HyperSOR: Context-Aware Graph Hypernetwork for Salient Object Ranking

Minglang Qiao, Mai Xu, *Senior Member, IEEE*, Lai Jiang, *Member, IEEE*, Peng Lei, Shijie Wen, Yunjin Chen, and Leonid Sigal

***Abstract*—Salient object ranking (SOR) aims to segment salient objects in an image and simultaneously predict their saliency rankings, according to the shifted human attention over different objects. The existing SOR approaches mainly focus on object-based attention, e.g., the semantic and appearance of object. However, we find that the scene context plays a vital role in SOR, in which the saliency ranking of the same object varies a lot at different scenes. In this paper, we thus make the first attempt towards explicitly learning scene context for SOR. Specifically, we establish a large-scale SOR dataset of 24,373 images with rich context annotations, i.e., scene graphs, segmentation, and saliency rankings. Inspired by the data analysis on our dataset, we propose a novel graph hypernetwork, named HyperSOR, for context-aware SOR. In HyperSOR, an initial graph module is developed to segment objects and construct an initial graph by considering both geometry and semantic information. Then, a scene graph generation module with multi-path graph attention mechanism is designed to learn semantic relationships among objects based on the initial graph. Finally, a saliency ranking prediction module dynamically adopts the learned scene context through a novel graph hypernetwork, for inferring the saliency rankings. Experimental results show that our HyperSOR can significantly improve the performance of SOR.**

***Index Terms*—Graph neural network, hypernetwork, relative saliency, scene graph, salient object ranking.**

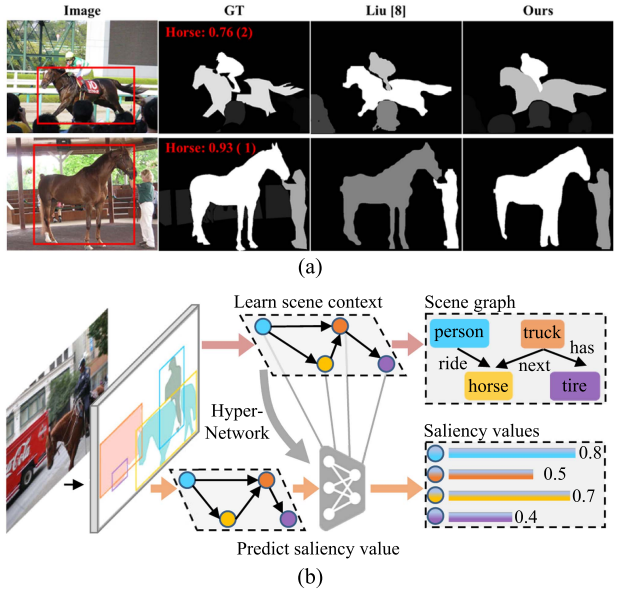


Fig. 1. (a) Two examples of SOR results. (b) Illustration of our graph-based hypernetwork, guiding the SOR task through the scene context learned from scene graph generation.

I. Introduction

SALIENT object detection (SOD) aims to generate a binary saliency map, which locates and segments the object regions attracting human attention in a visual scene. SOD simply assumes that the saliency values of segmented objects are one; however, such an assumption does not hold due to the selective attention of human visual system [1]. To address this issue, salient object ranking (SOR) refers to SOD and instance-wisely prediction on their relative saliency degrees. In contrast to SOD, SOR is capable of predicting a more precise saliency map for practical down-stream tasks of compression [2], image cropping [3], visual tracking [4], and image understanding [5], to name but a few. The work of SOR can be traced back to the end-to-end network [6], which predicts pixel-wise saliency map, and then ranks each object by averaging the saliency values inside the object. The "ranking-by-detection" paradigm is adopted in a set of following SOR works [7], [8], [9], [10], in which the objects are segmented, and then their ranking scores of saliency values are predicted individually as the final results. For example, Fang et al. [9] proposed predicting the scores of object ranking by applying self-attention to learn mutual information between objects. Liu et al. [11] developed a graph neural network (GNN) to model the mutual correlation between objects and then infer their relative saliency rankings.

Several psychological works [12], [13] have indicated that scene context influences human attention; therefore the scene context is closely related with the saliency degrees of objects. Unfortunately, the existing SOR approaches do not consider the rich scene context in a visual scene. As shown in Fig. 1(a),

Manuscript received 2 July 2023; revised 3 January 2024; accepted 18 February 2024. Date of publication 21 February 2024; date of current version 6 August 2024. This work was supported in part by Alibaba Innovative Reaserch, NSFC under Grant 62250001 and Grant 62231002, and in part by Beijing Natural Science Foundation under Grant L223021. Recommended for acceptance by G. Farinella. *(Minglang Qiao and Mai Xu contributed equally to this work.) (Corresponding Author: Lai Jiang.)*

Minglang Qiao, Mai Xu, Lai Jiang, Peng Lei, and Shijie Wen are with the School of Electronic and Information Engineering, Beihang University, Beijing 100191, China (e-mail: minglangqiao@buaa.edu.cn; maixu@buaa.edu.cn; jianglai.china@buaa.edu.cn; buaaray@gmail.com; wenshijie@buaa.edu.cn).

Yunjin Chen is with Alibaba Group, Hangzhou 311121, China (e-mail: chenyunjin_nudt@hotmail.com).

Leonid Sigal is with the Department of Computer Science, University of British Columbia, Vancouver, BC V6Z 3B7, Canada (e-mail: lsigal@cs.ubc.ca).

This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2024.3368158, provided by the authors.

Digital Object Identifier 10.1109/TPAMI.2024.3368158

the horse appears to be less salient than the horseman in the scene of horse riding, in contrast to that of grooming; the existing approach [8] fails to accurately predict the saliency rankings, since it ignores the context relationships between human and horse, i.e., horse riding and grooming. To address this issue, we propose a context-aware graph hypernetwork for SOR (HyperSOR), which learns to capture the scene context via explicitly modeling the objects and their context relationships. As illustrated in Fig. 1(b), a hypernetwork is designed in our approach to guide the task of SOR through dynamic adoption of scene context from the task of scene graph generation. Note that similar to our approach, the recent work of [8] also adopts GNN as the backbone structure, but it can only learn the correlations of object features, rather than the semantic relationships of scene context in our HyperSOR approach. Thus, [8] fails to accurately predict the saliency rankings in the diverse scenarios like Fig. 1(a).

Specifically, we first establish a large-scale SOR dataset, namely saliency ranking of salient object dataset (SalSOD), which includes 24,373 images and 133,338 objects with annotated saliency rankings, bounding boxes, and segmentation masks. We also annotate the scene graphs of objects in our dataset for modeling scene context. Then, we conduct thorough analysis over our dataset and obtain several findings about the factors that are highly related to the saliency rankings, especially from the perspective of scene context. Inspired by the findings, we propose our context-aware HyperSOR approach, which is composed of the initial graph (IG) module, the scene perception graph (SPG) module and the ranking prediction graph (RPG) module. To be more specific, the IG module aims to detect, classify, and segment salient objects, and then construct an initial graph via a geometry-aware relationship proposal block. Given the initial graph, the SPG module is designed to generate the scene graph for learning semantic relationships among objects. In the SPG module, a set of multi-path graph attention blocks are developed to attend and fuse the features of the objects and semantic relationships. Finally, the RPG module is designed to rank salient objects with the multi-level multi-head hyper guidance blocks, which dynamically adopt the learned context from scene graph. Extensive experiments show that our HyperSOR approach achieves state-of-the-art performance over three SOR datasets.

To the best of our knowledge, our HyperSOR approach is the first attempt to learn scene context of visual scenes for the task of SOR. In summary, the contributions of this paper are three-fold:

- We establish a large-scale SOR dataset with the annotations of segmentation mask, saliency value and scene graph. The dataset is public online: https://github.com/MinglangQiao/SalSOD.
- We thoroughly mine our dataset and obtain several findings about the correlations between scene context and object rankings.
- We propose a novel context-aware hypernetwork-based SOR framework, i.e., HyperSOR, which leverages scene context to guide saliency ranking via learning the scene graph in an explicit manner.

II. Related Work

A. Salient Object Detection and Ranking

The early works of SOD [14], [15], [16], [17], [18] are mainly based on hand-crafted features for detecting salient objects, including contrast, color, edge, structure, *etc*. For example, Cheng et al. [14] proposed a histogram-based approach to estimate saliency, via measuring the global contrast of each image region. In [18], a context-aware algorithm was developed to highlight the salient objects with surroundings by considering low-level cues, global considerations, visual organization rules and high-level priors. Most recently, the convolutional neural network (CNN) based approaches have been proposed, significantly boosting the performance of SOD. Particularly, Wu et al. [19] proposed utilizing an extreme down-sampling technique on top of deep CNN model to learn the overall understanding of image, which produces effective high-level features for SOD. Similarly, Liu et al. [20] designed a pooling module to fuse the high-level semantic information with features at different levels, enabling more precise localization and segmentation of salient objects. In [21], Zhang et al. proposed to leverage the high level semantic knowledge of captioning to boost the performance of SOD. In [22], [23], a multi-source weakly supervised approach was developed to leverage the annotations of object categories and captions to detect salient objects. However, it is still challenging to directly apply the existing SOD approaches for the task of SOR, since SOR needs to further distinguish and rank the instance levels of salient objects.

As a further step beyond SOD, Islam et al. [6] introduce the concept of SOR, in which the rankings of the salient objects are obtained according to the agreement across multiple subjects. Accordingly, a hierarchical deep learning model was designed in [6] to predict the pixel-wise saliency map, such that the saliency rankings can be obtained via averaging the saliency values inside each object mask. Similarly, Siris et al. [7] proposed a novel approach by inferring human attention shift order to predict the saliency rankings. However, this approach can only predict the rankings of at most five salient objects. Considering the fact that SOR is sensitive to object position and scale, Fang et al. [9] proposed a multi-task framework with position-preserved attention module to simultaneously perform object segmentation and SOR prediction. In [8], Liu et al. proposed a graph-based approach to learn different saliency cues for SOR, especially the interaction and competition among objects. More recently, Tian et al. [10] have proposed to jointly model the spatial and object-based attention to rank the salient objects. In addition to image SOR, there are a few works intended for video SOR [24], [25]. For example, a fixation prediction model and a SOD model are combined in [24] to generate the rankings of objects in a video. In [25], Lin et al. further proposed a unified model, which explicitly learns the inter-frame spatial information and intra-frame temporal relations for inferring the rankings of objects in video. However, all the above approaches neglect the semantic scene context of objects, leading to inferior performance in the complicated scenes. To solve this issue, we propose a context-aware approach in this paper for the SOR

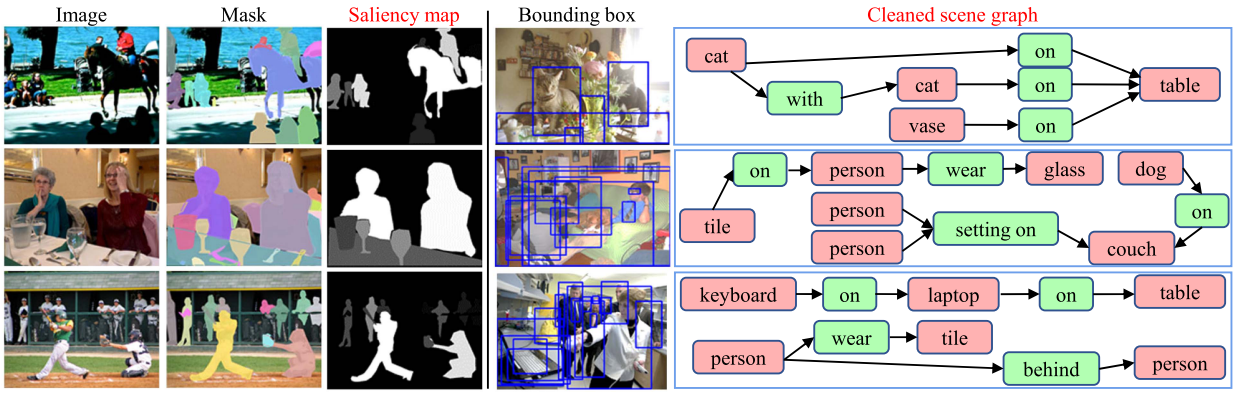


Fig. 2. Some examples of our SalSOD dataset. From left to right: input images, segmentation masks, saliency maps, bounding boxes, and scene graphs.

on images, in which the scene graph is leveraged to boost its performance.

B. Scene Graph

Scene graph is an effective tool for understanding visual context, which depicts both the objects and their semantic relationships in a visual scene. The pioneering work of Johnson et al. [26] first proposed the definition of scene graph, and applied it in semantic image retrieval. After that, there have emerged a great surge of works for scene graph generation [27], [28], [29], [30], [31]. For example, Xu et al. [28] proposed a recurrent neural network (RNN) model to generate scene graph, which iteratively propagates contextual information between objects and relationships along the primal-dual graph structure. In [30], Zellers et al. analyzed the frequently appearing substructures of scene graphs in Visual Genome [32], and proposed the long short-term memory networks (LSTMs) to model the recurring substructures for inferring the categories of objects and predicates. To facilitate the scene graph generation upon the graph theory, some works [29], [31], [33], [34] mainly focus on developing advanced GNN models. For instance, Yang et al. [29] proposed a new relational proposal network and an attention-based graph convolutional network, which can effectively detect objects and capture their relationships in static scenes. Most recently, Li et al. [31] have developed a bipartite GNN for scene graph generation, in which an adaptive message propagation mechanism is designed for encoding scene context.

In addition to the development of architectures for scene graph generation, many works concentrate on applying scene graph in the computer vision tasks, such as image generation [35], [36], image captioning [37], [38], visual question answering [39], [40], and image retrieval [26], [41], *etc*. As verified in these works, scene graph is an effective tool to understand and build up the visual context. For image generation, Johnson et al. [35] proposed an end-to-end generative adversarial network (GAN) based approach to synthetize images directly from a given scene graph. For image captioning, Li et al. [37] designed a framework to yield the caption of images by utilizing the semantic knowledge from the scene graph with a hierarchical attention mechanism. However, to the best of our knowledge, there is no work that explores scene graph in attention related tasks like SOR. In this paper, we make the first attempt to learn the rich semantic information in scene graph for ranking the salient objects.

III. Dataset Establishment

In this section, we establish a new dataset of 24,373 images, named SalSOD, with multiple annotations on the salient objects inside images: 1) semantic segmentation masks and bounding boxes, 2) saliency values and saliency rankings, and 3) scene graphs of objects. Fig. 3 illustrates how to construct our SalSOD dataset from the aspects of image collection, saliency annotation, and scene graph annotation. More details are discussed below.

Image collection: For our SalSOD dataset, we collected images with diverse scenarios from the dataset of COCO [42]. The selected images contain a wide range of scene contexts, including sports, cooking, game playing, driving, and so forth. For each image, the bounding boxes and segmentation masks of objects were also obtained from COCO. To guarantee the contextual diversity, we only collected images that have at least 2 objects. Fig. 2 shows some examples from our SalSOD dataset.

Saliency annotation: Some collected images are the same as those of the SALICON dataset [43], since they are both selected from COCO [42]. Hence, the mouse-contingent gaze data in SALICON can be utilized to obtain the values and ranking scores of saliency for each object, by counting the fixation proportion inside the object. For the rest of images in our SalSOD dataset, we applied the SAM approach [44] to generate the saliency values, instead of using gaze data. Then, the saliency values are multiplied with the object-level segmentation to obtain the saliency maps. However, the above saliency maps suffer from obvious drawbacks, such as missing salient objects, over-valued background, and crowd annotation. Some examples are illustrated in Fig. 2 of the supplemental material. To avoid the above drawbacks, 41 volunteers are asked to subjectively judge whether the saliency maps are reasonable, and then generate the refined saliency map by manually selecting the objects with visually correct saliency. For the images that cannot be refined well enough at the object level, we finally remove them from our dataset. Note that all volunteers are trained and evaluated over

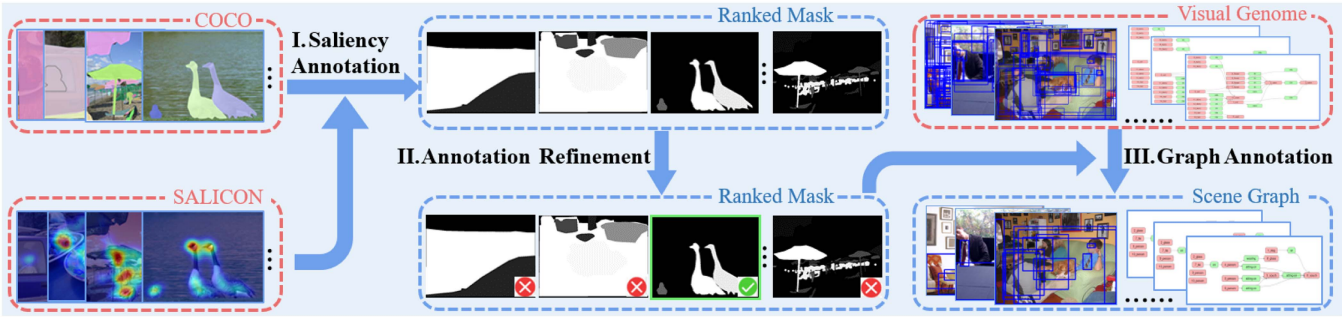


Fig. 3. Procedure of building up our dataset: (I) Saliency annotation by combining the saliency labels from SALICON and masks from COCO, (II) annotation refinement by manually removing bad cases, and (III) graph annotation by cleaning and refining the scene graph from Visual Genome.

a small set of annotations, to ensure they are qualified for the annotation cleaning according to the same standard. Besides, we allow the volunteers to mark some images as "uncertain", and subsequently these images with uncertainty are refined by three senior researchers once again.

Scene graph annotation: We construct the scene graphs on top of the annotations in VG dataset [32]. First, we find out the overlapped images between SalSOD and VG. However, we cannot directly utilize the original scene graph in VG, as the bounding boxes and predicates (i.e., the nodes and edges) in VG are rather dense and noisy, due to the following issues: 1) The number of object categories in VG is much larger than ours (33,877 vs. 80 categories). 2) VG contains over-annotated bounding boxes for SOR task, such as those of non-object regions and tiny objects. 3) The classes of predicates in VG are redundant and there exist different tenses or synonyms of the same predicate. To solve the first issue, we apply WordNet [45] synsets to rename the synonymous object categories in VG into the 80 classes in SalSOD (e.g., "boy", "man" to "person"), and exclude the rest objects out of 80 classes. To solve the second issue, we align the bounding boxes in VG to SalSOD, by picking up the bounding box with the largest intersection over union (IoU) for each object. To solve the third issue, we use the NLTK toolkit [46] to unify the predicates with same meaning (e.g., synonyms, different tenses and singular/plural form), and randomly remove the repeated predicates for the same pair of objects. Finally, the images containing a minimum of two objects and one relationship are reserved for constructing our dataset.

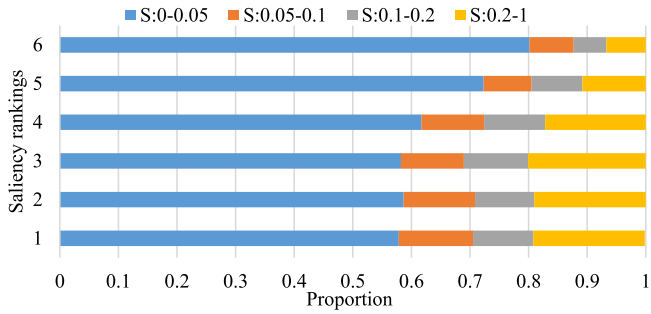
In summary, compared with the existing SOR datasets [7], [8], [47], our SalSOD dataset has the following advantages: 1) *Scene graph annotation.* Our dataset first includes the scene graph annotations, enabling the learning of scene context for SOR task. 2) *Larger scale.* Our dataset significantly enlarges the data scale of SOR datasets, 2.1 times that of the largest existing SOR dataset [47]. 3) *High quality.* A completed annotation refinement scheme is conducted by 41 volunteers, to ensure the annotation quality of our dataset.

IV. Data Analysis

In this section, we thoroughly mine our SalSOD dataset, and obtain the following four findings about the saliency values for the objects in images.

Metric	PLCC	SROCC	KROCC
Object scale	0.71	0.78	0.59
Random	-5.74×10^{-5}	-3.18×10^{-5}	-1.53×10^{-4}

(a) Correlation values.



(b) Proportions of objects with different scales.

Fig. 4. (a) Correlation between object scales and saliency values. (b) The proportions of objects with different scales under 6 saliency rankings. Note that "S" indicates the normalized object scale in image.

Finding 1: In an image, the saliency values of objects are closely correlated with their scales and positions.

Analysis: Here we explore the correlation between the saliency values and the low-level attributes of the objects, i.e., the object scales and locations. For all objects in our SalSOD dataset, we calculate the Pearson linear correlation coefficients (PLCC), Kendall rank correlation coefficient (KROCC) and Spearman rank-order correlation coefficient (SROCC) between their saliency values and object scales. As shown in Fig. 4(a), compared with a random baseline, the PLCC, SROCC, and KROCC between scales and saliency values are significantly higher, with the values of 0.71, 0.78 and 0.59, respectively. In Fig. 4(b), we further calculate the proportions of the object scale for each saliency ranking. As shown, the more salient objects with lower ranking trend to be with larger scales. The above results imply that the objects with larger scales are more likely to have higher saliency values. Besides, we explore the relationship between saliency values of objects and their locations in images. To this end, the averaged saliency values are calculated in Fig. 5, according to the locations of all objects in SalSOD. As shown, the objects that are close to the image center tend to rank higher in terms of saliency. This phenomenon is similar to the center-bias of eye fixations in images, as investigated in [48], [49]. Finally, the analysis of *Finding 1* is completed.

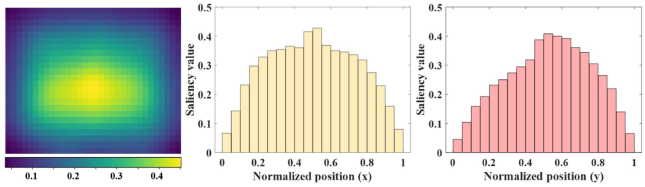


Fig. 5. Averaged saliency values of objects along the horizontal and vertical direction of image. The positions of objects are normalized into [0, 1], instead of the raw pixel values in image.

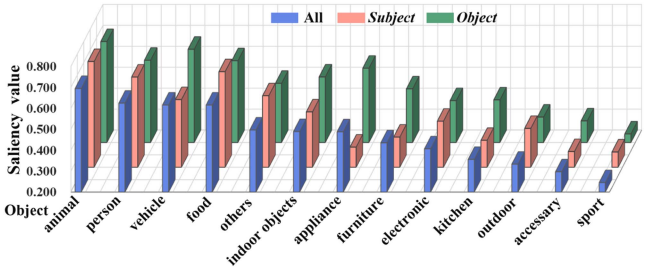


Fig. 6. Averaged saliency values of different object categories (the blue bars). Note that "*subject*" and "*object*" indicate the averaged values when the objects act as *subject* and *object* in the scene graph, respectively.

Finding 2: In an image, the saliency values of objects are influenced by their categories, and they vary when the roles of objects differ.

Analysis: We first investigate whether the saliency values of objects are correlated with their categories. To this end, we calculate the averaged saliency values for each category of objects in our SalSOD dataset. As illustrated in Fig. 6, the saliency values vary across object categories. In particular, the average saliency values of animal, person and vehicle are 0.694, 0.624 and 0.616, respectively, while those of accessory and sporting goods are 0.297 and 0.245, respectively. Therefore, the saliency values of objects are influenced by the corresponding categories, consistent with the results of [8].

There are two roles for the objects in the images, i.e., *subject* and *object*, which can be modeled by the context relationship $\langle subject \rightarrow predicate \rightarrow object \rangle$. See Fig. 2 for an example. Fig. 6 shows the averaged saliency values for each category of objects acting as *subject* and *object*, respectively. We can see that there exists difference of saliency values between the roles of *subject* and *object*, for each object category. Hence, the saliency values vary at different roles for the objects. This completes the analysis of *Finding 2*.

Finding 3: The saliency values of objects are influenced by their relationships with other objects.

Analysis: We use the object-predicate pairs to model the relationships between object instances. Then, we calculate the averaged saliency values for all object-predicate pairs over our SalSOD dataset. Fig. 7(a) illustrates the averaged saliency values along with the object-predicate pairs of four object categories of "Electronic", "Kitchenware", "Furniture", and "Vehicle". We can observe from this figure that the saliency values of objects are influenced by the predicates, i.e., relationships with other objects. Similarly, Fig. 7(b) shows the averaged saliency values of the object-predicate pairs, based on the particular predicates (e.g.,"Be", "Along", "In" and "Have"). The above results imply that the influence of relationships on saliency values varies at different predicates for each object category. Finally, this analysis is completed.

Finding 4: In an image, the objects are likely to have higher saliency values, when semantically related to more objects.

Analysis: Intuitively, in an image, people tend to pay attention on the object with more relationships to other objects. For example, the microwave in Fig. 8(a) has a higher saliency value than the rest objects, probably because it is semantically related to almost all objects in the image, i.e., multiple objects are in or on the microwave. To verify this, we utilize the scene graph to model the semantic relationships between the objects. In the scene graph, the nodes denote the objects, and the edges represent the relationships between these objects. Then, we count the number of edges for each node, denoted as the node degree. We further count the in-degree and out-degree of each node, which refer to the number of edges coming into and going out the node, respectively. Fig. 8(b) shows the averaged saliency values along with the increased node degree, over our SalSOD dataset. We can see from this figure that the averaged saliency value is increased at a larger node degree. This implies that the objects in an image are likely to have higher saliency values, when owing the semantic relationships with more objects. Finally, the analysis of *Finding 4* is completed.

V. THE PROPOSED APPROACH

In light of the above findings in Section IV, we propose the context-aware HyperSOR approach for SOR, and its framework is shown in Fig. 9. As seen in this figure, the framework of our context-aware HyperSOR approach is comprised by the IG module, the SPG module, and the RPG module. Note that we build both the SPG and RPG modules on top of graph neural network, which is effective in modeling relationships and propagating semantic features, and thus it is appropriate for our tasks of scene graph generation and SOR. Specifically, the input image is first fed into the IG module to detect, classify and segment its objects. Meanwhile, a geometry-aware relationship proposal block is developed in the IG module, to construct an initial graph for the following SPG module and RPG module, by considering the geometry characteristics of the detected objects. Subsequently, the initial graph is input to the SPG module for scene graph generation, which includes a set of multi-path graph attention blocks, to learn the semantic relationships between objects. This way, the scene context of the input image can be captured to guide the learning of SOR. Parallel to the SPG module, the initial graph from the IG module is also input to the RPG module for predicting the saliency scores of the objects. Particularly, the multi-level multi-head hyper guidance blocks are developed to transfer the captured scene context from the SPG module to guide graph reasoning in the RPG module. The hyper mechanism in the RPG module is capable of improving generalization ability of the proposed approach when facing different visual scenes. Finally, the predicted saliency scores are combined with the corresponding segmentation masks to yield

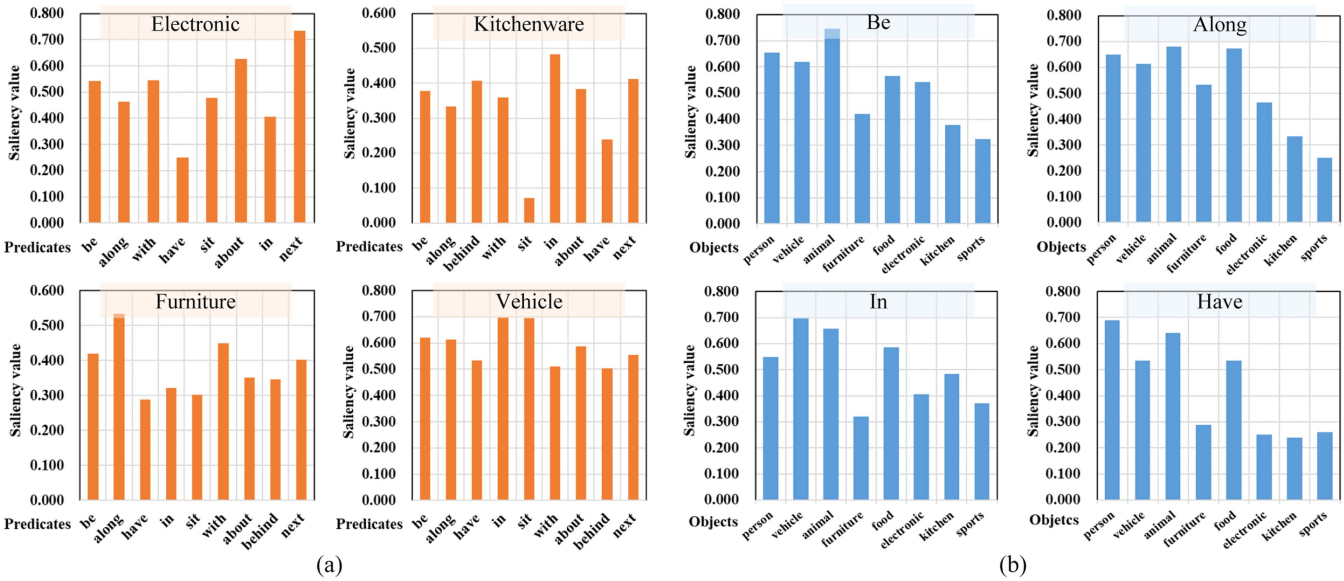
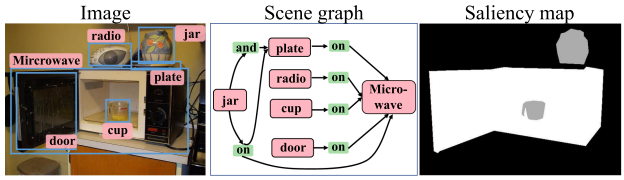


Fig. 7. Averaged saliency values of object-predicate pairs over four object categories (a) and four predicate categories (b).



(a) Example of the scene graph and saliency map for one image.

Degree	0	1	2	3	4+
Node degree	0.24	0.47	0.54	0.59	0.66
In-degree	0.29	0.47	0.49	0.51	0.59
Out-degree	0.28	0.50	0.63	0.69	0.73

(b) Saliency values of objects.

Fig. 8. (a) Example to show the correlation between node degrees of objects in scene graph and saliency values. (b) The averaged saliency values of objects under different number of node degrees.

TABLE I
SUMMARY OF SOME KEY NOTATIONS IN THIS PAPER

Notation	Description
$\mathbf{O}_i$	The feature map of detected object
$\mathbf{b}_i$	The bounding box of detected object
$\mathbf{c}_i$	The category of detected object
$\mathbf{M}_i$	The segmentation mask of detected object
$\tilde{s}_{i,j}$ / $s_{i,j}$	The predicted / ground-truth edge confidence score
$\mathbf{v}_i^o$ / $\mathbf{v}_{i,j}^p$	The initial feature vector of object / predicate node
$\hat{\mathbf{v}}_i^o$ / $\hat{\mathbf{v}}_{i,j}^p$	The updated $\mathbf{v}_i^o$ / $\mathbf{v}_{i,j}^p$ in the SPG module
y_i^o / $y_{i,j}^p$	The predicted category of object / predicate node
$\tilde{y}_i^o$ / $\tilde{y}_{i,j}^p$	The ground-truth category of object / predicate node
$\mathbf{u}_i^o$ / $\mathbf{u}_{i,j}^p$	The input object / predicate feature of the HG block
$\hat{\mathbf{u}}_i^o$ / $\hat{\mathbf{u}}_{i,j}^p$	The output object / predicate feature of the HG block
r_i / $\hat{r_i}$	The predicted / ground-truth saliency values of object

Note that i and j are the indexes of objects; o and p indicate object and predicate, respectively.

the saliency ranking map. More details about the IG, SPG and RPG modules are introduced in the following. Table I lists the main notations in this paper.

A. Initial Graph (IG) Module

In our HyperSOR approach, the initial graph (IG) module is built for both object detection and graph initialization. The structure of the IG module is shown in Fig. 9, and more details are discussed as follows.

Object detection: For object detection, a ResNet-101 [50] network is applied to extract multi-level convolutional features from the input image. Then, the feature pyramid network (FPN) [51] followed by a region proposal network [52] is employed to merge multi-scale features at different levels, and generate a set of object proposals as well as the corresponding feature maps $\{\mathbf{O}_i\}_{i=1}^n$, in which n is the number of objects. Finally, the feature maps are fed into the box, class, and mask heads, to inference the bounding boxes $\{\mathbf{b}_i\}_{i=1}^n$, the object categories $\{\mathbf{c}_i\}_{i=1}^n$ and the segmentation masks $\{\mathbf{M}_i\}_{i=1}^n$, respectively.

Graph initialization: Given the detection results, the initial graph is constructed with the representation of both objects and relationships between objects. Different from the graphs in existing SOR or SOD approaches [8], [53], [54] that are constructed only by object nodes, the graph in our approach further includes predicate nodes for better learning scene context information. Specifically, the initial graph is composed of object and predicate nodes. We firstly initialize the graph as a fully connected direct graph to comprehensively learn the relationships between different nodes. For graph initialization, each node in the graph is represented by an initial feature vector. To be more specific, $\mathbf{v}_i^o$ denotes the initial vector of the i-th

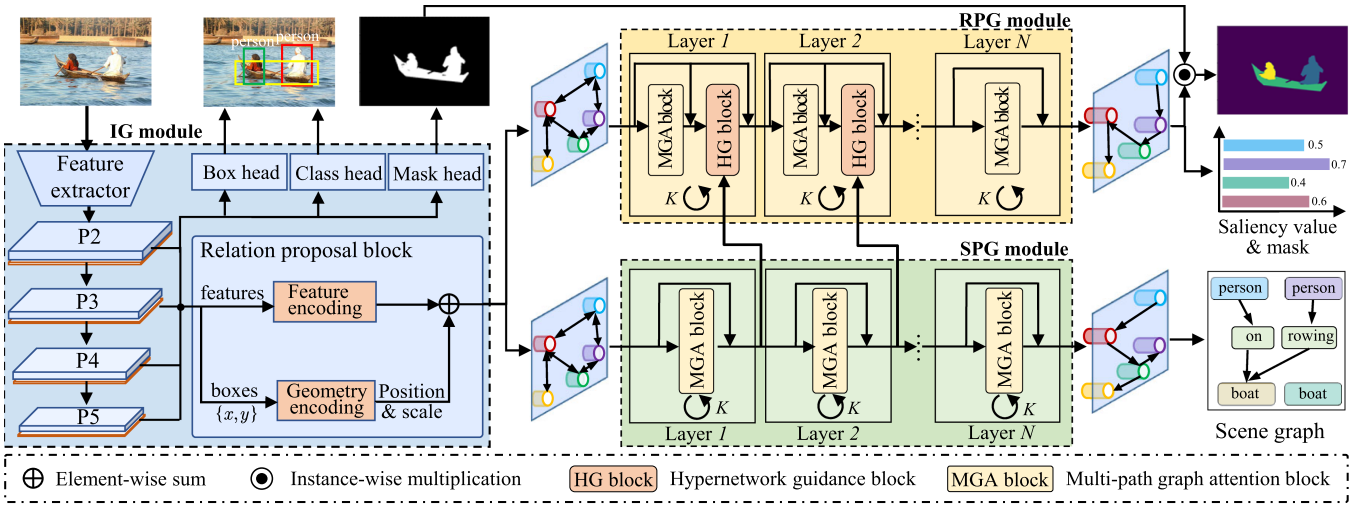


Fig. 9. Overall framework of our HyperSOR approach, including the IG, SPG, and RPG modules. The IG module constructs the initial graph by considering both the object geometries and semantics over the detected objects. Subsequently, the SPG module learns the context information between the objects via scene graph generation. Finally, the RPG module predicts the saliency ranking scores with the guidance of the SPG module, based on the MGA and HG blocks.

object node, while $\mathbf{v}_{i,j}^{p}$ is the initial vector of the predicate node between i-th and j-th object nodes. As introduced in *Finding 1*, the geometry information, especially the position and scale, is closely related with the saliency values of objects. Hence, both position and scale information of objects are embedded in our graph initialization. Specifically, the initial representations for object node $\mathbf{v}_{i}^{o}$, and predicate node $\mathbf{v}_{i,j}^{p}$ can be formulated as

$$\begin{aligned}\mathbf{v}_{i}^{o} &= \text{Conv}\left(\mathbf{O}_{i}\right) + \text{MLP}\left(\mathbf{b}_{i}\right), \\ \mathbf{v}_{i,j}^{p} &= \text{Conv}\left(u\left(\mathbf{O}_{i}, \mathbf{O}_{j}\right)\right) + \text{MLP}\left(\mathbf{b}_{i}\right) + \text{MLP}\left(\mathbf{b}_{j}\right). \end{aligned} \quad (1)$$

In the above equation, $\mathbf{O}_{i}$ and $\mathbf{O}_{j}$ are the feature maps of the i-th and j-th objects, while $\mathbf{b}_{i}$ and $\mathbf{b}_{j}$ are their bounding boxes. In addition, $\text{Conv}(\cdot)$ is the convolutional block to project the feature map into a high-dimensional feature vector, and $u(\cdot)$ is the union function to obtain union region of two bounding boxes. Specifically, $\text{Conv}(\cdot)$ consists of five 2D convolutional layers with kernel size of 3×3 and 1×1, which takes the 256-dimensional object feature map as input. Besides, the learnable multi-layer perceptron (MLP), denoted as $\text{MLP}(\cdot)$, is conducted to encode the position and scale information of the object bounding box.

Similar to the object proposal in Mask R-CNN [55], our IG module may propose an abundance of edge candidates for input image. Therefore, we estimate the confidence score of each edge to filter out the unnecessary edges and preserve the significant ones for our initial graph. Let $s_{i,j}$ denote the confidence scores of the two edges in the relationship triplet $\langle \mathbf{v}_{i}^{o} \rightarrow \mathbf{v}_{i,j}^{p} \rightarrow \mathbf{v}_{j}^{o} \rangle$. Then, $s_{i,j}$ can be calculated by considering both object semantic and geometry scores as follows,

$$s_{i,j} = \sigma\big(\underbrace{\text{MLP}(\mathbf{c}_{i}) \odot \text{MLP}(\mathbf{c}_{j})}_{\text{semantic score}} + \underbrace{\text{MLP}\big(g_{p}(\mathbf{b}_{i}, \mathbf{b}_{j})\big)}_{\text{geometry score}}\big). \quad (2)$$

where $\mathbf{c}_{i}$ and $\mathbf{c}_{j}$ are the probability of category for object i and j, respectively. Besides, $\sigma(\cdot)$ is the logistic sigmoid function, and $\odot$ indicates dot product. Moreover, $g_{p}(\cdot)$ denotes the projection function that maps the bounding boxes $\mathbf{b}_{i}$ and $\mathbf{b}_{j}$ into a high-dimensional geometry vector, and in this paper the geometry encoding approach [56] is used to model $g_{p}(\cdot)$. Note that we follow [29] to obtain semantic score via MLPs and dot product. The geometry score is directly computed from the input space, similar to [56]. After obtaining the confidence score $s_{i,j}$ for all edges, the edges with top-K scores are preserved as the initial edges. Finally, the initial graph is constructed with the initial node representations and preserved edges, and then fed into the SPG module and the RPG module for scene graph generation and ranking prediction.

B. Scene Perception Graph (SPG) Module

In this section, the SPG module is developed to learn the scene context and then generate scene graph based on the initial graph yielded by the IG module. This is motivated by our analysis that the saliency values of objects are highly related with the scene context, which can be learned by scene graph generation. Therefore, we propose to leverage the learned knowledge during scene graph generation for guiding the inferring of salient object rankings. As shown in Fig. 9, the SPG module includes N-layers of the multi-path graph attention (MGA) blocks, which are developed to update the features of each node on the top of the initial graph. For each layer of the SPG module, it contains a MGA block with a residual connection, and the MGA block is recurred with K iterations. Finally, the last layer of the SPG module generates the scene graph, including the predicted categories of each object y_{i}^{o} and predicate $y_{i,j}^{p}$. Moreover, the output features of each layer in the SPG module are utilized to guide the ranking prediction of the RPG module as described in Section V-C.

MGA block: As introduced above, the graph in our approach contains two sets of nodes, i.e., object nodes and predicate nodes. In specific, the object nodes have three types of relations connected to the neighboring nodes, i.e., object $\rightarrow$*predicate*,

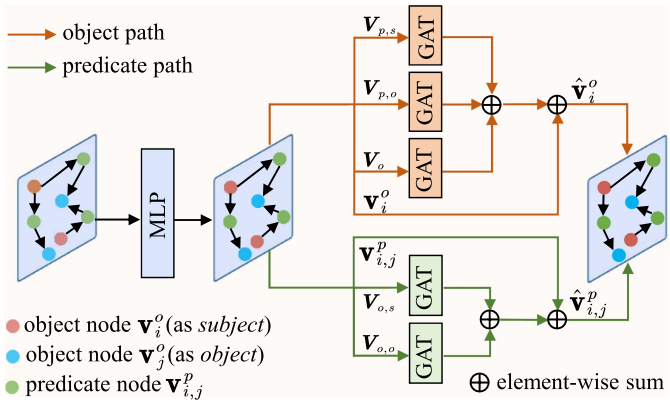


Fig. 10. Proposed multi-path graph attention block. Considering the semantic discrepancy of objects and predicates, we update the object (orange lines) and predicate (green lines) nodes separately, based on the graph attention network (GAT).

predicate→ object, and object → object. Similarly, the predicate nodes contains two types of connection relations for *subject*→predicate and predicate→*object*, respectively. In order to individually learn the representations of the object and predicate, we develop two separate paths in the MGA block for updating the features of object and predicate nodes, respectively. In the MGA block, the features for each object node $\mathbf{v}_i^o$ and each predicate node $\mathbf{v}_{i,j}^p$ are firstly transformed by a learnable MLP, and then aggregated and updated via a two-path structure as illustrated in Fig. 10.

In the object path of the MGA block, three types of neighboring nodes are individually aggregated to update each object node. Specifically, taking the i-th object node as the target node, the neighboring nodes include:

1) The neighboring predicate nodes of the target node with the target node acting as $subject$ in the triplet: $\langle$*subject*→*predicate*→*object*$\rangle$. For these nodes, we denote the set of features as $\mathbf{V}_{p,s} = \{\mathbf{v}_{i,j}^p | j \in \mathcal{N}(i), \text{target node is } subject\}$.
2) The neighboring predicate nodes of the target node with the target node acting as $object$ in the triplet: $\langle$*subject*→*predicate*→*object*$\rangle$. For these nodes, we denote the set of features as $\mathbf{V}_{p,o} = \{\mathbf{v}_{i,j}^p | j \in \mathcal{N}(i), \text{target node is } object\}$.
3) Other object nodes. For these nodes, we denote the set of features as $\mathbf{V}_o = \{\mathbf{v}_j^o | j \in \mathcal{N}(i)\}$. Note that we follow [29] to connect all object nodes when aggregating to capture the underlying relationships between them.

Note that the neighboring predicate nodes and the other object nodes constitute the one-hop and two-hop neighborhoods, respectively. This motivates the multi-iteration and multi-layer designs in Fig. 9. Given the above features, the feature of the i-th object node $\mathbf{v}_i^o$ can be updated as follows,

$$\hat{\mathbf{v}}_i^o = \sigma\left(\mathbf{v}_i^o + \mathrm{A}_{p,s}\left(\mathbf{V}_{p,s}\right) + \mathrm{A}_{p,o}\left(\mathbf{V}_{p,o}\right) + \mathrm{A}_o\left(\mathbf{V}_o\right)\right), \quad (3)$$

where $\mathrm{A}_{p,s}(\cdot)$, $\mathrm{A}_{p,o}(\cdot)$ and $\mathrm{A}_o(\cdot)$ are three GATs [57] with different parameters. Benefiting from the attentional structure of GAT, the scene context can be effectively learned for each object node. Our choice of GAT is based on two reasons: (1) GAT can learn the importance of different neighborhoods and aggregate them in an attention manner, and therefore is effective for the *inductive* tasks of scene graph generation and salient object ranking. (2) GAT is computationally efficient for feature aggregation, as it can be parallelized across edges and heads, via its self-attention and parameters-sharing schemes.

In the predicate path of the MGA block, each predicate is updated by aggregating its neighboring object nodes, i.e., the object node acts as $subject$ and the object node acts as $object$. Take the predicate node between i-th and j-th object nodes as an example. Recall that the relationship triplet of these nodes can be denoted as $\langle \mathbf{v}_i^o \rightarrow \mathbf{v}_{i,j}^p \rightarrow \mathbf{v}_j^o \rangle$, in which $\mathbf{v}_{i,j}^p$, $\mathbf{v}_i^o$, and $\mathbf{v}_j^o$ are the features of the corresponding nodes. Then, the feature of the predicate node $\mathbf{v}_{i,j}^p$ can be updated as

$$\hat{\mathbf{v}}_{i,j}^p = \sigma\left(\mathbf{v}_{i,j}^p + \mathrm{A}_{o,s}\left(\mathbf{v}_i^o\right) + \mathrm{A}_{o,o}\left(\mathbf{v}_j^o\right)\right). \quad (4)$$

In the above equation, $\mathrm{A}_{o,s}$ and $\mathrm{A}_{o,o}$ are two different GATs, and $\hat{\mathbf{v}}_{i,j}^p$ is the updated feature for $\mathbf{v}_{i,j}^p$.

C. Rank Prediction Graph (RPG) Module

The RPG module is developed to predict the saliency rankings, on the basis of the initial graph from the IG module. As shown in Fig. 9, similar to the SPG module, the RPG module is also built on a N-layer architecture, in which each layer contains an MGA block and a developed hypernetwork guidance (HG) block. Specifically, the MGA block is designed to update the graph features (as introduced in Section V-B), while the HG block is developed to adopt the learned scene context information from the SPG module. Note that both MGA and HG blocks at each layer are recurred for K iterations, and the residual connection is applied to alleviate the gradient vanishing issue (see Fig. 11(a)). Finally, the last layer of the RPG model outputs the saliency values $\{r_i\}_{i=1}^{N_o}$ of the N_o objects, and they are combined with the segmentation masks of corresponding objects to generate the final saliency map for SOR. The detailed structure about the HG block is introduced as follows.

HG block: As revealed in Section IV, even for the same type of object, its saliency values can vary significantly with changes in the scene context. Therefore, for our RPG module, the way to infer saliency rankings of objects should vary according to the scene context. Motivated by this, a graph hyper mechanism is developed in the HG block, which dynamically guides the feature update by adopting the learned scene context from the SPG module. As illustrated in Fig. 11(b), our HG block has a multi-head structure with a hyper mechanism. To be more specific, given an input image, the parameters of the HG block are adjusted according to the learned scene context of the image. Take the i-th object node in the input initial graph as an example. For each HG block, $\mathbf{u}_i^o$ and $\hat{\mathbf{u}}_i^o$ denote the input and output features of the i-th object node, respectively. Besides, as presented in (3), $\hat{\mathbf{v}}_i^o$ is the updated object feature from the corresponding MGA block in the SPG module. Mathematically, the output feature $\hat{\mathbf{u}}_i^o$ can be obtained through a dynamic fully

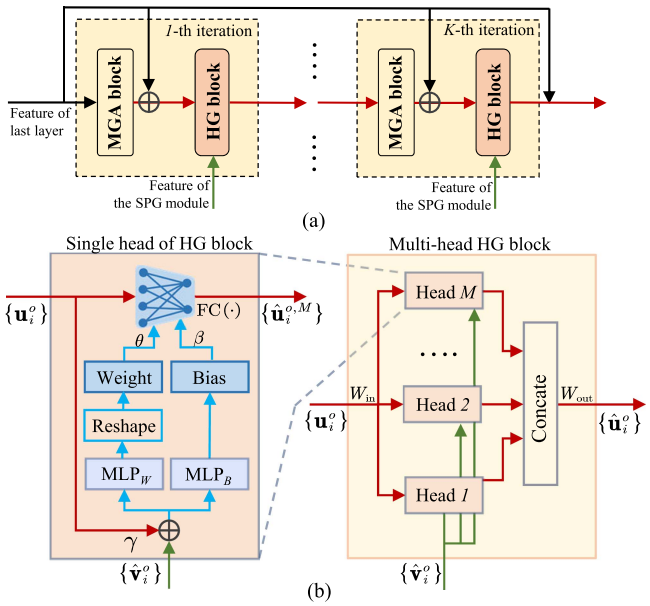


Fig. 11. Illustration of the proposed HG block in each layer. (a) The recursion and residual structure of MGA and HG blocks with K iterations. (b) The multi-head HG block (left part) with the detailed single-head structure (right part). Note that the HG block is developed in a hyper mechanism, conditioned on the context-aware features from the SPG module.

connected layer FC$(\cdot)$ conditioned on $\hat{\mathbf{v}}_i^o$,

$$\begin{aligned} \hat{\mathbf{u}}_i^o &= \mathrm{FC}(\mathbf{u}_i^o \mathbf{W}_{\text{in}}; \theta, \beta), \\ \text{where } \theta &= \mathrm{MLP}_W(\hat{\mathbf{v}}_i^o + \gamma \cdot \mathbf{u}_i^o), \\ \beta &= \mathrm{MLP}_B(\hat{\mathbf{v}}_i^o + \gamma \cdot \mathbf{u}_i^o). \end{aligned} \tag{5}$$

In the above equation, $\mathbf{W}_{\text{in}}$ is a learnable matrix to reduce the dimension of input feature, such that the computation complexity can be reduced. Moreover, θ and β are the learnable weights and biases of FC$(\cdot)$, generated by two MLPs, i.e., MLP$_W(\cdot)$ and MLP$_B(\cdot)$. As shown in Fig. 11(b), the input of the MLP consists of both $\mathbf{u}_i^o$ and $\hat{\mathbf{v}}_i^o$, with a hyper-parameter of weight γ. This way, the feature update of FC$(\cdot)$ in the RPG module can be parameterized by the learned scene context in $\hat{\mathbf{v}}_i^o$ from the SPG module.

In order to infer the saliency ranking at diverse scenarios, the multi-head structure of the HG block is developed, which jointly guides the feature update with several parallel heads through the hyper mechanism. Note that similar multi-head mechanism has been verified to be effective for enhancing model capacity in the existing works [11], [57]. Assume that there are M heads in the HG block. The structure of the M-head HG block is shown in Fig. 11(b), and it can be formulated as follows,

$$\begin{aligned} \hat{\mathbf{u}}_i^o &= \mathrm{Concat}\left(\hat{\mathbf{u}}_i^{o,1}, \ldots, \hat{\mathbf{u}}_i^{o,M}\right) \mathbf{W}_{\text{out}}, \\ \text{where } \hat{\mathbf{u}}_i^{o,m} &= \mathrm{FC}(\mathbf{u}_i^o \mathbf{W}_{\text{in}}; \theta^m, \beta^m),\ m = 1, \ldots, M, \end{aligned} \tag{6}$$

where $\mathbf{W}_{\text{out}}$ is a learnable matrix with the linear transform to resume the dimension of the output feature. Besides, Concat$(\cdot)$ denotes the concatenation operation. In the HG block, the feature of each predicate node $\mathbf{u}_{i,j}^p$ is also updated to $\hat{\mathbf{u}}_{i,j}^p$, the same as that of object node $\mathbf{u}_i^o$.

D. Loss Function

In this section, we mainly focus on the loss functions for our HyperSOR approach, including the loss functions of the IG, SPG and RPG modules. The loss functions for the individual modules are presented as follows.

Loss for the IG module: For the IG module, the predicted edge confidence scores in the generated initial graph, i.e., the $s_{i,j}$ in (2), are supervised by the relationships in the ground-truth scene graph. Let N_o denote the number of objects in an image. Then, a binary cross entropy loss is defined,

$$\mathcal{L}_{\text{IG}} = \sum_{i=1}^{N_o} \sum_{j=1}^{N_o} \{\tilde{s}_{i,j} \cdot \log(s_{i,j}) + (1 - \tilde{s}_{i,j}) \cdot \log(1 - s_{i,j})\}. \tag{7}$$

In the above equation, $\tilde{s}_{i,j}$ is the binary ground-truth label for $s_{i,j}$, and it is set to 1 when the edge exists, otherwise is set to 0.

Loss for the SPG module: For the SPG module, the multi-class cross entropy loss is applied for both the predicted objects and predicates. Recall that y_i^o and $y_{i,j}^p$ are the predicted categories of object and predicate from the SPG module. Mathematically, the loss function for the SPG module can be formulated as,

$$\mathcal{L}_{\text{SPG}} = \sum_{\tilde{y}_i^o \in \tilde{\mathcal{Y}}_o} \tilde{y}_i^o \cdot \log(y_i^o) + \sum_{\tilde{y}_{i,j}^p \in \tilde{\mathcal{Y}}_p} \tilde{y}_{i,j}^p \cdot \log\left(y_{i,j}^p\right), \tag{8}$$

where $\tilde{y}_i^o \in \tilde{\mathcal{Y}}_o$ and $\tilde{y}_{i,j}^p \in \tilde{\mathcal{Y}}_p$ are the ground-truth categories for y_i^o and $y_{i,j}^p$, respectively.

Loss for the RPG module: As discussed in Section V-C, $\{r_i\}_{i=1}^{N_o}$ are the predicted saliency rankings of objects from our RPG module, which are supervised by the corresponding ground-truth $\{\tilde{r}_i\}_{i=1}^{N_o}$. To comprehensively supervise the SOR prediction, we incorporate a ranking loss and a mean square error (MSE) loss for the RPG module. The ranking loss prompts predictions with ground-truth saliency rankings, and the MSE Loss encourage predictions to closely match ground-truth saliency values. Consequently, predictions are compelled to be closer to ground-truth, in terms of both saliency rankings and values. Inspired by [8], the ranking loss $\mathcal{L}_{\text{RANK}}$ is developed for each object pair, as formulated by

$$\begin{aligned} \mathcal{L}_{\text{RANK}} &= \sum_{i=1}^{N_o} \sum_{j=1}^{N_o} w_{i,j} \mathrm{R}(r_i, r_j), \\ \text{where } \mathrm{R}(r_i, r_j) &= \begin{cases} \log(1 + \exp(-r_i + r_j)), & \text{if } \tilde{r}_i > \tilde{r}_j \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{9}$$

Here, $w_{i,j}$ is an importance weight assigned to each pair of objects according to their ranking difference. In addition, the MSE loss can be formulated as

$$\mathcal{L}_{\text{MSE}} = \sum_{i=1}^{N_o} \|\tilde{r}_i - r_i\|^2. \tag{10}$$

Finally, we combine the ranking loss and MSE loss as the loss function of the RPG module:

$$\mathcal{L}_{\text{RPG}} = \lambda_{\text{R}} \mathcal{L}_{\text{RANK}} + \lambda_{\text{M}} \mathcal{L}_{\text{MSE}}, \tag{11}$$

where λ_{R} and λ_{M} are the balance weights for each loss.

Total loss: The total loss function $\mathcal{L}$ for training our HyperSOR model is a weighted sum of $\mathcal{L}_{\mathrm{IG}}$, $\mathcal{L}_{\mathrm{SPG}}$ and $\mathcal{L}_{\mathrm{RPG}}$,

$$\mathcal{L} = \mathcal{L}_{\mathrm{RPG}} + \lambda_{\mathrm{IG}}\mathcal{L}_{\mathrm{IG}} + \lambda_{\mathrm{SPG}}\mathcal{L}_{\mathrm{SPG}}, \tag{12}$$

where λ_{IG} and λ_{SPG} are the hyper-parameters for balancing the different types of losses.

VI. Experiments and Results

In this section, we conduct extensive experiments to validate the effectiveness of the proposed HyperSOR approach. In Section VI-A, we introduce the experimental settings, including dataset, implementation details, and metrics. In Section VI-B, we present the quantitative and qualitative results of the proposed HyperSOR approach for salient object ranking, in comparison with 11 other state-of-the-art approaches. In Section VI-C, we discuss the results of our ablation studies to investigate the contribution of each component in our HyperSOR approach. Finally, we evaluate the scene graph generation performance of our approach in Section VI-D.

A. Experimental Settings

Datasets: For the task of scene graph generation, the VG150 [28] dataset is introduced for training and evaluation of the IG and SPG modules. For the task of SOR, the images in our SalSOD are divided into training and test sets in a ratio of $2:1$, according to the data split of SALICON. Additionally, for evaluating the generalization ability of HyperSOR, other two publicly available SOR datasets are adopted to test the trained model, including PASCAL-S [6] and Siris' dataset [7]. Note that, similar to our SalSOD dataset, Siris' dataset is generated based on the SALICON and COCO, which have been noted in Section III.

Implementation details: For network training, the backbone structures in the IG module (i.e., ResNet-101 [50] and FPN [51]) are initialized with their pre-trained models on the COCO dataset, while the rest of structures in our HyperSOR are initialized with Xavier uniform [58]. The layer number N is set to 2 in both the SPG and the RPG modules, and the head number is set to 2 in each MGA/HG block. Following [29], the number of initial edges K, is empirically set by 256 to preserve edges capturing significant relationships during graph initialization. To minimize the loss function in (12), the Adam [59] optimizer with initial learning rate of 5×10^{-3} is adopted for optimization, and the batch size is set to 8. Other key hyper-parameters for model training are listed in Table II. Note that the above hyper-parameters are tuned over the training set for our experiments. We train and test our model on a server with one CPU (Intel Platinum 8163 CPU@2.50 GHz), 256 GB RAM, and 8 NVIDIA Tesla V100 GPUs with 32 GB of memory. In practical, our and compared approaches are trained and tested on a single GPU.

Evaluation metrics: For evaluating the SOR performance, we adopt 3 existing metrics, i.e., siris' salient object ranking (SSOR) [7], segmentation-aware SOR (SA-SOR) [8] and mean absolute error (MAE). Besides, a new metric named as symmetrical salient object ranking (SYSOR) is proposed by considering both under-prediction and over-prediction situations. Note that the larger value of SYSOR/SSOR/SA-SOR and smaller value of MAE indicate the higher prediction accuracy for SOR. Here, we briefly introduce the basic ideas of the existing metrics.

- SSOR: SSOR first matches each ground-truth salient object among all predicted objects according to their IoU of masks. If a ground-truth salient object fails to be matched over predictions, this ground-truth object is simply dropped without penalization. Finally, the correlation coefficient (CC) is calculated between the saliency rankings of predicted and ground-truth objects. As a result, SSOR can not correctly evaluate the SOR results when under-prediction and over-prediction.
- SA-SOR: Different from SSOR, SA-SOR strictly matches each ground-truth salient object among all predicted objects in a one-to-one manner. If a ground-truth salient object is not matched, the corresponding predicted saliency ranking is assigned as 0. At last, the CC is calculated between the saliency rankings of predicted and ground-truth objects. However, SA-SOR still fails to correctly evaluate the SOR results in cases of over-prediction, due to its ignorance of redundant objects in the predictions.
- MAE: Unlike SSOR and SA-SOR, MAE measures the absolute pixel-wise distance between the ground-truths and predicted saliency maps. MAE roughly evaluates the SOR results at the map level, but ignoring the instance level difference.

In Fig. 12, we provide some examples of SSOR and SA-SOR. As shown, both SSOR and SA-SOR make incorrect evaluation when an SOR saliency map over-predicts salient objects than the ground-truth. For instance, the values of SSOR and SA-SOR remain unchanged as 1 despite the presence of redundant objects altering the overall predictions. To address this issue, we propose a new metric of SYSOR, which computes correlations between predictions and ground-truth based on the forward and reverse matching results. Therefore, SYSOR penalizes both missing and redundant objects in predictions, and achieves a more comprehensive evaluation. Specifically, given each ground-truth salient object, we strictly match it among all predicted salient objects according to their IoU of masks. For those not matched, we manually add the corresponding predicted objects with zero saliency ranking as penalization. Let $\mathbf{r}_g$ denote the saliency rankings of ground-truth objects, and $\mathbf{r}_{p\to g}$ indicate the saliency rankings of the corresponding predicted object after matching.

TABLE II
Values of Some Critical Hyper-Parameters

Modules	Parameters	Values
IG module & SPG module	Input size	$W, H = 480, 640$
	K value in the IG module	256
	Iteration number	2
	Learning rate	5×5^{-3}
	Weight decay	5×10^{-5}
RPG module	Input size	$W, H = 480, 640$
	Iteration number	2
	Learning rate	5×5^{-3}
	Weight decay	5×10^{-5}
	Coefficient for λ_{R} and λ_{M} in (11)	$\lambda_{\mathrm{R}} = 0.5, \lambda_{\mathrm{M}} = 0.5$
	Coefficient for weights in (12)	$\lambda_{\mathrm{IG}} = 0.33, \lambda_{\mathrm{SPG}} = 0.33$

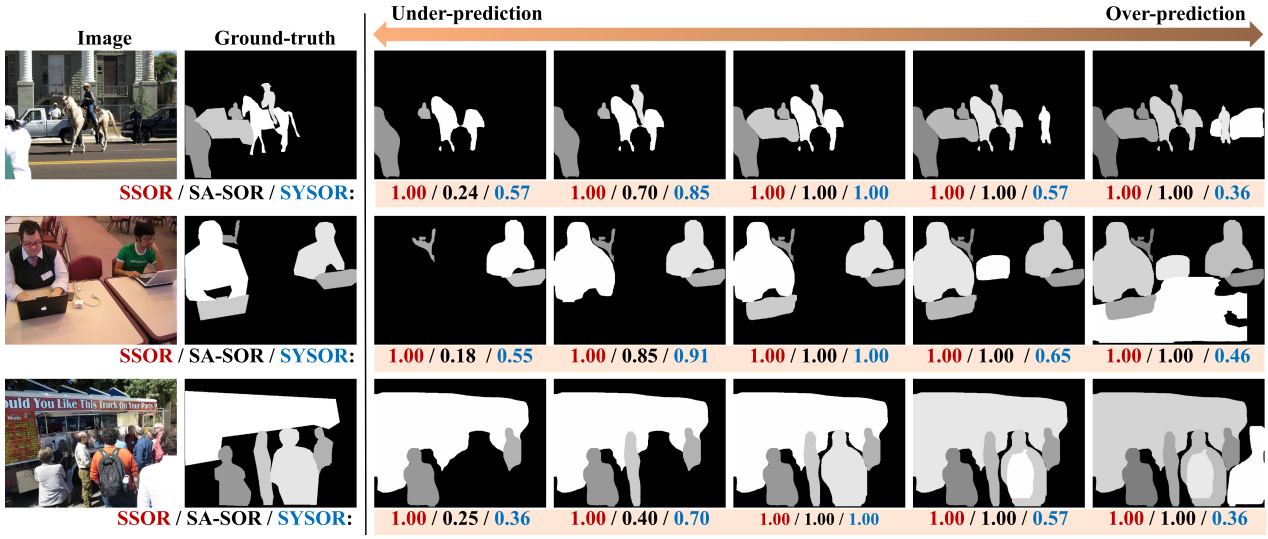


Fig. 12. Examples of SSOR (in red), SA-SOR (in black) and our SYSOR (in blue) when facing under- and over- prediction. From the 3rd to 7th columns are SOR maps with increasing predicted salient objects.

Moreover, the same matching process is conducted over the predicted salient objects, to obtain $\mathbf{r}_p$ and $\mathbf{r}_{g\to p}$. In this way, redundant objects in over-prediction are penalized with ground-truth saliency rankings of zeros. Finally, SYSOR can be defined as

$$\text{SYSOR} = \frac{\text{CC}\left(\mathbf{r}_g, \mathbf{r}_{p\to g}\right) + \text{CC}\left(\mathbf{r}_p, \mathbf{r}_{g\to p}\right)}{2}, \tag{13}$$

where $\text{CC}(\cdot,\cdot)$ is the correlation operation. Mathematically, the first and the second items in (13) can efficiently penalize the under-prediction and over-prediction, respectively. As shown in Fig. 12, the scores of our SYSOR metric are more reasonable in comparison with SSOR and SA-SOR.

B. Evaluation on Salient Object Ranking

In this section, we compare the performance of our HyperSOR with 11 state-of-the-art approaches. Among them, RSDNet [6], Fang et al. [9] and Liu et al. [8] are the deep learning based SOR approaches with released codes. Besides, we also adopt 6 state-of-the-art SOD approaches (i.e., PoolNet+ [20], EDN [19], BASNet [61], CPD-R [62], SCRN [63], and EGNet [64]), and 2 semantic image segmentation (SIS) approaches (i.e., Mask R-CNN [55] and CenterMask [60] under two different backbones). Note that we re-implement SOD approaches by using the same object proposals as ours. For SIS approaches, we take the averaged confidence value of each instance mask as its saliency value, and fine-tuned them under SOR ground-truth. For fair comparison, we re-train all compared approaches over our training set, the same as our HyperSOR approach. Besides, the key hyperparameters of compared methods are tuned to be optimal, according to the SOR performance. More details of the training process can be found in the supplementary material.

Evaluation on SalSOD: Table III presents the evaluation results of HyperSOR and 11 compared approaches on the test set of our SalSOD dataset. From this table, we can observe that our HyperSOR considerably outperforms other approaches in all four metrics. Specifically, compared with the second best approach, our HyperSOR obtain the gain of 0.028 in SYSOR and 0.061 in SA-SOR, achieving 4.0% and 9.3% improvements, respectively. Besides, among the SOR approaches, our HyperSOR further improves 0.188, 0.047 and 0.028 in SYSOR, in comparison with RSDNet [6], Liu et al. [8] and Fang et al. [9], respectively. This verifies the effectiveness of our HyperSOR approach on the task of SOR. Moreover, we can observe that SOR approaches, such as our HyperSOR and Fang et al. [9], generally performs better than SOD and SIS approaches though they are trained with the same data, which implies that SOD and SIS methods are not adaptive for efficiently ranking salient objects. To evaluate the statistical significance of our approach, we follow [65], [66] to conduct paired t-test between our HyperSOR and each compared approach. As shown in Table I of the supplemental material, almost all p-values are significantly less than the 5% significance level, indicating that our HyperSOR approach is statistically better than all compared approaches.

In addition to the quantitative results, Fig. 13 visualizes the qualitative results of our and 11 compared approaches. As can be seen, the SOR maps generated by our HyperSOR approach are closer to the ground-truth, compared with other approaches. To be more specific, HyperSOR is able to well segment the salient objects and correctly predict the saliency values of each object. Furthermore, our HyperSOR makes less over- or under-prediction on salient objects, in comparison to other approaches. These results indicate that our approach is capable of better segmenting and ranking salient objects of the images in various scenes. Besides, we analyze the failure cases of our approach, which are detailed in the supplementary material.

Evaluation of other datasets: To evaluate the generalization ability of our approach, we further compare our HyperSOR with 11 other approaches on two additional SOR datasets, i.e., PASCAL-S [6], Siris' dataset [7]. Here, we also measure the performance of our and compared approaches in terms of SYSOR, SA-SOR, MAE, and SSOR. Note that both our and the

Fig. 13. Subjective results of predicted SOR maps by our and 11 compared approaches.

TABLE III
SOR RESULTS OF OUR AND OTHER 11 APPROACHES OVER SALSOD, SIRIS'S DATASET [7], AND PASCAL-S [6]

Approach	Task	SalSOD				Siris [7]				PASCAL-S [6]			
		SYSOR	SA-SOR	MAE	SSOR	SYSOR	SA-SOR	MAE	SSOR	SYSOR	SA-SOR	MAE	SSOR
Mask R-CNN(R50) [55]	SIS	0.628	0.615	0.142	0.777	0.595	0.545	0.123	0.781	0.653	0.702	0.112	0.898
Mask R-CNN(R101) [55]		0.644	0.633	0.136	0.795	0.607	0.568	0.119	0.795	0.683	0.672	0.106	0.900
CenterMask(V39) [60]		0.648	0.624	0.132	0.784	0.605	0.552	0.113	0.791	0.681	0.648	0.097	0.845
CenterMask(V99) [60]		0.660	0.638	0.128	0.807	0.614	0.559	0.110	0.810	0.722	0.706	0.094	**0.909**
PoolNet+ [20]	SOD	0.573	0.587	0.116	0.801	0.473	0.547	0.120	0.777	0.733	0.623	0.086	0.885
EDN [19]		0.576	0.592	**0.104**	0.805	0.487	0.563	0.110	0.798	0.721	0.609	**0.080**	0.899
BASNet [61]		0.519	0.537	0.127	0.774	0.435	0.493	0.132	0.765	0.715	0.625	0.087	0.871
CPD-R [62]		0.559	0.574	0.119	0.786	0.466	0.529	0.123	0.774	0.719	0.604	0.089	0.898
SCRN [63]		0.569	0.587	0.121	0.802	0.477	0.552	0.126	0.788	0.740	0.650	0.091	0.902
EGNet [64]		0.570	0.585	0.117	0.799	0.480	0.557	0.121	0.788	0.727	0.623	0.089	0.903
RSDNet [6]	SOR	0.546	0.567	0.156	0.785	0.459	0.521	0.155	0.765	0.733	0.629	0.116	0.882
Fang et al. [9]		0.706	0.645	0.115	0.833	0.646	0.598	**0.082**	0.828	0.753	0.671	0.085	0.901
Liu et al. [8]		0.687	0.660	0.123	0.807	0.614	0.628	0.115	0.801	0.718	0.694	0.108	0.885
HyperSOR (Ours)		**0.734**	**0.721**	**0.104**	**0.840**	**0.676**	**0.653**	0.101	**0.830**	**0.780**	**0.717**	0.084	0.904

Note that SIS, SOD, and SOR are the approaches of semantic image segmentation, salient object detection, and salient object ranking, respectively.
The best results are in bold and the second best are marked underlined.

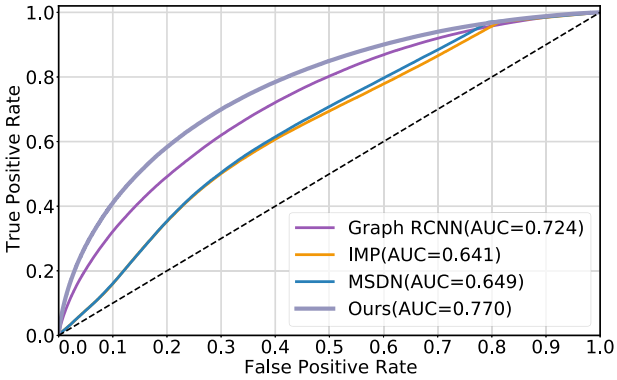


Fig. 14. ROC curve of the relationship proposal by our HyperSOR approach, IMP, MSDN and Graph RCNN.

compared SIS and SOD approaches are trained on our SalSOD dataset, and then directly applied to predict SOR maps over PASCAL-S and Siris' datasets. The results are also reported in Table III. It can be seen that our approach again outperforms all other approaches in almost all metrics. For instance, our HyperSOR approach outperforms the second best approach by 0.030 and 0.027 in SYSOR, over the datasets of Siris [7] and PASCAL-S [6], respectively. Besides, it is interesting to find that SIS and SOR approaches can achieve relative better performance in PASCAL-S dataset, in comparison with SalSOD and Siris' datasets. That is probably because the images in PASCAL-S are simpler with fewer salient objects. Overall, we can conclude that our HyperSOR approach possesses high generalization ability in the task of salient object ranking over different datasets.

C. Ablation Analysis

Here, we evaluate the effectiveness of different components of our model on SOR. We conduct experiments with different network configurations and design options. The ablation experiments are all conducted over our proposed SalSOD dataset.

TABLE IV
PERFORMANCE OF RELATIONSHIP PROPOSAL ON VG150

Approach	Accuracy	Precision	Recall	F1 Score	AUC
IMP [28]	0.876	0.144	0.254	0.142	0.641
MSDN [27]	0.882	0.145	0.246	0.140	0.649
Graph RCNN [29]	0.845	0.210	0.257	0.185	0.724
IG module (Our)	**0.901**	**0.213**	**0.277**	**0.194**	**0.770**

The best results are in bold.

TABLE V
RESULTS OF SALIENT OBJECT RANKING FOR ABLATING THE MGA BLOCK

Approach	SYSOR	SAOR	MAE	SSOR
MGA-AVERA	0.726	0.623	0.114	0.835
MGA-H1	0.733	0.718	0.111	0.838
MGA-H3	0.734	0.719	0.112	0.825
MGA-H4	0.731	0.715	0.113	0.819
MGA-H2 (Ours)	**0.734**	**0.721**	**0.104**	**0.840**

The best results are in bold.

TABLE VI
RESULTS OF SALIENT OBJECT RANKING FOR ABLATING THE HG BLOCK

Approach	SYSOR	SAOR	MAE	SSOR
w/o HG	0.720	0.707	0.106	0.829
HG-CONCAT	0.726	0.714	0.105	0.835
HG-SUM	0.723	0.717	0.106	0.832
HG-MULTI	0.723	0.714	0.106	0.833
HG-HYPER (Ours)	**0.734**	**0.721**	**0.104**	**0.840**

The best results are in bold.

Ablation on the IG module: For ablating the IG module, we regard the relationship proposal as a binary classification task. Specifically, we compare the relationship proposal results of our IG module with 3 state-of-the-art approaches, i.e., Iterative Message Passing (IMP) [28], Multi-level Scene Description Network (MSDN) [27] and Graph R-CNN [29], over the test set of VG150 [28]. In Fig. 14, we draw the receiver operating

TABLE VII
ABLATION ANALYSIS ON THE CHOICE OF GRAPH NEURAL NEURAL NETWORKS IN OUR HYPERSOR APPROACH

Approach	SalSOD				Siris [7]				PASCAL-S [6]			
	SYSOR	SA-SOR	MAE	SSOR	SYSOR	SA-SOR	MAE	SSOR	SYSOR	SA-SOR	MAE	SSOR
HyperSOR-GCN [67]	0.722	0.705	0.104	0.828	0.655	0.623	**0.099**	0.814	0.762	0.687	0.085	0.892
HyperSOR-TransGCN [68]	0.726	0.711	0.105	0.836	0.666	0.638	0.102	0.824	0.773	0.713	0.087	0.897
HyperSOR (Ours)	**0.734**	**0.721**	**0.104**	**0.840**	**0.676**	**0.653**	0.101	**0.830**	**0.780**	**0.717**	**0.084**	**0.904**

The best results are in bold.

characteristic (ROC) curves of our and 3 baseline models. Furthermore, Table IV tabulates the averaged results of accuracy, precision, recall, F_1 score, and area under ROC curve (AUC) of our and 3 approaches. The above results show that our IG module performs better than the baseline models, and therefore verify the effectiveness of IG module in HyperSOR.

Ablation on the MGA block: Here, we conduct ablation experiments on the MGA block for the task of SOR. Table V presents the SYSOR, SA-SOR, MAE, and SSOR results of the ablated models that are trained with the same setting. Specifically, we investigate the effectiveness of our MGA block by replacing it with simple average aggregation, denoted as MGA-AVERA. It can be observed in Table V that the performance drops significantly after ablating the MGA block. Besides, we further compare the performance when conducting different numbers of attention heads in the MGA block. Let MGA-H1, MGA-H3, and MGA-H4 denote the MGA blocks with head numbers of 1, 3, and 4, respectively. Note that the default number of our MGA block is 2, which is denoted as ours in Table V. As can be seen from Table V, the default setting achieves the best performance. In summary, the above results verify the effectiveness of our MGA block for salient object ranking.

Ablation on the HG block: Here, we evaluate the effectiveness of HG block in our HyperSOR approach through the following ablation experiments. Firstly, we remove the whole RPG module with HG blocks from our HyperSOR approach, which results in a simple baseline denoted as "w/o HG". Secondly, in each HG block, the hypernetwork is replaced by the classical operations of feature concatenation, summation, and multiplication, respectively. In Table VI, these models are denoted as "HG-CONCAT", "HG-SUM" and "HG-MULTI", respectively. We can see that SOR performance significantly degrades after removing the HG block. Besides, it can be observed that, compared with wo HG, the operations of feature concatenation, summation, and multiplication can improve SOR performance, though it is less effective than the developed hypernetwork in the HG block. This implies that our proposed HG block is more effective in leveraging the learned scene context to facilitate salient object ranking.

Ablation on GAT: To evaluate the effectiveness of GAT, we introduce two baseline models, where the GAT in our HyperSOR approach is replaced by the regular GCN [67] and Transformer GCN [68], respectively. The above models are denoted as HyperSOR-GCN and HyperSOR-TransGCN. Note that the label propagation algorithm in [68] is removed due to its incompatibility with our tasks. We then train and evaluate these baseline models under the same experimental settings as ours. The performance of scene graph generation is shown in Table VIII. We can see that both HyperSOR-GCN and HyperSOR-TransGCN are inferior to the GAT version of HyperSOR. The performance of HyperSOR-TransGCN is slightly better than that of HyperSOR-GCN, but is still worse than the GAT version of HyperSOR with a degradation of 2.1 and 3.8 at R@50 on SGDet and SGCls, respectively. The reason is probably that GCN and Transformer GCN heavily rely on the graph structure in the training stage, and thus cannot well generalize to unseen graphs in the test stage. Similar observation can be found in SOR results in Table VII. As shown, the GAT version of HyperSOR again outperforms that of HyperSOR-GCN and HyperSOR-TransGCN in terms of almost all four SOR metrics, over all three compared databases. The above results verify the effectiveness of GAT for the tasks of scene graph generation and salient object ranking.

TABLE VIII
PERFORMANCE OF SCENE GRAPH GENERATION BY DIFFERENT APPROACHES

Model	SGDet		SGCls	
	R@50	R@100	R@50	R@100
IMP [28]	20.7	24.5	34.6	35.4
Unbiased [69]	19.4	23.2	25.4	27.9
MSDN [27]	10.7	14.2	24.3	26.5
Graph RCNN [29]	11.4	13.7	29.6	31.6
HyperSOR-GCN [67]	21.3	24.4	33.8	36.7
HyperSOR-TransGCN [68]	22.1	24.6	35.4	37.6
HyperSOR	**24.2**	**27.1**	**39.2**	**41.3**

The best results are in bold.

D. Evaluation on Scene Graph Generation

The SPG module in our HyperSOR approach learns to extract context information and generate the scene graph of the input image. Here, we evaluate the performance of scene graph generation of our HyperSOR approach. Specifically, the generated scene graph from our SPG module is evaluated over the test set of VG150, in comparison with IMP [28], Unbiased [69], MSDN [27] and Graph R-CNN [29]. Then, the generated scene graphs are evaluated under two sub-tasks: scene graph detection (SGDet) and scene graph classification (SGCls). In brief, SGDet needs to detect the objects and then predict their pair-wise relationships. SGCls only needs to classify the categories of objects and relationships, based on the ground-truth object proposals. For more details about SGDet and SGCls, please refer to [28]. Subsequently, we follow [28], [29] to calculate the recall metric R@K (K=50 and 100) for SGDet and SGCls, in which K indicates the top-K predictions. As shown in Table VIII, the performance of SGDet and SGCls by our SPG module are higher than those of the baseline approaches. Besides, Fig. 15 illustrates some generated scene graphs of our and baseline approaches. As seen in this figure, our SPG module is able to

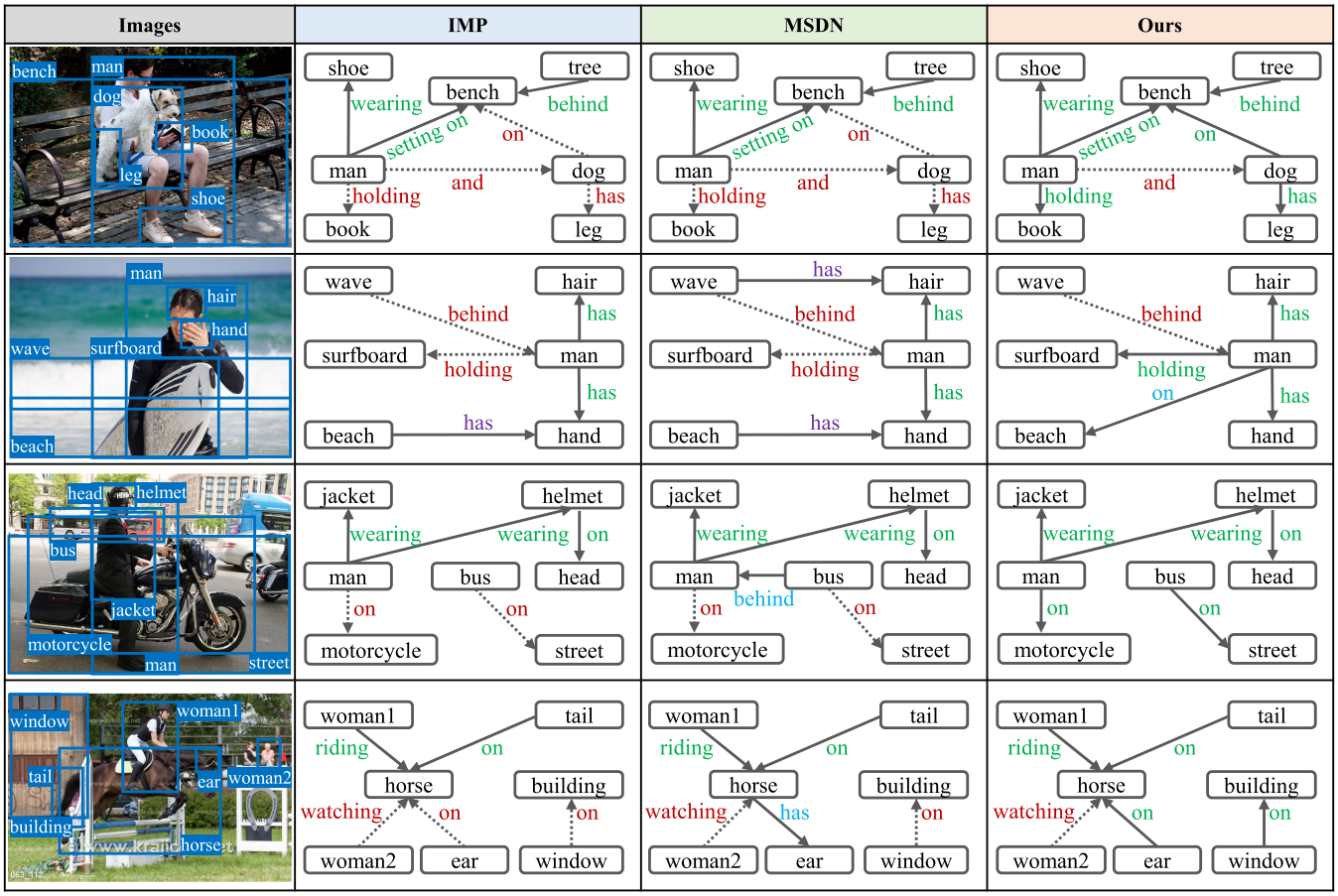


Fig. 15. Examples of scene graphs generated by IMP, MSDN, and our SPG module over the VG150 dataset. Note that the predicates in green are the correct predictions, while the predicates in red and purple indicate uncaught ground-truth and misclassified relationships, respectively. Besides, the blue predicates are reasonable predictions but without ground truth.

capture the informative relationships and accurately predict the scene graph of the image. For example, our approach captures the relationships of $\langle man \rightarrow holding \rightarrow book \rangle$ in the first example, $\langle man \rightarrow holding \rightarrow surfboard \rangle$ in the second example, and $\langle man \rightarrow on \rightarrow motorcycle \rangle$ in the third example. In summary, the experimental results validate that our SPG module can comprehensively capture the scene context information for the task of SOR prediction.

VII. Conclusion

In this paper, we propose the HyperSOR approach to predict the saliency rankings of salient objects in a context-aware manner. Specifically, we established a large-scale SOR dataset with 24,373 images, which is a first dataset including scene graph annotations in addition to the saliency rankings. Then, we found that the saliency rankings of objects are closely correlated with their surrounding scene context, through the analysis over our dataset. Motivated by our findings, we proposed the context-aware graph hypernetwork for SOR. In our approach, we developed an initial graph module to detect the objects and construct the initial graph with semantic information. Subsequently, we proposed a scene perception module based on multi-level graph neural networks, in which the scene context is captured and the scene graphs are generated. Finally, we designed a hypernetwork-based rank prediction module to dynamically embed the learned scene context, such that the saliency rankings of objects can be accurately inferred. The extensive experiments show that the proposed HyperSOR approach significantly outperforms 11 state-of-the-art approaches in saliency ranking prediction over 3 datasets. In the future, it is interesting to explore the practical applications of our HyperSOR approach. For instance, the predicted SOR map can be used to locate region-of-interest for plenty of computer vision and multimedia tasks, such as image compression, object tracking, and image quality assessment. On the other hand, it is also an interesting future work to extend our approach for SOR on videos, in which a dynamic GNN should be developed to learn the temporal relationship of the objects across frames.

References

[1] E. Matin, "Saccadic suppression: A review and an analysis," *Psychol. Bull.*, vol. 81, no. 12, 1974, Art. no. 899.
[2] M. Xu, L. Jiang, X. Sun, Z. Ye, and Z. Wang, "Learning to detect video saliency with HEVC features," *IEEE Trans. Image Process.*, vol. 26, no. 1, pp. 369–385, Jan. 2017.
[3] W. Wang, J. Shen, and H. Ling, "A deep network solution for attention and aesthetics aware photo cropping," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1531–1544, Jul. 2019.
[4] G. Zhang, Z. Yuan, N. Zheng, X. Sheng, and T. Liu, "Visual saliency based object tracking," in *Proc. Asian Conf. Comput. Vis.*, 2009, pp. 193–203.

[5] F. Zhang, B. Du, and L. Zhang, "Saliency-guided unsupervised feature learning for scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 4, pp. 2175–2184, Apr. 2015.

[6] M. A. Islam, M. Kalash, and N. D. Bruce, "Revisiting salient object detection: Simultaneous detection, ranking, and subitizing of multiple salient objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7142–7150.

[7] A. Siris, J. Jiao, G. K. Tam, X. Xie, and R. W. Lau, "Inferring attention shift ranks of objects for image saliency," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12133–12143.

[8] N. Liu, L. Li, W. Zhao, J. Han, and L. Shao, "Instance-level relative saliency ranking with graph reasoning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 8321–8337, Nov. 2022.

[9] H. Fang et al., "Salient object ranking with position-preserved attention," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 16331–16341.

[10] X. Tian, K. Xu, X. Yang, L. Du, B. Yin, and R. W. Lau, "Bi-directional object-context prioritization learning for saliency ranking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 5882–5891.

[11] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[12] M. B. Neider and G. J. Zelinsky, "Scene context guides eye movements during visual search," *Vis. Res.*, vol. 46, no. 5, pp. 614–621, 2006.

[13] A. Torralba, A. Oliva, M. S. Castelhano, and J. M. Henderson, "Contextual guidance of eye movements and attention in real-world scenes: The role of global features in object search.," *Psychol. Rev.*, vol. 113, no. 4, Art. no. 766, 2006.

[14] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 569–582, Mar. 2015.

[15] P. L. Rosin, "A simple method for detecting salient regions," *Pattern Recognit.*, vol. 42, no. 11, pp. 2363–2371, 2009.

[16] D. A. Klein and S. Frintrop, "Center-surround divergence of feature statistics for salient object detection," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 2214–2219.

[17] K. Shi, K. Wang, J. Lu, and L. Lin, "Pisa: Pixelwise image saliency by aggregating complementary appearance contrast measures with spatial priors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2115–2122.

[18] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 10, pp. 1915–1926, Oct. 2012.

[19] Y.-H. Wu, Y. Liu, L. Zhang, M.-M. Cheng, and B. Ren, "EDN: Salient object detection via extremely-downsampled network," *IEEE Trans. Image Process.*, vol. 31, pp. 3125–3136, 2022.

[20] J.-J. Liu, Q. Hou, Z.-A. Liu, and M.-M. Cheng, "PoolNet: Exploring the potential of pooling for salient object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 887–904, Jan. 2023.

[21] L. Zhang, J. Zhang, Z. Lin, H. Lu, and Y. He, "CapSal: Leveraging captioning to boost semantics for salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6024–6033.

[22] Y. Zeng, Y. Zhuge, H. Lu, L. Zhang, M. Qian, and Y. Yu, "Multi-source weak supervision for saliency detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6074–6083.

[23] H. Zhang, Y. Zeng, H. Lu, L. Zhang, J. Li, and J. Qi, "Learning to detect salient object with multi-source weak supervision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3577–3589, Jul. 2022.

[24] Z. Wang, X. Yan, Y. Han, and M. Sun, "Ranking video salient object detection," in *Proc. 27th ACM Int. Conf. Multimedia*, 2019, pp. 873–881.

[25] J. Lin, H. Guan, and R. W. Lau, "Rethinking video salient object ranking," 2022, *arXiv:2203.17257*.

[26] J. Johnson et al., "Image retrieval using scene graphs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3668–3678.

[27] Y. Li, W. Ouyang, B. Zhou, K. Wang, and X. Wang, "Scene graph generation from objects, phrases and region captions," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1261–1270.

[28] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei, "Scene graph generation by iterative message passing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5410–5419.

[29] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh, "Graph R-CNN for scene graph generation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 670–685.

[30] R. Zellers, M. Yatskar, S. Thomson, and Y. Choi, "Neural motifs: Scene graph parsing with global context," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5831–5840.

[31] R. Li, S. Zhang, B. Wan, and X. He, "Bipartite graph network with adaptive message passing for unbiased scene graph generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 11109–11119.

[32] R. Krishna et al., "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *Int. J. Comput. Vis.*, vol. 123, no. 1, pp. 32–73, 2017.

[33] M. Khademi and O. Schulte, "Deep generative probabilistic graph neural networks for scene graph generation," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 11237–11 245.

[34] A. Dornadula, A. Narcomey, R. Krishna, M. Bernstein, and F.-F. Li, "Visual relationships as functions: Enabling few-shot scene graph prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops*, 2019, pp. 0–0.

[35] J. Johnson, A. Gupta, and L. Fei-Fei, "Image generation from scene graphs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1219–1228.

[36] G. Mittal, S. Agrawal, A. Agarwal, S. Mehta, and T. Marwah, "Interactive image generation using scene graphs," 2019, *arXiv: 1905.03743*.

[37] X. Li and S. Jiang, "Know more say less: Image captioning based on scene graphs," *IEEE Trans. Multimedia*, vol. 21, no. 8, pp. 2117–2130, Aug. 2019.

[38] X. Yang, H. Zhang, and J. Cai, "Auto-encoding and distilling scene graphs for image captioning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2313–2327, May 2022.

[39] Z. Yang, Z. Qin, J. Yu, and Y. Hu, "Scene graph reasoning with prior visual relationship for visual question answering," 2018, *arXiv: 1812.09681*.

[40] M. Hildebrandt, H. Li, R. Koner, V. Tresp, and S. Günnemann, "Scene graph reasoning for visual question answering," 2020, *arXiv: 2007.01072*.

[41] S. Ramnath, A. Saha, S. Chakrabarti, and M. M. Khapra, "Scene graph based image retrieval–a case study on the clevr dataset," 2019, *arXiv: 1911.00850*.

[42] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[43] X. Huang, C. Shen, X. Boix, and Q. Zhao, "SALICON: Reducing the semantic gap in saliency prediction by adapting deep neural networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 262–270.

[44] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara, "Predicting human eye fixations via an LSTM-based saliency attentive model," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5142–5154, Oct. 2018.

[45] G. A. Miller, "Wordnet: A lexical database for english," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[46] S. Bird, "NLTK: The natural language toolkit," in *Proc. COLING/ACL Interactive Presentation Sessions*, 2006, pp. 69–72.

[47] M. Kalash, M. A. Islam, and N. D. Bruce, "Relative saliency and ranking: Models, metrics, data and benchmarks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 204–219, Jan. 2021.

[48] A. Borji and L. Itti, "State-of-the-art in visual attention modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 185–207, Jan. 2013.

[49] B. W. Tatler, "The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions," *J. Vis.*, vol. 7, no. 14, pp. 4–4, 2007.

[50] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[51] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2117–2125.

[52] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[53] X. Lu, W. Wang, J. Shen, D. Crandall, and L. V. Gool, "Segmenting objects from relational visual data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 7885–7897, Nov. 2022.

[54] A. Luo, X. Li, F. Yang, Z. Jiao, H. Cheng, and S. Lyu, "Cascade graph neural networks for RGB-D salient object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 346–364.

[55] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2961–2969.

[56] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3588–3597.

[57] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=rJXMpikCZ

[58] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist. Workshop Conf.*, 2010, pp. 249–256.

[59] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations*, San Diego, CA, USA, 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[60] Y. Lee and J. Park, "CenterMask: Real-time anchor-free instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 13906–13915.
[61] X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, and M. Jagersand, "BASNet: Boundary-aware salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7479–7489.
[62] Z. Wu, L. Su, and Q. Huang, "Cascaded partial decoder for fast and accurate salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3907–3916.
[63] Z. Wu, L. Su, and Q. Huang, "Stacked cross refinement network for edge-aware salient object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 7264–7273.
[64] J.-X. Zhao, J.-J. Liu, D.-P. Fan, Y. Cao, J. Yang, and M.-M. Cheng, "EGNet: Edge guidance network for salient object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 8779–8788.
[65] G. Li, Z. Liu, and H. Ling, "ICNet: Information conversion network for RGB-D based salient object detection," *IEEE Trans. Image Process.*, vol. 29, pp. 4873–4884, 2020.
[66] H.-H. Yeh, K.-H. Liu, and C.-S. Chen, "Salient object detection via local saliency estimation and global homogeneity refinement," *Pattern Recognit.*, vol. 47, no. 4, pp. 1740–1750, 2014.
[67] M. Welling and T. N. Kipf, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2016, pp. 1–14.
[68] Y. Shi, Z. Huang, S. Feng, H. Zhong, W. Wang, and Y. Sun, "Masked label prediction: Unified message passing model for semi-supervised classification," 2020, *arXiv: 2009.03509*.
[69] K. Tang, Y. Niu, J. Huang, J. Shi, and H. Zhang, "Unbiased scene graph generation from biased training," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3716–3725.

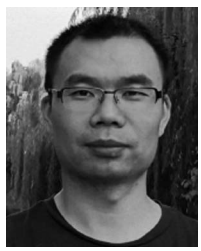
Minglang Qiao received the BS degree from the School of Electronic and Information Engineering, Beihang University in 2018. He is currently working toward the PhD degree with MC^2 Lab, Beihang University. His research interests mainly include saliency detection of images and videos, and salient object detection. He has published several papers in the international journals and conference proceedings, e.g., *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *International Journal of Computer Vision*, ECCV.

Mai Xu (Senior Member, IEEE) received the BS degree from Beihang University, Beijing, China, in 2003, the MS degree from Tsinghua University, Beijing, China, in 2006, and the PhD degree from Imperial College London, London, U.K., in 2010. From 2010 to 2012, he was a research fellow with the Department of Electrical Engineering, Tsinghua University. Since January 2013, he has been with Beihang University, where he was an associate professor and was promoted to full professor in 2019. From 2014 to 2015, he was a visiting researcher with MSRA. His main research interests include image processing and computer vision. He has authored or coauthored more than 200 technical papers in international journals and conference proceedings, e.g., *International Journal of Computer Vision*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Image Processing*, *IEEE Journal of Selected Topics in Signal Processing*, CVPR, ICCV, ECCV, and AAAI. He is the recipient of the best/top paper awards of IEEE/ACM conferences, such as ACM MM. He served as an associate editor of *IEEE Transactions on Image Processing* and *IEEE Transactions on Multimedia*, a lead guest editor of *IEEE Journal of Selected Topics in Signal Processing*, and an area chair or TPC Member for many conferences, such as ICME, AAAI, etc. He received outstanding AE awards twice (Years 2021 and 2022). He is an elected member of the Multimedia Signal Processing Technical Committee, IEEE Signal Processing Society.

Lai Jiang (Member, IEEE) received the BS and PhD degrees from Beihang University, China, in 2015 and 2021, respectively. From 2021 to 2023, he worked as research scientist and post postdoctoral researcher with Alibaba Cloud, China, and University of British Columbia, Canada. Currently, he is working as an associate professor with the Department of Electronic Information Engineering, Beihang University, China. His research interests include computer vision, medical image processing, and multimedia. He has published more than 30 technical papers in top-tier journals and conference, including *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *International Journal of Computer Vision*, *IEEE Transactions on Image Processing*, *IEEE Transactions on Medical Imaging*, CVPR, ICCV, AAAI, and so forth.

Peng Lei received the BS and PhD degrees in electrical engineering from Beihang University, Beijing, China, in 2006 and 2012, respectively. He is currently an associate professor with the School of Electronic and Information Engineering, Beihang University. His research interests include signal processing, image processing, target recognition, and machine learning in electrical engineering.

Shijie Wen received the BS degree from the School of Information Science and Technology, Beijing University of Chemical Technology, in 2017. He is currently working towards the PhD degree with Beihang University. His research interests mainly include video saliency prediction and video quality assessment.



Yunjin Chen received the BSc degree in applied physics from the Nanjing University of Aeronautics and Astronautics, China, the MSc degree in optical engineering from the National University of Defense Technology, China, and the PhD degree in computer science from the Graz University of Technology, Austria, in 2007, 2010, and 2015, respectively. Currently, he is a senior algorithm expert with Alibaba Cloud. His current research interests include image/video restoration, and its application with cloud video transcoding system.

Leonid Sigal received the BSc degrees in computer science and mathematics from Boston University, in 1999, the MA degree from Boston University, in 1999, the MS degree from Brown University, in 2003, and the PhD degree from Brown University, in 2008. He is an associate professor with the Department of Computer Science, University of British Columbia. Until 2017, he was a senior research scientist with Disney Research. From 2007 to 2009, he was a postdoctoral fellow with the Department of Computer Science, University of Toronto. From 1999 to 2001, he worked as a senior vision engineer with Cognex Corporation, where he developed industrial vision applications for pattern analysis and verification. His research interests mainly include the areas of computer vision, machine learning, and computer graphics. He has published more than 70 peer reviewed papers in venues and journals in these fields. His work received the Best Paper Awards at the AMDO conference in 2006 /2012 and at WACV in 2014. He has also coedited the book Guide to Visual Analytics of Humans: Looking at People (Springer, 2011)