Co-Salient Object Detection with Semantic-Level Consensus Extraction and Dispersion

Peiran Xu
Peking University
Beijing, China
2301112093@pku.edu.cn

Yadong Mu*
Peking University
Beijing, China
muyadong@gmail.com

ABSTRACT

Given a group of images, co-salient object detection (CoSOD) aims to highlight the common salient object in each image. There are two factors closely related to the success of this task, namely consensus extraction, and the dispersion of consensus to each image. Most previous works represent the group consensus using local features, while we instead utilize a hierarchical Transformer module for extracting semantic-level consensus. Therefore, it can obtain a more comprehensive representation of the common object category, and exclude interference from other objects that share local similarities with the target object. In addition, we propose a Transformer-based dispersion module that takes into account the variation of the co-salient object in different scenes. It distributes the consensus to the image feature maps in an image-specific way while making full use of interactions within the group. These two modules are integrated with a ViT encoder and an FPN-like decoder to form an end-to-end trainable network, without additional branch and auxiliary loss. The proposed method is evaluated on three commonly used CoSOD datasets and achieves state-of-the-art performance.

CCS CONCEPTS

• **Computing methodologies → Computer vision**; **Interest point and salient region detections**.

KEYWORDS

co-salient object detection, Transformer

ACM Reference Format:
Peiran Xu and Yadong Mu. 2023. Co-Salient Object Detection with Semantic-Level Consensus Extraction and Dispersion. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23), October 29–November 3, 2023, Ottawa, ON, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3581783.3612133

1 INTRODUCTION

Co-salient object detection (CoSOD) is a group-based image understanding task that aims to capture the common salient object presented in each image within the given group. Due to its wide spectrum of applications in object detection, semantic segmentation, and image retrieval, significant research efforts have been invested within this field over the past decade.

Conventionally, methods for solving this task rely on hand-crafted visual features of the images. [36] is one of the first attempts to apply deep learning to this problem. From then on, a variety of network structures have been proposed and greatly improved the performance of this task. Most of these works could be formalized as a four-stage paradigm, which involves (1) image feature representation through an encoder, (2) consensus extraction performed on a group of image features, (3) the dispersion of consensus to individual image features, and (4) co-saliency mask generation through a decoder. Among them, consensus extraction and dispersion (similar to the term "summarization and search" in [42]) are two key steps and have been implemented in diverse ways in previous works. We will give a detailed review on these methods in Section 2.

In this work, we also follow the four-stage paradigm since it is basically consistent with the human cognitive process. Despite significant breakthroughs achieved in this line of work, we have observed two common issues among the majority of prior research. Firstly, most previous works obtain the consensus by selecting or averaging local features [10, 12, 27, 34, 44–47], which can capture the representative region of the common object. However, it cannot provide complete, semantic-level information on the common object's category. Thus, such methods may fail when encountering large intra-class differences, and may highlight some irrelevant objects that have local similarities to the target object but do not belong to the same category. Moreover, the majority of previous methods implement the dispersion of consensus to image features by concatenation, addition, element-wise multiplication, dot-product, or their combination [10, 12, 16, 27, 28, 30, 33, 34, 38–40, 44–47]. Nevertheless, the common object may present very different attributes and appearances in different images. Therefore, simply adding or multiplying consensus to the image feature maps in the same way may not reflect this diversity, resulting in poor performance on challenging examples. Though some earlier works [14, 42] have taken the image-specific variation of consensus into account explicitly or implicitly, they fail to maintain a compact and consistent consensus representation, which may lead to inconsistent detection targets and increased computation. We argue that it is important to balance the commonality and the specificity, while controlling the computational complexity in a reasonable range.

Based on these considerations, we develop two novel blocks for semantic-level comprehensive consensus extraction and image-specific dispersion. In particular, a hierarchical Transformer module

*Corresponding author.

is designed for consensus extraction. It alternates between aggregating each image's salient object information to its corresponding class token, and aggregating a group of class tokens to the consensus representation. In addition, an image-specific dispersion module is put forward. It first refines the consensus vector with cross-attention to add more context details to the consensus. Then it performs dispersion through several Transformer blocks. Interactions within the group are also introduced in this Transformer, so the easy cases may help the difficult ones to achieve a proper dispersion result. Besides, we leverage a pre-trained Vision Transformer [5, 24] as the encoder to provide high-quality features, and an FPN-like network to decode the co-saliency masks.

In addition to the four-stage main branch for co-saliency masks generation, recent CoSOD methods tend to introduce auxiliary tasks or side pathways to enhance performance, such as additional classification task [10, 27, 38, 45], salient object detection (SOD) priors [8, 14, 43], various kinds of contrastive learning [10, 34, 45, 46], discriminator [46] and so on. These techniques contribute to the gain in the final performance, but at the cost of complicated model structure and abundant hyper-parameters. Moreover, some of them require extra supervision in the training phase (*e.g.* class label [10, 27, 38, 45], single image saliency maps (SISM) [14]). In contrast to this trend, our proposed model does not include any additional designs beyond the mask generation pipeline. Experiments show that our compact model with consensus extraction and dispersion is able to achieve comparable or superior performance compared to the previous models with complex modules.

To sum up, our contributions are as follows:

- We design a hierarchical consensus extraction module that leverages high-level semantics embedded in the class token to obtain a comprehensive representation of the common object category, thus avoiding the ambiguity and incompleteness brought by using local information as consensus. It also includes iterative refinements to gradually improve the image semantics and the consensus representation.
- This work proposes a Transformer-based dispersion module that distributes the consensus to the image feature maps in an image-specific manner, taking the diversity of the common object in different images into account. It also enables intra-group interactions during the dispersion process, thus maintaining a balance between the commonality and the specificity.
- The two novel modules are combined with a plain ViT encoder and an FPN-like decoder to form an end-to-end trainable CoSOD network. Without additional structure and auxiliary loss, the proposed method is characterized by its simplicity and clarity. Extensive experiments on several public datasets have proven the effectiveness of our model.

2 RELATED WORKS

Most of the deep-learning-based CoSOD methods can be categorized into the four-stage pipeline, namely image encoding, consensus extraction, consensus dispersion, and mask decoding. A brief summary of each of above ingredients is presented below.

Encoder and Decoder. For the encoder, a large body of prior works use VGG [25] as a defaulted choice. More recent research has explored the use of the Transformer architecture. For instance, [12] uses a T2T-ViT [35] branch accompanied with a CNN branch. [46] uses a single PVTv2 [32]. [41] utilizes a Swin Transformer [22]. As for the decoder, the idea of feature pyramid [20] is often utilized. Multi-scale feature maps generated by the encoder are sent to the decoder, and the predicted masks are obtained by aggregating the maps of different scales. Although some improved designs may be introduced to these two modules (*e.g.* Transformer necks between the encoder and the decoder [27], decoder with intra-group interactions [43], decoder that explicitly focuses on edges [38]), we prioritize the modules directly related to the group consensus. Our proposed network uses a plain ViT [5] as the encoder to get high-quality image feature map and semantic vector (*i.e.* the class token), and a simple CNN-based FPN-like network as the decoder to better capture the local information such as boundaries. Such methodology decouples the task-specific designs for CoSOD with the general feature extractor and dense predictor.

Consensus Extraction and Dispersion. These two components are the key to solving CoSOD, and represent the core of the network. Therefore, previous studies have proposed various implementations. As for consensus extraction, the key lies in integrating the features of a group of images. [33] concatenates the feature maps of the group and performs convolutional operations on them to get consensus. [16] uses an RNN to sequentially read the feature map of each image, and the hidden state is used as consensus after the whole group is processed. [8, 39] adopt PCA to get an optimal direction (which can be seen as some type of consensus representation), and project the image feature maps to it. [30] uses Low-Rank Bilinear Pooling [15] to obtain the consensus vector. [43] first fuses each image's SISM prediction with its feature map, then performs consensus extraction by a Group-Attentional Semantic Aggregation module that includes concatenation, block shuffle, atrous convolution, and self-attention. [27, 44] apply global average pooling (GAP) to the group of feature maps to get the consensus representation. [40] and [38] also employ GAP, but prior to that, they process the group of feature maps using GCN and Gromov-Wasserstein-distance-based matching [26], respectively. [47] first predicts SISMs and performs GAP to the salient regions. Then, K pixels with the highest similarity to the pooled feature within the group are selected and concatenated as the consensus. [10, 12, 34, 45, 46] utilize the attention mechanism to compute correspondences between each pair of pixels in the group. The attention matrix is then processed with pooling and softmax operation and becomes a per-pixel weight, which is used to perform weighted average pooling on the image feature maps and obtain the consensus representation.

The dispersion of the consensus representation to the image feature maps is typically defined as concatenation, addition, element-wise multiplication, dot-product, or their combination [10, 12, 16, 27, 28, 30, 33, 34, 38–40, 44–47]. On this basis, [44] computes the gradient of the dot-product results with respect to the feature maps, and multiplies the pooled gradient maps to the feature maps; [40] utilizes weighted K-means algorithm to refine the dot-product results; [12, 34] first multiplies the consensus to the feature maps, then enhances the fused maps using self-attention. [14, 42, 43] notice the importance of image-specific dispersion, which is similar to our ideas. Specifically, [43] multiplies an element-wise weight to the consensus before adding it to the feature map of each image, and the weight is computed by a squeeze-and-excitation block. [14] does not produce an explicit consensus representation. Instead, it extracts salient object representations for each image separately under the guidance of SISMs. Cosine similarity is calculated between each image feature map and each representation vector, so N masks are obtained for each of the N images (N is the size of the group). Next, it integrates the N masks for each image through weighted averaging (weights are determined by the similarity between them), and the resulting single mask is multiplied with the image feature map. [42] uses convolution and self-attention operations to transform the group of feature maps into a series of kernels (including both image-specific and group-shared ones), and implements the dispersion process by dynamic convolution.

Different from the above approaches, we put forward two novel Transformer-based modules to perform extraction at the semantic level, and implement image-specific dispersion while maintaining a consistent consensus representation.

Extra Modules. Additionally, many prior works have also incorporated various additional modules or auxiliary tasks to further enhance the performance and stabilize the training. [16] draws inspiration from the perceptual loss in neural style transfer [11] to ensure the consistency of the predicted foreground and the groundtruth. [8, 39] require complicated refinements to be applied to the model output. [42, 44] design novel data augmentation approaches for training images. [28, 43, 47] co-train the CoSOD model with an SOD branch. [8, 14] use SISMs predicted by another SOD model. [46] adds a discriminator at the end of the pipeline to perform adversarial learning. To enhance the consensus learning, [10, 27, 30, 38, 45] employ the image category labels and carry out an additional classification task on the extracted consensus; [10, 45] utilize contrastive learning between different groups while [46] further proposes to use memory-based contrastive learning; [34] explores self-contrastive learning by masking the co-salient object or the remaining area; [47] involves iterative refinement and purification.

Instead of investing efforts in this regard, we argue that these additional designs are not a necessity. The pipeline of our model is simple and direct, requiring only a minimal number of hyperparameters and loss functions.

Vision Transformer. Originally proposed in the field of NLP, Transformer [29] leverages the attention mechanism to process sequential data. Its notable application in computer vision, Vision Transformer (ViT) [5], has shown superior performance to conventional CNN due to its ability to model global correspondence in the image. Afterward, a large number of ViT variants and related techniques have been proposed, such as [2, 9, 13, 19, 22, 31, 32, 35]. They focus on introducing more inductive bias, improving the scalability, and designing new pre-training schemes. Though more advanced architecture may further boost the performance, we use the original ViT as backbone to keep the model simple. Such choice is also coincident with the philosophy of [3, 18] that seeks to decouple ViT pre-training and downstream fine-tuning.

Apart from the encoder, Transformer blocks are also employed in the consensus extraction and dispersion stages of our model. In this aspect, [28] shares a similar idea with our method. It simply concatenates a group of feature maps together and feeds them into a Transformer to obtain the consensus. Then it concatenates the consensus with the feature map of each image and feeds them into another Transformer for dispersion. Such an approach does not explicitly reflect the meaning of consensus, and incurs significant computation and memory overhead. In contrast, our model leverages the high-level semantics embedded in the class token to effectively obtain a consensus representation that is better aligned with the co-object's category.

In addition, several successful attempts have also been made to introduce Transformer to SOD (*e.g.* [21, 23]), and RGB-D CoSOD (*e.g.* [41]). Our proposed CoSOD method differs from these works in that it places greater emphasis on inter-image interactions without leveraging the depth information.

3 METHOD

The goal of CoSOD can be formalized as follows: given a group of images $\{I^n\}_{n=1}^N$ that contain objects of the same category, a group of co-saliency masks $\{M^n\}_{n=1}^N$ should be generated that highlight the co-salient object in each image. Here the input images are resized to the same shape, so $I^n \in \mathbb{R}^{3\times H\times W}$, while $M^n \in [0,1]^{1\times H\times W}$.

The overall pipeline of our proposed network is presented in Figure 1. As stated in Section 1, four stages are involved in the network. The L-layer ViT encoder first takes $\{I^n\}_{n=1}^N$ as the input and generates feature maps[1] $\left\{F^{L,n}\right\}_{n=1}^N$, $F^{L,n} \in \mathbb{R}^{C\times h\times w}$. Then the consensus extraction module extracts the representation of the common object class $\boldsymbol{g} \in \mathbb{R}^D$. The consensus dispersion module applies $\boldsymbol{g}$ to each feature map, and finally the decoder takes the fused maps and outputs mask predictions $\{M^n\}_{n=1}^N$. It should be noted that the above pipeline is also a general procedure in most of the previous works, though many of them introduce some additional branches or tasks to it.

The whole model is trained in an end-to-end fashion. Given the predictions $\{M^n\}_{n=1}^N$ and the ground-truth masks $\{G^n\}_{n=1}^N$, the objective is to minimize the Binary Cross Entropy (BCE) loss

$$L_{\text{BCE}} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{i,j}(G^n_{i,j}\log M^n_{i,j} + (1-G^n_{i,j})\log(1-M^n_{i,j})), \quad (1)$$

and Intersection over Union (IoU) loss

$$L_{\text{IoU}} = 1 - \frac{1}{N}\sum_{n=1}^{N}\frac{\sum_{i,j} G^n_{i,j}M^n_{i,j}}{\sum_{i,j}(G^n_{i,j} + M^n_{i,j} - G^n_{i,j}M^n_{i,j})}. \quad (2)$$

Putting these two together we have

$$L_{\text{total}} = L_{\text{BCE}} + L_{\text{IoU}}. \quad (3)$$

The utilization of these loss functions is a common practice in CoSOD methods [12, 27, 38, 45, 46].

In the following subsections, we first describe the encoder and decoder in details. Based on the ViT feature, a simple yet effective baseline model is proposed. Next, we introduce the newly-designed extraction and dispersion modules, and demonstrate how to incorporate them into the baseline model to further enhance the performance.

[1]"Feature map" (in $\mathbb{R}^{C\times h\times w}$) is commonly used in the context of CNN, while "patch tokens" (in $\mathbb{R}^{hw\times C}$) are suitable for the context of ViT. They can be converted to each other through the operations of flattening and reshaping. In this paper, we consider these two terms interchangeably.

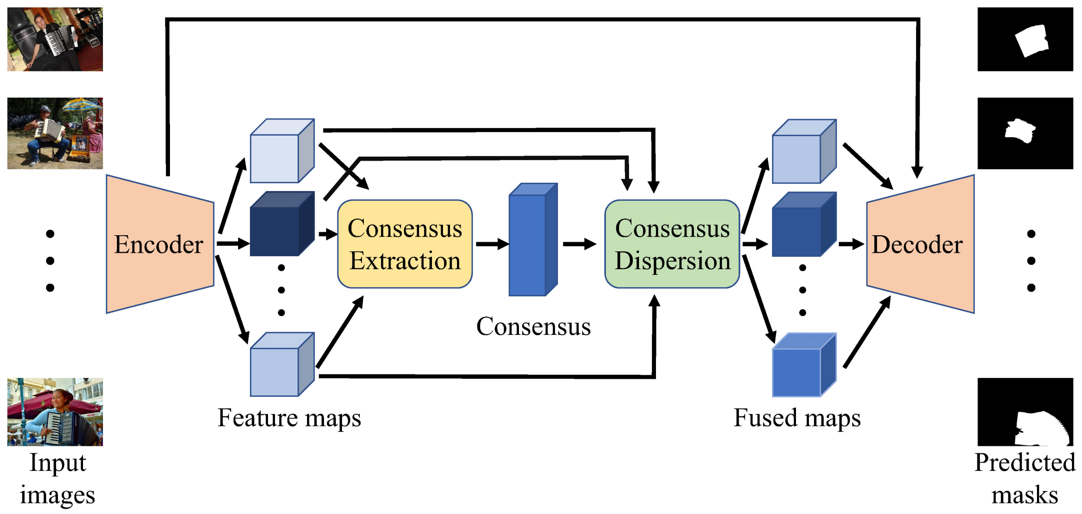

Figure 1: The pipeline of the proposed co-salient object detection model.

3.1 Encoder

We use a plain ViT [5] as the image encoder. The input image is first divided into non-overlapping patches and linearly projected to obtain a sequence of token embeddings. These patch tokens along with a learnable class token are then fed into multiple Transformer layers. The Transformer architecture enables efficient parallel processing and effective global interactions.

Previously, CoSOD models that use Transformer-base encoders typically employ variants of ViT that are better suited for dense prediction tasks, *e.g.* PVTv2 [32] in [46], T2T-ViT [35] in [12], Swin [22] in [41]. On the contrary, the use of original ViT decouples the design of the pre-training process and the need of downstream tasks, as mentioned in [3, 18]. Therefore, our model is fully compatible with the research progress in pre-training, scalability, and other aspects of ViT. In this paper, we choose the ViT model pre-trained with CLIP [24]. CLIP adds a linear projection to the original ViT's output, and the projected class token is treated as the CLIP image feature vector. Meanwhile, CLIP utilizes another Transformer to encode the sentence corresponding to the image into a CLIP text vector, and performs cross-modal contrastive learning. We choose CLIP's ViT weight for two reasons. Firstly, the large-scale contrastive pre-training of CLIP will endow the image features with rich and high-quality semantic information. Secondly, the sentence supervision used by CLIP provides a more complete description of the images compared to other supervision signals such as category labels, which ensures that image features will not miss crucial information about the co-object when the scene is complex. More advanced pre-training schemes can be easily introduced to our model by simply changing the encoder's initial weight.

3.2 Decoder

Following the previous works [10, 27, 38, 45–47], we adopt an FPN-like [20] decoder, whose input includes not only the image feature maps fused with consensus, but also the output of intermediate layers in the encoder. Unlike CNNs and some ViT variants that have hierarchical architecture, the intermediate features of ViT remain at the same scale throughout the network, and the output of earlier layers does not necessarily contain finer-grained information than that of later layers. Thus, we ignore the order of these intermediate maps and simply sum them together. We do not use the multi-scale supervision like [10, 14, 34, 42, 44–46] either, *i.e.* the loss is computed only once. Specifically, the decoder works as:

$$\mathrm{Dec}(F^1, ..., F^K) = \mathrm{Conv}(\sum_{l=1}^{K} \mathrm{Up}^l(F^l)), \tag{4}$$

where Conv is a 1×1 convolution that reduces the channel number to one and produces the mask prediction; Up^l is a series of 3×3 convolutions and $2\times$ bilinear upsamplings that progressively expand the spatial dimension.

3.3 A Naive Approach

Based on the pre-trained encoder, a straightforward idea is to extract consensus from the CLIP feature vectors. As an initial try, we simply use the mean of these vectors after normalization as consensus, and disperse it to the image feature maps by element-wise multiplication. Formally, we first obtain the CLIP feature vectors:

$$v^n_{\mathrm{CLIP}} = \mathrm{Linear}_{\mathrm{CLIP}}(c^{L,n}), n = 1, ..., N, \tag{5}$$

where $c^{L,n}$ is the class token in the ViT's final output, $\mathrm{Linear}_{\mathrm{CLIP}}$ is CLIP's additional linear projection. Next, we conduct the averaging operation:

$$\boldsymbol{g} = \frac{1}{N} \sum_{n=1}^{N} \mathrm{Norm}(\boldsymbol{v}^{\boldsymbol{n}}_{\mathrm{CLIP}}), \tag{6}$$

where Norm is the $l2$ normalization. And

$$\hat{F}^{l,n} = F^{l,n} \cdot \mathrm{Linear}^l_{\mathrm{Dis}}(\boldsymbol{g}), l \in \{l_1, ..., l_K\}, n = 1, ..., N, \tag{7}$$

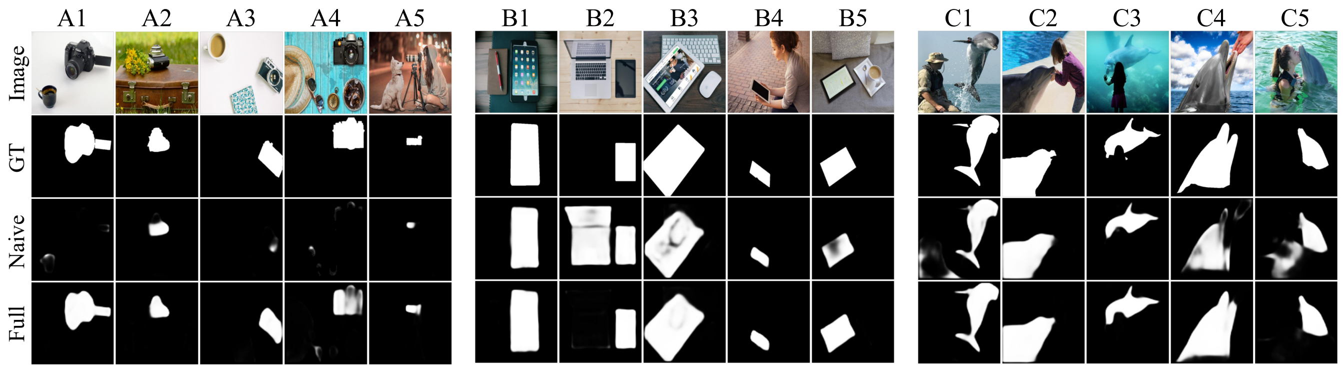

Figure 2: Failure cases of the naive approach. A/B/C represent the group of camera/tablet/dolphin, respectively. The input images, the ground-truth masks, the results of our naive model and full model are presented. In group A, the naive model cannot capture the target object in many cases. While in group B/C, the majority of the results are correct (*e.g.* B1/B4/C2/C3), but some predictions fail to fully cover the details of the co-object (B3/B5/C1/C4) or are interfered by other objects (B2/C1/C5).

$$M^n = \mathrm{Dec}(\hat{F}^{l_1,n}, ..., \hat{F}^{l_K,n}), n = 1, ..., N, \tag{8}$$

where $F^{l,n}$ is the output feature map of ViT's layer l; $\mathrm{Linear}_{\mathrm{Dis}}$ is a linear projection that adjusts the dimension of the consensus vector $\boldsymbol{g}$; $\hat{F}^{l,n}$ is the map after dispersion; $\{l_1, ..., l_K\} \subset \{1, ..., L\}$ are the selected intermediate layers' indices ($l_K = L$).

Note that in (7), we disperse $\boldsymbol{g}$ to not only $F^{L,n}$ but also all the inputs of the decoder, so as to reinforce the consensus signal during the decoding process. This multi-stage dispersion is also a commonly adopted procedure in prior works [27, 30, 38].

It is shocking that such a simple strategy has achieved a fairly competitive performance compared with previous works, as can be later seen in Table 1. The results demonstrate the quality of CLIP vectors and the effectiveness of the four-stage pipeline. In order to further improve the model, we examine the output masks, and some typical results are shown in Figure 2. Two types of problems can be identified. (I) In the first case, the predictions of *many* images in the group deviate significantly from the target object, indicating that there may be further room for improvement in the consensus extraction stage. (II) In the second case, most images' co-object is detected correctly, while *a few* hard examples make the model go wrong. It can be inferred that the consensus is well-extracted in these groups, but it is not sufficient to guide the model to capture the whole co-object or distinguish the co-object from interference in a complex scene.

3.4 Semantic-Level Consensus Extraction

We first modify Equation (6) to facilitate more advanced consensus extraction. In addition to the CLIP vectors, we also utilize the CLIP feature maps to include richer information, *i.e.*

$$F^{L,n}_{\mathrm{CLIP}} = \mathrm{Linear}_{\mathrm{CLIP}}(F^{L,n}), n = 1, ..., N. \tag{9}$$

Based on the feature maps $\left\{F^{L,n}_{\mathrm{CLIP}}\right\}_{n=1}^{N}$ and the semantic vectors $\{\boldsymbol{v}^{\boldsymbol{n}}_{\mathrm{CLIP}}\}_{n=1}^{N}$, two steps are performed alternatively, as shown in Figure 3. In the global step, a group-based semantic agreement Transformer (GSAT) layer refines the image semantics and extracts consensus simultaneously. Specifically, it takes the semantic vectors along with their average as input and performs self-attention:

$$[\bar{\boldsymbol{v}}^{(i)}, \left\{\boldsymbol{v}_g^{n,(i)}\right\}_{n=1}^{N}] = \mathrm{Trans}^{(i)}_{\mathrm{GSAT}}([\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{v}^{n,(i)}, \left\{\boldsymbol{v}^{n,(i)}\right\}_{n=1}^{N}]), \tag{10}$$

where $i = 1, ..., M + 1$ is the layer index; the initial $\boldsymbol{v}^{n,(1)}$ is $\boldsymbol{v}^{\boldsymbol{n}}_{\mathrm{CLIP}}$. In the local step, an image-based local information aggregation Transformer (ILIAT) layer comes into play. It processes each image individually, and can be defined with self-attention:

$$[\boldsymbol{v}^{n,(i+1)}, F^{L,n,(i+1)}_{\mathrm{CLIP}}] = \mathrm{Trans}^{(i)}_{\mathrm{ILIAT\text{-}SA}}([\boldsymbol{v}_g^{n,(i)}, F^{L,n,(i)}_{\mathrm{CLIP}}]), \tag{11}$$

or cross-attention:

$$\boldsymbol{v}^{n,(i+1)} = \mathrm{Trans}^{(i)}_{\mathrm{ILIAT\text{-}CA}}(\boldsymbol{v}_g^{n,(i)}, F^{L,n}_{\mathrm{CLIP}}, F^{L,n}_{\mathrm{CLIP}}), \tag{12}$$

where $i = 1, ..., M; n = 1, ..., N$; the three inputs of $\mathrm{Trans}_{\mathrm{ILIAT\text{-}CA}}$ are used as query, key, and value, respectively; the initial $F^{L,n,(1)}_{\mathrm{CLIP}}$ is $F^{L,n}_{\mathrm{CLIP}}$.

Intuitively, ILIAT processes local information and aggregates the salient object features of each individual image into its semantic vector. The GSAT layer facilitates semantic-level information exchange, extracting the common component from the semantics of individual images. It is similar in effect to the averaging operation in Equation (6), but the additional parameters and self-attention mechanism allow for more flexible interactions of the image semantics. GSAT also updates the semantic vector of each image, thus they can participate in the following ILIAT layer and provide more reliable guidance for the aggregation of local information.

After $M+1$ layers of GSAT and M layers of ILIAT, the last GSAT's output corresponding to the average semantic token is treated as the final consensus representation. To fully utilize the important semantic priors endowed by the pre-training of CLIP vectors, we also add Equation (6) to the consensus, *i.e.*,

$$\boldsymbol{g} = \bar{\boldsymbol{v}}^{(M+1)} + \frac{1}{N}\sum_{n=1}^{N}\mathrm{Norm}(\boldsymbol{v}^{\boldsymbol{n}}_{\mathrm{CLIP}}). \tag{13}$$

The proposed Transformer-based consensus extraction differs from the commonly used attention-based method [10, 12, 34, 45, 46] in that it leverages the ViT's class token to perform semantic-level

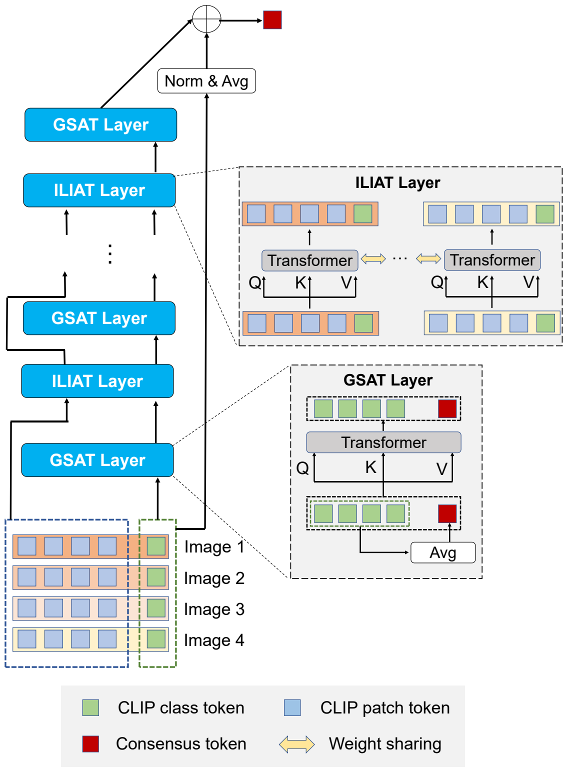

Figure 3: The proposed consensus extraction module. As for the ILIAT layer, a self-attention-based version is presented here, which can be replaced by a cross-attention-based version as in Equation (12).

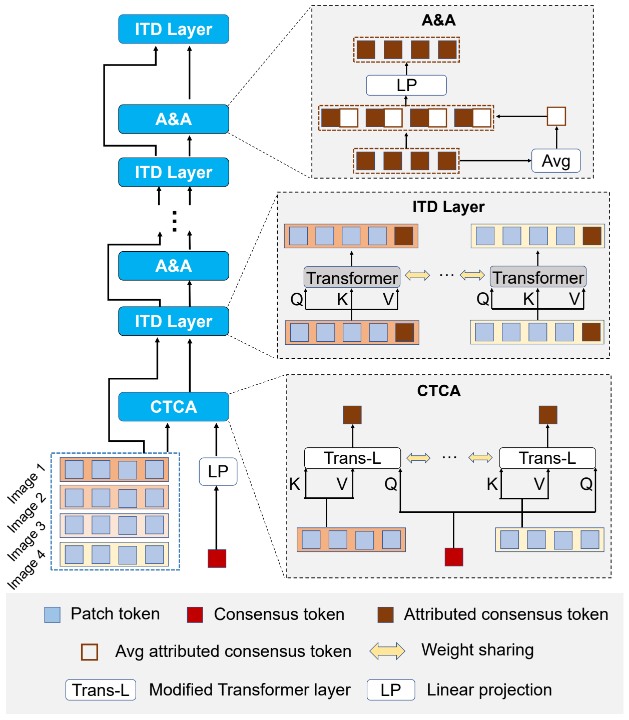

Figure 4: The proposed dispersion module.

aggregation and iterative refinement. Therefore, it is able to grasp the complete representation of the co-object without being easily disrupted by other objects that are locally similar to the target. It also reduces the cost of computing the attention matrix. More detailed comparisons can be found in the supplementary materials.

3.5 Image-Specific Consensus Dispersion

In order to better cope with the variation of the co-object, we design another Transformer-based module to perform the dispersion of consensus to image features, as shown in Figure 4. It first situates the consensus representation within the specific context of each image:

$$\tilde{g}^n = \text{Trans}_{\text{CTCA}}(\text{Linear}_{\text{CTCA}}(g), F^{L,n}, F^{L,n}), n = 1, ..., N, \quad (14)$$

where $\text{Trans}_{\text{CTCA}}$ is the contextualization Transformer layer with cross-attention (CTCA), and the three inputs are used as query, key, and value, respectively; $\text{Linear}_{\text{CTCA}}$ is a linear projection that adjusts the channel number. The output $\tilde{g}^n$ thus represents the image-specific status and attributions of the co-object in image I_n. There is a slight difference between $\text{Trans}_{\text{CTCA}}$ and the original Transformer layer. We multiply the results of cross-attention and MLP with a learnable channel-wise coefficient, thus allowing the network to adaptively learn the degree of contextualization.

Subsequently, the attributed consensus is concatenated with the corresponding image feature map and processed by multiple image-based Transformer layers for dispersion (ITD). Moreover, to further facilitate interactions within the group, we insert an aggregation and allocation (A&A) operation between adjacent ITD layers. It is implemented by concatenating the output consensus token with their average and applying a linear projection, as shown below in Equation (15). Through the communication of the attributed consensus, the easy cases (where the co-object can be effortlessly located) may help the difficult cases (where the co-object is concealed or interference exists) to better focus on the common object. Besides, the A&A process guarantees that the dispersion will not deviate significantly from the extracted consensus while ensuring the flexibility of the dispersion (*i.e.* multiple Transformer layers). A similar idea is also adopted by [43] in its decoder design.

Besides, we also retain the dispersion implemented by multiplication, since it has exhibited satisfactory performance and remains fully compatible with the improved pipeline. To sum up, Equation (7) in the naive approach is modified to Equation (14) followed by:

$$\begin{cases} [\tilde{g}_{\text{pre}}^{n,(i+1)}, F^{L,n,(i+1)}] = \text{Trans}_{\text{ITD}}^{(i)}([\tilde{g}^{n,(i)}, F^{L,n,(i)}]) \\ \tilde{g}^{n,(i+1)} = \text{Linear}_{\text{A\&A}}^{(i)}([\tilde{g}_{\text{pre}}^{n,(i+1)}, \frac{1}{N}\sum_{n=1}^{N}\tilde{g}_{\text{pre}}^{n,(i+1)}]) \end{cases}, \quad (15)$$

$$\hat{F}^{l,n} = F^{l,n} \cdot \text{Linear}_{\text{Dis}}^{l}(\tilde{g}_{\text{pre}}^{n,(M'+1)}), \quad (16)$$

$$\hat{F}^{L,n} = F^{L,n,(M'+1)} \cdot \text{Linear}_{\text{Dis}}^{L}(\tilde{g}_{\text{pre}}^{n,(M'+1)}), \quad (17)$$

where $l \in \{l_1, ..., l_{K-1}\}$; $n = 1, ..., N; i = 1, ..., M'$ is the layer index; the initial $\tilde{g}^{n,(1)} = \tilde{g}^n, F^{L,n,(1)} = F^{L,n}$.

Table 1: Performance comparision with the state-of-the-art methods. C/S/D stand for COCO-9213/COCO-SEG/DUTS-class dataset, respectively. The best and second-best results under each metric are marked in bold and underlined respectively.

	Training	CoCA [44]				CoSOD3k [8]				CoSal2015 [37]			
Method	Set	$E_\xi^{\max}\uparrow$	$S_\alpha\uparrow$	$F_\beta^{\max}\uparrow$	$\epsilon\downarrow$	$E_\xi^{\max}\uparrow$	$S_\alpha\uparrow$	$F_\beta^{\max}\uparrow$	$\epsilon\downarrow$	$E_\xi^{\max}\uparrow$	$S_\alpha\uparrow$	$F_\beta^{\max}\uparrow$	$\epsilon\downarrow$
DeepACG(CVPR21)[38]	S	0.771	0.688	0.552	0.102	0.838	0.792	0.756	0.089	0.892	0.854	0.842	0.064
GCoNet(CVPR21)[10]	D	0.760	0.673	0.544	0.105	0.860	0.802	0.777	0.071	0.887	0.845	0.847	0.068
CoEGNet(TPAMI21)[8]	D	0.717	0.612	0.493	0.106	0.825	0.762	0.736	0.092	0.882	0.836	0.832	0.077
CADC(ICCV21)[42]	C+D	0.744	0.681	0.548	0.132	0.840	0.801	0.759	0.096	0.906	0.866	0.862	0.064
CoSFormer(arxiv21)[28]	C+D	0.770	0.724	0.603	0.103	0.879	0.835	0.807	0.066	0.929	**0.894**	0.891	**0.047**
DCFM(CVPR22)[34]	C	0.783	0.710	0.598	0.085	0.874	0.810	0.805	0.067	0.892	0.838	0.856	0.067
UFO(TMM23)[27]	S	0.782	0.697	0.571	0.095	0.874	0.819	0.797	0.073	0.906	0.860	0.865	0.064
CoRP(TPAMI23)[47]	C+D	0.741	0.703	0.575	0.110	0.866	0.825	0.801	0.072	0.915	0.877	0.888	$\underline{0.049}$
GCoNet+(TPAMI23)[45]	S+D	$\underline{0.814}$	$\underline{0.738}$	**0.637**	$\underline{0.081}$	0.901	0.843	0.834	0.062	0.924	0.881	0.891	0.056
MCCL(AAAI23)[46]	S+D	0.796	0.714	0.590	0.103	0.903	$\underline{0.858}$	0.837	0.061	0.927	$\underline{0.890}$	0.891	0.051
Ours (naive)	S+D	0.804	0.707	0.598	**0.080**	$\underline{0.914}$	0.851	$\underline{0.847}$	$\underline{0.056}$	**0.942**	0.888	**0.903**	**0.047**
Ours (full model)	S+D	**0.831**	**0.741**	$\underline{0.631}$	$\underline{0.081}$	**0.922**	**0.863**	**0.857**	**0.054**	$\underline{0.941}$	$\underline{0.890}$	$\underline{0.902}$	**0.047**

4 EXPERIMENTS

4.1 Datasets

We evaluate the proposed method on three widely-used CoSOD datasets, including Cosal2015 [37], CoSOD3k [8], and CoCA [44]. Both of CoSOD3k and CoCA are challenging since there are many distracting objects in each group, and the co-object exhibits high diversity in different images. The metrics for evaluation include maximum E-measure [7], S-measure [6], maximum F-measure [1], and mean absolute error (MAE) [4]. We use the evaluation tools provided by GICD [44]. The evaluation is performed at the original scale of each image.

There are currently three supervised training sets for CoSOD, namely COCO-9213 [33], DUTS-class [44], and COCO-SEG [30]. Our model is trained on the combination of COCO-SEG and DUTS-class to keep pace with the state-of-the-art (SOTA) methods [45, 46].

4.2 Implementation Details

All images are rescaled to 224×224 as the input. The number of GSAT/ILIAT/ITD layers are set to 3/2/4, respectively. The batch size is set to 16 for training. During inference, the whole group is processed in a single pass regardless of its size. The model is trained using Adam optimizer for 15 epochs. The learning rate is set to 3e-5. We use the CLIP [24] pre-trained ViT-B/16 as the image encoder. To take full advantage of the pre-trained weight, we set the learning rate of the encoder to $\frac{1}{10}$ of the rest of the network.

All experiments are conducted on a single NVIDIA GeForce 3080Ti. Using the benchmark designed by [46], the inference speed is ~95fps, which is on par with recent SOTA methods. Please refer to the appendix for a detailed description on the model configuration.

4.3 Main Results

Quantitative Results. Table 1 shows the quantitative performances of our model and some representative works in recent years. To reduce the effect of randomness, we independently train our model three times and report the averaged performance. Among the previous works, CoSFormer [28], GCoNet+ [45], MCCL [46] achieve SOTA performance on CoSal2015, CoCA, CoSOD3k, respectively. Our naive model performs better than CoSFormer on CoSal2015 and better than MCCL on CoSOD3k (except for S-measure). Its results on CoCA are similar to that of CoSFormer and MCCL but inferior to that of GCoNet+. Equipped with the newly-designed extraction and dispersion modules, our full model demonstrates better performance. It surpasses the naive model on the two hard test sets (CoSOD3k and CoCA) while maintaining competitive performance on the simple one (CoSal2015). Specifically, the performance of the full model on CoCA is comparable to that of GCoNet+, while the pipeline and objective function of our model are much simpler than GCoNet+ and MCCL. The full model performs exceptionally well on CoSOD3k, surpassing all previous methods by a large margin (*e.g.* +1.9% max E-measure and +2.0% max F-measure). It also outperforms CoSFormer on most metrics, showing the benefits of designing task-specific Transformer modules.

Qualitative Results. Figure 5 shows some predicted masks of our model and previous SOTA methods. Five types of test images are considered here, including the normal cases and four typical kinds of difficult cases. "Tiny Object" requires the model not to miss the subtle details in the image. By leveraging the rich semantics in the CLIP vector and aggregating the local information in the ILIAT layer, our model manages to detect very small co-objects like bubbles. "Distraction" implies the existence of salient interfering objects in the image, while "Variation" means that the co-object exhibits very different appearances in different images. Methods that rely solely on local information may be misled by such complex scenes, resulting in detecting incorrect objects or the omission of certain parts of the co-object. In contrast, our method avoids these issues through semantic-level consensus extraction and image-specific dispersion, thus it effectively obtains the information of the co-object and conducts accurate detection. Lastly, "Complex Boundary" refers to the images with complicated co-object structures. Due to the lack of specialized modules for boundaries and the relatively low resolution of the network, our model does not perform perfectly on the fine structures, which is also a common problem in existing CoSOD methods. In future studies, we plan to refer to the latest

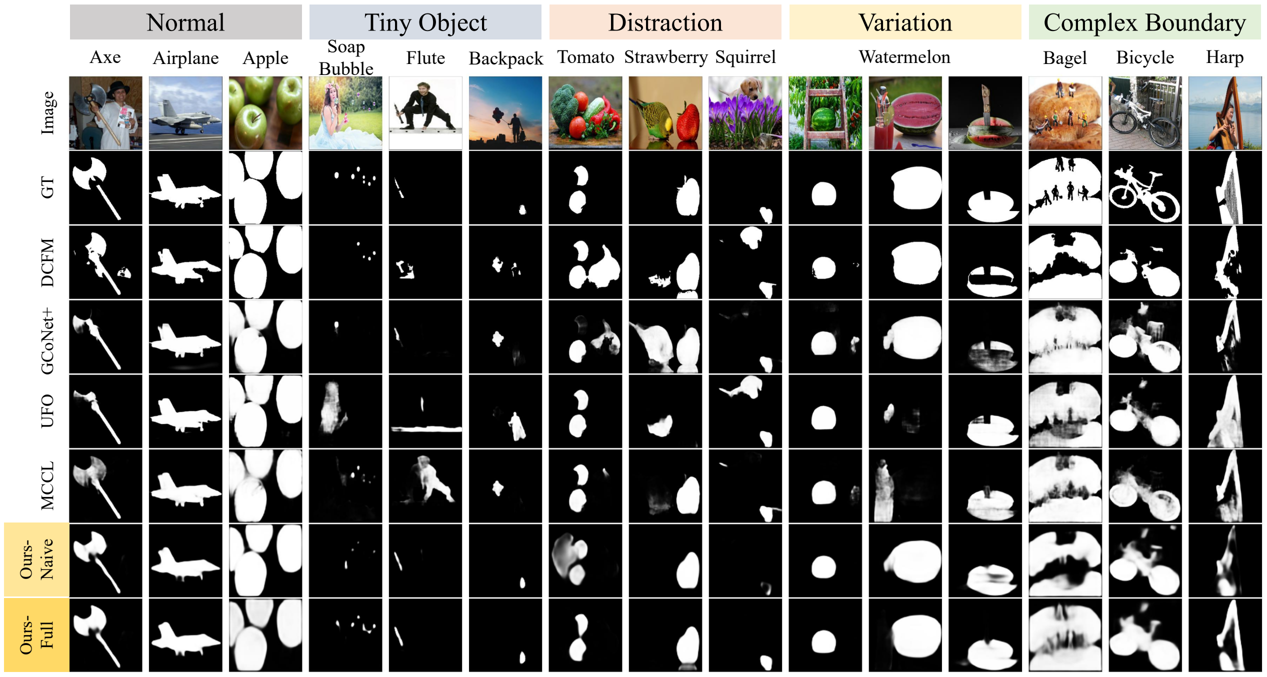

Figure 5: Qualitative results of the proposed method and some state-of-the-art methods.

Table 2: Impacts of the proposed modules.

Extraction	Dispersion	$E_\xi^{\max}\uparrow$	$S_\alpha\uparrow$	$F_\beta^{\max}\uparrow$
avg CLIP vec	multiply	0.804	0.707	0.598
GSAT	multiply	0.808	0.715	0.606
GSAT+ILIAT(CA)	multiply	0.806	0.721	0.610
GSAT+ILIAT(SA)	multiply	0.809	0.710	0.604
avg CLIP vec	**CTCA**	0.812	0.715	0.598
avg CLIP vec	**CTCA+ITD** (w/o A&A)	0.820	0.733	0.624
avg CLIP vec	**CTCA+ITD**	0.827	0.738	0.630
GSAT+ILIAT(CA)	**CTCA+ITD**	**0.831**	**0.741**	**0.631**

research in the field of image segmentation to obtain insights for addressing this issue.

4.4 Ablation Studies

We conduct experiments to examine the role of the proposed modules. Different choices of the extraction and dispersion method are tried as in Table 2. We report the results on CoCA since it contains the most challenging scenarios which are precisely the focus of our model design. It can be concluded that: (1) each module (GSAT, ILIAT, CTCA, ITD w/ A&A) can improve the performance of the naive method. (2) Combining these modules further enhances the performance, indicating the complementary nature of the extraction and dispersion process. (3) Using cross-attention in ILIAT exhibits better performance than using self-attention, suggesting that updating local features during consensus extraction is not necessary. (4) The improvement in performance brought by modifying the dispersion stage (especially introducing ITD) is greater than that of modifying the extraction stage, which means that designing a distribution strategy capable of handling complex scenes is more significant than improving the quality of the naive average-CLIP-vector-based consensus representation. In Section 3.3, we note that the naive model has fewer Type I failures (mainly due to extraction issues) than Type II failures (mainly due to dispersion issues), which is consistent with the experimental findings. Additional ablation experiments are provided in the supplementary materials.

5 CONCLUSION

We summarize the mainstream CoSOD methods into a general four-stage paradigm. Based on that, we note two major issues regarding the group consensus. We propose to focus on semantic-level information to extract a comprehensive consensus representation, and consider the image-specific variation of the co-object when dispersing the consensus to image features. We first present a simple yet effective model leveraging high-level semantics in CLIP vectors. Then an improved model is put forward that involves hierarchical consensus extraction, iterative semantics refinement, image-specific consensus attribution, and dispersion with cross-image interactions. The performance of the full model and the effectiveness of its modules are demonstrated by extensive experiments.

ACKNOWLEDGMENTS

This work is supported by Beijing Natural Science Foundation (Z190001), National Key R&D Program of China (2022ZD0160305) and a research grant from BOE Technology.

REFERENCES

[1] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Susstrunk. 2009. Frequency-tuned salient region detection. In *2009 IEEE conference on computer vision and pattern recognition*. IEEE, 1597–1604.

[2] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. 2022. BEiT: BERT Pre-Training of Image Transformers. In *International Conference on Learning Representations*. https://openreview.net/forum?id=p-BhZSz59o4

[3] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. 2023. Vision Transformer Adapter for Dense Predictions. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=plKu2GByCNW

[4] Ming-Ming Cheng, Jonathan Warrell, Wen-Yan Lin, Shuai Zheng, Vibhav Vineet, and Nigel Crook. 2013. Efficient salient region detection with soft image abstraction. In *Proceedings of the IEEE International Conference on Computer vision*. 1529–1536.

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*. https://openreview.net/forum?id=YicbFdNTTy

[6] Deng-Ping Fan, Ming-Ming Cheng, Yun Liu, Tao Li, and Ali Borji. 2017. Structure-measure: A new way to evaluate foreground maps. In *Proceedings of the IEEE international conference on computer vision*. 4548–4557.

[7] Deng-Ping Fan, Cheng Gong, Yang Cao, Bo Ren, Ming-Ming Cheng, and Ali Borji. 2018. Enhanced-alignment Measure for Binary Foreground Map Evaluation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, 698–704. https://doi.org/10.24963/ijcai.2018/97

[8] Deng-Ping Fan, Tengpeng Li, Zheng Lin, Ge-Peng Ji, Dingwen Zhang, Ming-Ming Cheng, Huazhu Fu, and Jianbing Shen. 2021. Re-thinking co-salient object detection. *IEEE transactions on pattern analysis and machine intelligence* 44, 8 (2021), 4339–4354.

[9] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. 2021. Multiscale vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 6824–6835.

[10] Qi Fan, Deng-Ping Fan, Huazhu Fu, Chi-Keung Tang, Ling Shao, and Yu-Wing Tai. 2021. Group collaborative learning for co-salient object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12288–12298.

[11] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2016. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2414–2423.

[12] Yanliang Ge, Qiao Zhang, Tian-Zhu Xiang, Cong Zhang, and Hongbo Bi. 2022. TCNet: Co-Salient Object Detection via Parallel Interaction of Transformers and CNNs. *IEEE Transactions on Circuits and Systems for Video Technology* (2022).

[13] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16000–16009.

[14] Wen-Da Jin, Jun Xu, Ming-Ming Cheng, Yi Zhang, and Wei Guo. 2020. Icnet: Intra-saliency correlation network for co-saliency detection. *Advances in Neural Information Processing Systems* 33 (2020), 18749–18759.

[15] Shu Kong and Charless Fowlkes. 2017. Low-rank bilinear pooling for fine-grained classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 365–374.

[16] Bo Li, Zhengxing Sun, Lv Tang, Yunhan Sun, and Jinlong Shi. 2019. Detecting Robust Co-Saliency with Recurrent Co-Attention Neural Network.. In *IJCAI*, Vol. 2. 6.

[17] Long Li, Junwei Han, Ni Zhang, Nian Liu, Salman Khan, Hisham Cholakkal, Rao Muhammad Anwer, and Fahad Shahbaz Khan. 2023. Discriminative Co-Saliency and Background Mining Transformer for Co-Salient Object Detection. arXiv:2305.00514 [cs.CV]

[18] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. 2022. Exploring plain vision transformer backbones for object detection. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX*. Springer, 280–296.

[19] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. 2022. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4804–4814.

[20] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2117–2125.

[21] Nian Liu, Ni Zhang, Kaiyuan Wan, Ling Shao, and Junwei Han. 2021. Visual saliency transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*. 4722–4732.

[22] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*. 10012–10022.

[23] Zhengyi Liu, Yuan Wang, Zhengzheng Tu, Yun Xiao, and Bin Tang. 2021. TriTransNet: RGB-D salient object detection with a triplet transformer embedding network. In *Proceedings of the 29th ACM international conference on multimedia*. 4481–4490.

[24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.

[25] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[26] Justin Solomon, Gabriel Peyré, Vladimir G Kim, and Suvrit Sra. 2016. Entropic metric alignment for correspondence problems. *ACM Transactions on Graphics (ToG)* 35, 4 (2016), 1–13.

[27] Yukun Su, Jingliang Deng, Ruizhou Sun, Guosheng Lin, Hanjing Su, and Qingyao Wu. 2023. A Unified Transformer Framework for Group-based Segmentation: Co-Segmentation, Co-Saliency Detection and Video Salient Object Detection. *IEEE Transactions on Multimedia* (2023), 1–13. https://doi.org/10.1109/TMM.2023.3264883

[28] Lv Tang and Bo Li. 2021. CoSformer: Detecting co-salient object with transformers. *arXiv preprint arXiv:2104.14729* (2021).

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[30] Chong Wang, Zheng-Jun Zha, Dong Liu, and Hongtao Xie. 2019. Robust deep co-saliency detection with group semantic. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 8917–8924.

[31] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. 2021. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*. 568–578.

[32] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. 2022. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media* 8, 3 (2022), 415–424.

[33] Lina Wei, Shanshan Zhao, Omar El Farouk Bourahla, Xi Li, and Fei Wu. 2017. Group-wise deep co-saliency detection. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 3041–3047.

[34] Siyue Yu, Jimin Xiao, Bingfeng Zhang, and Eng Gee Lim. 2022. Democracy does matter: Comprehensive feature mining for co-salient object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 979–988.

[35] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. 2021. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF international conference on computer vision*. 558–567.

[36] Dingwen Zhang, Junwei Han, Chao Li, and Jingdong Wang. 2015. Co-saliency detection via looking deep and wide. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2994–3002.

[37] Dingwen Zhang, Junwei Han, Chao Li, Jingdong Wang, and Xuelong Li. 2016. Detection of co-salient objects by looking deep and wide. *International Journal of Computer Vision* 120 (2016), 215–232.

[38] Kaihua Zhang, Mingliang Dong, Bo Liu, Xiao-Tong Yuan, and Qingshan Liu. 2021. Deepacg: Co-saliency detection via semantic-aware contrast gromov-wasserstein distance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13703–13712.

[39] Kaihua Zhang, Tengpeng Li, Bo Liu, and Qingshan Liu. 2019. Co-saliency detection via mask-guided fully convolutional networks with multi-scale label smoothing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3095–3104.

[40] Kaihua Zhang, Tengpeng Li, Shiwen Shen, Bo Liu, Jin Chen, and Qingshan Liu. 2020. Adaptive graph convolutional network with attention graph clustering for co-saliency detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9050–9059.

[41] Ni Zhang, Junwei Han, and Nian Liu. 2022. Learning implicit class knowledge for rgb-d co-salient object detection with transformers. *IEEE Transactions on Image Processing* 31 (2022), 4556–4570.

[42] Ni Zhang, Junwei Han, Nian Liu, and Ling Shao. 2021. Summarize and search: Learning consensus-aware dynamic convolution for co-saliency detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4167–4176.

[43] Qijian Zhang, Runmin Cong, Junhui Hou, Chongyi Li, and Yao Zhao. 2020. CoADNet: Collaborative aggregation-and-distribution networks for co-salient object detection. *Advances in neural information processing systems* 33 (2020), 6959–6970.

[44] Zhao Zhang, Wenda Jin, Jun Xu, and Ming-Ming Cheng. 2020. Gradient-induced co-saliency detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*. Springer, 455–472.

[45] Peng Zheng, Huazhu Fu, Deng-Ping Fan, Qi Fan, Jie Qin, and Luc Van Gool. 2022. GCoNet+: A Stronger Group Collaborative Co-Salient Object Detector. *arXiv preprint arXiv:2205.15469* (2022).
[46] Peng Zheng, Jie Qin, Shuo Wang, Tian-Zhu Xiang, and Huan Xiong. 2023. Memory-aided Contrastive Consensus Learning for Co-salient Object Detection. *arXiv preprint arXiv:2302.14485* (2023).
[47] Ziyue Zhu, Zhao Zhang, Zheng Lin, Xing Sun, and Ming-Ming Cheng. 2023. Co-Salient Object Detection with Co-Representation Purification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

A MODEL CONFIGURATIONS

In this section, we provide a detailed description of the specific structure and hyper-parameters choice of the proposed model.

A.1 Encoder

Following the ViT-B/16 in [5], the ViT encoder has 12 Transformer layers. The patch size is 16×16. The hidden dimension is 768, and the number of Transformer heads is 12. We use the pre-trained weights provided by CLIP [24]. To be consistent with CLIP, the output of the ViT’s final layer is processed with layer normalization and linear projection, resulting in $512d$ tokens.

A.2 Consensus Extraction and Dispersion

The hidden dimension of GSAT and ILIAT is 512, which is the same as the CLIP feature dimension. In CTCA, the $512d$ consensus is first linearly projected to $768d$. Then it participates in the cross-attention with the $768d$ ViT feature maps. The hidden dimension of ITD is set to 768, which is the same as the dimension of the feature maps. We intuitively set the number of GSAT layers to 3, the number of ILIAT layers to 2, and the number of ITD layers to 4. The number of Transformer heads is set to 8 in all these modules. The A&A module projects the $768d + 768d$ concatenated vector to $768d$. The projection before multiplying the consensus to the feature maps is $768d \longrightarrow 768d$ for the full model, and $512d \longrightarrow 768d$ for the naive model.

A.3 Decoder

The indices of the intermediate features that are sent to the decoder (denoted as $\{l_1, .., l_K\}$ in the paper) are $\{4, 6, 8, 12\}$. They are all of the shape $N \times 196 \times 768$, where N is the size of the group, $196 = \frac{224}{16} \times \frac{224}{16}$ is the number of tokens (the class token is discarded). In the full model, ITD is applied to the last group of features (*i.e.* index 12) and it does not change the shape. In the decoder, each group of features is first reshaped to $N \times 768 \times 14 \times 14$ and then processed by three consecutive CNN blocks. Each block contains a 3×3 convolution followed by batch normalization, ReLU activation, and a $2\times$ bilinear upsampling in sequence. The first block reduces the channel number to 256, while the following blocks do not change the channel dimension. After that, the four groups of features with shape $N \times 256 \times 112 \times 112$ are added together. The sum is sent to another 3×3 convolution and bilinearly upsampled again to get the final output.

A.4 Training Scheme

The code is implemented with PyTorch. During training, the batch size is set to 16 and we use the batch sampler designed by [14]. In each iteration, it randomly selects a group and samples 16 images from it. Batches with less than 16 images are allowed, but batches with only one image are discarded since CoSOD is ill-defined for a single image. No image will be sampled more than once in one epoch. During inference, each group is processed in a single pass, regardless of the group size. The input image is resized to 224×224 and normalized for both training and inference. The ground-truth is resized to 224×224 for training, and the loss is calculated at this scale. When evaluating the performance at the inference stage, the network output is resized to the image’s original size. We use the evaluation tools provided by [44][2]. No data augmentation is used during training, except for horizontal flipping with a probability of 50%.

We combine the DUTS_class [44] dataset and the COCO-SEG [30] dataset together, so there are 369 groups and 209k images in total. With the sub-group size (*i.e.* batch size) set to 16, each training epoch contains 13k iterations. We train the model for 15 epochs (about 200k iterations). We use Adam optimizer with weight decay set to 1e-4 and betas set to [0.9, 0.99] empirically. The learning rate is set to 3e-5 by a few trials.

A.5 Model Efficiency

Table 3: Model statistics.

Model	#params	FLOPs	inference time	fps
DCFM [34]	142.3M	31.7G	0.008s	~125
GCoNet+ [45]	18.4M	27.5G	0.009s	~120
MCCL [46]	27.0M	5.9G	0.017s	~60
DMT [17]	40.4M	84.4G	0.023s	~45
Ours-naive	99.6M	28.4G	0.007s	~150
Ours-full	156.7M	34.7G	0.010s	~95

Since CoSOD is a group-based task, the inference time is related to the size of the group. On a single NVIDIA GeForce 3080Ti, the speed of our full model is about 0.10s per group in CoCA [44] (about 16 images per group), 0.12s per group in CoSOD3k [8] (about 21 images per group), and 0.23s per group in CoSal2015 [37] (about 40 images per group). The speed of the naive model is about 0.09s per group in CoCA, 0.10s per group in CoSOD3k, and 0.20s per group in CoSal2015. We also evaluate the inference time and speed on the benchmark proposed by [46][3], and compare them with recent models. Following the original papers, we set the input shape of DCFM [34] and our models to 224×224, and the input shape of GCoNet+ [45]/MCCL [46]/DMT [17] to 256×256. We also utilize fvcore to calculate the FLOPs of these models[4], and count the number of their parameters. The results are shown in Table 3. Though our model is heavier than previous ones in terms of parameters, its computational complexity does not show a significant increase compared to DCFM [34] and GCoNet+ [45]. Although MCCL [46] has a very lightweight structure, its inference speed is slower than our model. In summary, our model outperforms previous models in most evaluation metrics while maintaining competitive inference efficiency.

Table 4 illustrates the distribution of parameters and computations across the modules of our model. It can be observed that the encoder occupies a substantial portion of the parameters, while the calculation is concentrated in the encoding and decoding stages.

[2]https://github.com/zzhanghub/eval-co-sod
[3]https://github.com/ZhengPeng7/CoSOD_fps_collection. The original benchmark is used on an A100, while we evaluate all models on a 3080Ti to get the results in Table 3.
[4]The batch size is set to 1 for all models when checking FLOPs.

Table 4: Detailed statistics of the full model.

Module	#params	FLOPs
Encoder	86.2M	17.7G
Consensus Extraction	15.8M	0.2G
Consensus Dispersion	42.9M	6.1G
Decoder	11.8M	10.7G
Total	156.7M	34.7G

B ADDITIONAL EXPERIMENTAL RESULTS

The paper has presented the ablation studies on the proposed consensus extraction and dispersion modules. In this section we examine the design of other components in the pipeline. As in the paper, CoCA [44] is used as test set, since it contains the most challenging examples and thus can clearly showcase the impact of different designs. Due to time constraints, each ablation experiment is conducted only once. Although there may be some randomness, the trend of the results is quite evident in most cases.

B.1 Encoder Pre-training Scheme

We first focus on the encoder, and analyze the impact of CLIP pre-training. As shown in Table 5, we compare the default pre-training and fine-tuning scheme with three other settings: training from scratch, using the pre-trained weights without fine-tuning, using the pre-trained weights and fine-tuning it with the learning rate setting to the same value as other modules. It can be clearly seen that the default setting achieves the best performance, which shows the importance of both the priors from large-scale pre-training and the task-specific information from the downstream fine-tuning.

Table 5: Impacts of encoder initialization and training scheme.

Encoder	$E_\xi^{\max} \uparrow$	$S_\alpha \uparrow$	$F_\beta^{\max} \uparrow$
trained from scratch	0.699	0.593	0.387
pre-trained & frozen	0.759	0.645	0.527
pre-trained & fine-tuned (0.1× lr)	**0.804**	**0.707**	**0.598**
pre-trained & fine-tuned	0.742	0.646	0.478

B.2 Number of Layers

Next, we carry out the ablation experiment on the number of ILIAT/ITD layers. Note that the number of GSAT layers should always be the number of ILIAT layers plus one (when #ILIAT>0), so it is not mentioned explicitly here.

The results in Table 6 indicate that the *existence* of these two modules brings about the most significant performance improvement, while the impact of changing their *number* is relatively small. Concretely speaking, adding ITD layers has a greater benefit than adding GSAT and ILIAT layers. It reveals that the naive model requires more improvements in the dispersion stage rather than the extraction stage, which is consistent with the findings of the ablation experiments in the paper. Moreover, adding more layers to the default setting (#ILIAT=2, #ITD=4) will not further improve the performance. This may be due to overfitting caused by the over-complex consensus learning.

Table 6: Impacts of the number of layers.

#ILIAT	#ITD	$E_\xi^{\max} \uparrow$	$S_\alpha \uparrow$	$F_\beta^{\max} \uparrow$	#params
0	0	0.804	0.707	0.598	99.6M
2	2	0.828	0.734	0.623	140.1M
2	4	**0.831**	**0.741**	**0.631**	156.7M
4	2	0.830	0.736	0.623	152.8M
4	4	0.825	0.727	0.615	169.3M

B.3 Loss Functions

Table 7 shows the ablation study on the loss functions. Two loss functions are employed in our model, namely the Binary Cross Entropy (BCE) loss and the Intersection over Union (IoU) loss. As mentioned in [45], IoU loss supervises the model on the region level, while BCE loss helps the model focus on details. Experiments also show that combining these two losses leads to better performance than using either one independently.

Table 7: Impacts of the loss functions.

IoU loss	BCE loss	$E_\xi^{\max} \uparrow$	$S_\alpha \uparrow$	$F_\beta^{\max} \uparrow$
✓		0.820	0.724	0.626
	✓	0.822	0.728	0.613
✓	✓	**0.831**	**0.741**	**0.631**

B.4 Iterative Refinement

As suggested by the reviewer, we carry out an experiment on the effectiveness of the iterative refinement (in the consensus extraction module). The iterative refinement uses multiple GSAT-ILIAT process. It means that the extracted consensus from one GSAT layer is further refined by the following ILIAT layer and GSAT layer, utilizing local details obtained from the hierarchical structure. Table 2 has already proven the effectiveness of GSAT and ILIAT separately. Here we compare the default full model (with 3 GSAT layers and 2 ILIAT layers) against a model with less iterations (i.e. 2 GSAT layers and 1 ILIAT layer). The results are shown in Table 8. It can be concluded that introducing the iterative process in the consensus extraction stage can better refine the consensus representation with local details, thus enhance the overall performance.

Table 8: Impacts of iterative refinement.

#GSAT	#ILIAT	$E_\xi^{\max} \uparrow$	$S_\alpha \uparrow$	$F_\beta^{\max} \uparrow$
2	1	0.828	0.730	0.625
3 (default)	2 (default)	**0.831**	**0.741**	**0.631**