nian Liu, Kepan Nan, Wangbo Zhao, Xiwen Yao, and Junwei Han. Learning complementary spatial–temporal transformer for video salient object detection. *IEEE TNNLS*, 2023. 2, 6, 8
[52] Nian Liu, Ni Zhang, and Junwei Han. Learning selective self-mutual attention for rgb-d saliency detection. In *CVPR*, pages 13756–13765, 2020. 2
[53] Nian Liu, Ni Zhang, Ling Shao, and Junwei Han. Learning selective mutual attention and contrast for rgb-d saliency detection. *IEEE TPAMI*, 44(12):9026–9042, 2021. 2, 5, 7
[54] Nian Liu, Ni Zhang, Kaiyuan Wan, Ling Shao, and Junwei Han. Visual saliency transformer. In *ICCV*, pages 4722–4732, October 2021. 1, 2, 3, 5, 6, 7, 8
[55] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *CVPR*, pages 1871–1880, 2019. 2
[56] Weihuang Liu, Xi Shen, Chi-Man Pun, and Xiaodong Cun. Explicit visual prompting for universal foreground segmentations. *arXiv preprint arXiv:2305.18476*, 2023. 3, 6, 7, 8
[57] Yi Liu, Dingwen Zhang, Nian Liu, Shoukun Xu, and Jungong Han. Disentangled capsule routing for fast part-object relational saliency. *IEEE TIP*, 31:6719–6732, 2022. 1
[58] Yi Liu, Dingwen Zhang, Qiang Zhang, and Jungong Han. Part-object relational visual saliency. *TPAMI*, 44(7):3688–3704, 2021. 1
[59] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 3, 4, 6, 7, 8
[60] Zhengyi Liu, Song Shi, Quntao Duan, Wei Zhang, and Peng Zhao. Salient object detection for rgb-d image by single stream recurrent convolution neural network. *IJON*, 363:46–57, 2019. 2
[61] Yunqiu Lv, Jing Zhang, Yuchao Dai, Aixuan Li, Bowen Liu, Nick Barnes, and Deng-Ping Fan. Simultaneously localize, segment and rank the camouflaged objects. In *CVPR*, pages 11591–11601, 2021. 1, 2, 6, 8
[62] Vida Movahedi and James H Elder. Design and perceptual validation of performance measures for salient object segmentation. In *CVPR*, pages 49–56, 2010. 5, 7
[63] Xing Nie, Bolin Ni, Jianlong Chang, Gaomeng Meng, Chunlei Huo, Zhaoxiang Zhang, Shiming Xiang, Qi Tian, and Chunhong Pan. Pro-tuning: Unified prompt tuning for vision tasks. *arXiv preprint arXiv:2207.14381*, 2022. 2
[64] Yuzhen Niu, Yujie Geng, Xueqing Li, and Feng Liu. Leveraging stereopsis for saliency analysis. In *CVPR*, pages 454–461, 2012. 5, 7
[65] Peter Ochs, Jitendra Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. *IEEE TPAMI*, 36(6):1187–1200, 2013. 5, 8
[66] Youwei Pang, Xiaoqi Zhao, Tian-Zhu Xiang, Lihe Zhang, and Huchuan Lu. Zoom in and out: A mixed-scale triplet network for camouflaged object detection. In *CVPR*, pages

2160–2170, 2022. 1, 2
[67] Youwei Pang, Xiaoqi Zhao, Lihe Zhang, and Huchuan Lu. Multi-scale interactive network for salient object detection. In *CVPR*, pages 9413–9422, 2020. 2
[68] Youwei Pang, Xiaoqi Zhao, Lihe Zhang, and Huchuan Lu. Caver: Cross-modal view-mixed transformer for bi-modal salient object detection. *IEEE TIP*, 32:892–904, 2023. 7, 8
[69] Houwen Peng, Bing Li, Weihua Xiong, Weiming Hu, and Rongrong Ji. Rgbd salient object detection: A benchmark and algorithms. In *ECCV*, pages 92–109, 2014. 5, 7
[70] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, pages 724–732, 2016. 5, 8
[71] Yongri Piao, Wei Ji, Jingjing Li, Miao Zhang, and Huchuan Lu. Depth-induced multi-scale recurrent attention network for saliency detection. In *ICCV*, pages 7254–7263, 2019. 1, 2, 5, 7
[72] Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jagersand. Basnet: Boundary-aware salient object detection. In *CVPR*, pages 7479–7489, 2019. 2
[73] Sucheng Ren, Chu Han, Xin Yang, Guoqiang Han, and Shengfeng He. Tenet: Triple excitation network for video salient object detection. In *ECCV*, pages 212–228. Springer, 2020. 2
[74] Hongmei Song, Wenguan Wang, Sanyuan Zhao, Jianbing Shen, and Kin-Man Lam. Pyramid dilated deeper convlstm for video salient object detection. In *ECCV*, pages 715–731, 2018. 2
[75] Yukun Su, Jingliang Deng, Ruizhou Sun, Guosheng Lin, Hanjing Su, and Qingyao Wu. A unified transformer framework for group-based segmentation: Co-segmentation, co-saliency detection and video salient object detection. *IEEE TMM*, 2023. 8
[76] Zhengzheng Tu, Zhun Li, Chenglong Li, Yang Lang, and Jin Tang. Multi-interactive dual-decoder for rgb-thermal salient object detection. *IEEE TIP*, 30:5678–5691, 2021. 2, 6, 7, 8
[77] Zhengzheng Tu, Yan Ma, Zhun Li, Chenglong Li, Jieming Xu, and Yongtao Liu. Rgbt salient object detection: A large-scale dataset and benchmark. *IEEE TMM*, 2022. 1, 2, 5, 6, 7
[78] Zhengzheng Tu, Tian Xia, Chenglong Li, Xiaoxiao Wang, Yan Ma, and Jin Tang. Rgb-t image saliency detection via collaborative graph learning. *IEEE TMM*, 22(1):160–173, 2019. 5, 7
[79] Guizhao Wang, Chenglong Li, Yunpeng Ma, Aihua Zheng, Jin Tang, and Bin Luo. Rgb-t saliency detection benchmark: Dataset, baselines, analysis and a novel approach. In *IJIG*, pages 359–369. Springer, 2018. 5, 7
[80] Jie Wang, Kechen Song, Yanqi Bao, Liming Huang, and Yunhui Yan. Cgfnet: Cross-guided fusion network for rgb-t salient object detection. *IEEE TCSVT*, 32(5):2949–2961, 2021. 1, 2
[81] Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. Learning to detect salient objects with image-level supervision. In *CVPR*, pages 136–145, 2017. 5, 6, 7
[82] Linzhao Wang, Lijun Wang, Huchuan Lu, Pingping Zhang, and Xiang Ruan. Salient object detection with recurrent fully convolutional networks. *IEEE TPAMI*, 41(7):1734–1746, 2018. 2
[83] Tiantian Wang, Ali Borji, Lihe Zhang, Pingping Zhang, and Huchuan Lu. A stagewise refinement model for detecting salient objects in images. In *ICCV*, pages 4019–4028, 2017. 1, 2
[84] Wenguan Wang, Jianbing Shen, Xingping Dong, and Ali Borji. Salient object detection driven by fixation prediction. In *CVPR*, pages 1711–1720, 2018. 2
[85] Wenguan Wang, Jianbing Shen, and Ling Shao. Consistent video saliency using local gradient flow optimization and global refinement. *IEEE TIP*, 24(11):4185–4196, 2015. 5, 8
[86] Wenguan Wang, Jianbing Shen, and Ling Shao. Video salient object detection via fully convolutional networks. *IEEE TIP*, 27(1):38–49, 2017. 1, 2
[87] Xinlong Wang, Wen Wang, Yue Cao, Chunhua Shen, and Tiejun Huang. Images speak in images: A generalist painter for in-context visual learning. In *CVPR*, pages 6830–6839, 2023. 3
[88] Xinlong Wang, Xiaosong Zhang, Yue Cao, Wen Wang, Chunhua Shen, and Tiejun Huang. Seggpt: Segmenting everything in context. *arXiv preprint arXiv:2304.03284*, 2023. 3
[89] Yi Wang, Ruili Wang, Xin Fan, Tianzhu Wang, and Xiangjian He. Pixels, regions, and objects: Multiple enhancement for salient object detection. In *CVPR*, pages 10031–10040, 2023. 7, 8
[90] Jun Wei, Shuhui Wang, Zhe Wu, Chi Su, Qingming Huang, and Qi Tian. Label decoupling framework for salient object detection. In *CVPR*, pages 13025–13034, 2020. 2
[91] Lina Wei, Shanshan Zhao, Omar Farouk Bourahla, Xi Li, Fei Wu, Yueting Zhuang, Junwei Han, and Mingliang Xu. End-to-end video saliency detection via a deep contextual spatiotemporal network. *IEEE TNNLS*, 32(4):1691–1702, 2020. 1
[92] Zongwei Wu, Danda Pani Paudel, Deng-Ping Fan, Jingjing Wang, Shuo Wang, Cédric Demonceaux, Radu Timofte, and Luc Van Gool. Source-free depth for object pop-out. In *ICCV*, pages 1032–1042, 2023. 1, 2, 8
[93] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *NeurIPS*, 34:12077–12090, 2021. 8
[94] Jin Xie, Hisham Cholakkal, Rao Muhammad Anwer, Fahad Shahbaz Khan, Yanwei Pang, Ling Shao, and Mubarak Shah. Count-and similarity-aware r-cnn for pedestrian detection. In *ECCV*, pages 88–104. Springer, 2020. 1
[95] Bin Yan, Yi Jiang, Jiannan Wu, Dong Wang, Ping Luo, Zehuan Yuan, and Huchuan Lu. Universal instance perception as object discovery and retrieval. In *CVPR*, pages 15325–15336, 2023. 3
[96] Pengxiang Yan, Guanbin Li, Yuan Xie, Zhen Li, Chuan Wang, Tianshui Chen, and Liang Lin. Semi-supervised video salient object detection using pseudo-labels. In *ICCV*, pages 7284–7293, 2019. 8
[97] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *CVPR*, pages 1155–1162, 2013. 5, 7
[98] Charig Yang, Hala Lamdouar, Erika Lu, Andrew Zisserman,

and Weidi Xie. Self-supervised video object segmentation by motion grouping. In *ICCV*, pages 7177–7188, 2021. 8
[99] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *CVPR*, pages 3166–3173, 2013. 5, 7
[100] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833. Springer, 2014. 4
[101] Qiang Zhai, Xin Li, Fan Yang, Chenglizhao Chen, Hong Cheng, and Deng-Ping Fan. Mutual graph learning for camouflaged object detection. In *CVPR*, pages 12997–13007, 2021. 1, 2, 8
[102] Dingwen Zhang, Junwei Han, Yu Zhang, and Dong Xu. Synthesizing supervision for learning deep saliency network without human annotation. *IEEE TPAMI*, 42(7):1755–1769, 2019. 2
[103] Jing Zhang, Deng-Ping Fan, Yuchao Dai, Xin Yu, Yiran Zhong, Nick Barnes, and Ling Shao. Rgb-d saliency detection via cascaded mutual information minimization. In *ICCV*, 2021. 7, 8
[104] Lu Zhang, Jianming Zhang, Zhe Lin, Huchuan Lu, and You He. Capsal: Leveraging captioning to boost semantics for salient object detection. In *CVPR*, pages 6024–6033, 2019. 2
[105] Miao Zhang, Jie Liu, Yifei Wang, Yongri Piao, Shunyu Yao, Wei Ji, Jingjing Li, Huchuan Lu, and Zhongxuan Luo. Dynamic context-sensitive filtering network for video salient object detection. In *ICCV*, pages 1553–1563, 2021. 8
[106] Miao Zhang, Weisong Ren, Yongri Piao, Zhengkun Rong, and Huchuan Lu. Select, supplement and focus for rgb-d saliency detection. In *CVPR*, pages 3472–3481, 2020. 2
[107] Qiang Zhang, Nianchang Huang, Lin Yao, Dingwen Zhang, Caifeng Shan, and Jungong Han. Rgb-t salient object detection via fusing multi-level cnn features. *IEEE TIP*, 29:3321–3335, 2019. 1, 2
[108] Xiaoning Zhang, Tiantian Wang, Jinqing Qi, Huchuan Lu, and Gang Wang. Progressive attention guided recurrent network for salient object detection. In *CVPR*, pages 714–722, 2018. 1, 2
[109] Jia-Xing Zhao, Yang Cao, Deng-Ping Fan, Ming-Ming Cheng, Xuan-Yi Li, and Le Zhang. Contrast prior and fluid pyramid integration for rgbd salient object detection. In *CVPR*, pages 3927–3936, 2019. 1, 2
[110] Jia-Xing Zhao, Jiang-Jiang Liu, Deng-Ping Fan, Yang Cao, Jufeng Yang, and Ming-Ming Cheng. Egnet:edge guidance network for salient object detection. In *ICCV*, pages 8779–8788, 2019. 2
[111] Wangbo Zhao, Kepan Nan, Songyang Zhang, Kai Chen, Dahua Lin, and Yang You. Learning referring video object segmentation from weak annotation. *arXiv preprint arXiv:2308.02162*, 2023. 1
[112] Wangbo Zhao, Jiasheng Tang, Yizeng Han, Yibing Song, Kai Wang, Gao Huang, Fan Wang, and Yang You. Dynamic tuning towards parameter and inference efficiency for vit adaptation. *arXiv preprint arXiv:2403.11808*, 2024. 2
[113] Xiaoqi Zhao, Youwei Pang, Lihe Zhang, Huchuan Lu, and Lei Zhang. Suppress and balance: A simple gated network for salient object detection. In *ECCV*, pages 35–51, 2020. 2
[114] Dehua Zheng, Xiaochen Zheng, Laurence T Yang, Yuan Gao, Chenlu Zhu, and Yiheng Ruan. Mffn: Multi-view feature fusion network for camouflaged object detection. In *WACV*, pages 6232–6242, 2023. 2
[115] Jiawen Zhu, Simiao Lai, Xin Chen, Dong Wang, and Huchuan Lu. Visual prompt multi-modal tracking. In *CVPR*, pages 9516–9526, 2023. 3
[116] Mingchen Zhuge, Deng-Ping Fan, Nian Liu, Dingwen Zhang, Dong Xu, and Ling Shao. Salient object detection via integrity learning. *IEEE TPAMI*, 2022. 1, 6, 7, 8
[117] Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Behl, Jianfeng Wang, Lu Yuan, et al. Generalized decoding for pixel, image, and language. In *CVPR*, pages 15116–15127, 2023. 3