Image and Vision Computing 129 (2023) 104590

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Exploring viewport features for semi-supervised saliency prediction in omnidirectional images

Mengke Huang [a,b], Gongyang Li [a,b,*], Zhi Liu [a,b], Yong Wu [a,b], Chen Gong [c,d], Linchao Zhu [e], Yi Yang [e]

[a] Shanghai Institute for Advanced Communication and Data Science, Shanghai University, Shanghai 200444, PR China
[b] School of Communication and Information Engineering, Shanghai University, Shanghai 200444, PR China
[c] PCA Lab, the Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, PR China
[d] Department of Computing, Hong Kong Polytechnic University, Hong Kong, PR China
[e] ReLER Lab, Australian Artificial Intelligence Institute, University of Technology Sydney, Ultimo, NSW 2007, Australia

ARTICLE INFO

Article history:
Received 2 August 2022
Received in revised form 20 October 2022
Accepted 15 November 2022
Available online 23 November 2022

Keywords:
Omnidirectional image
Saliency prediction
Semi-supervised learning

ABSTRACT

Compared with the annotated data for the 2D image saliency prediction task, the annotated data for training omnidirectional image (or 360° image) saliency prediction models are not sufficient. Most existing fully-supervised saliency prediction methods for omnidirectional images (ODIs) adopt a scheme, first training the methods on a labeled large 2D image saliency prediction dataset and then fine-tuning the methods on the labeled tiny ODI saliency prediction dataset. However, this strategy is time-consuming and may not inadequately mine the visual features built in ODIs. To explore the visual attributes targeted at ODIs and address the shortage of labels on these ODIs, in this paper, we propose an end-to-end semi-supervised network, namely VFNet, which relies on viewport features and only utilizes ODIs as training data, for ODI saliency prediction. Concretely, we adopt consistency regularization as our semi-supervised learning framework. The predictions between main and auxiliary saliency inference networks in the VFNet enforce consistency. Aiming at ODIs, we introduce a new form of perturbation, *i.e.*, DropView, to improve the effectiveness of consistency regularization. By randomly dropping out different 360° cubemap viewport features before the auxiliary saliency inference network, the proposed DropView enhances the robustness of the final ODI saliency prediction. To adaptively interact with the equirectangular and different cubemap viewport features according to their contributions, we introduce a Viewport Feature Adaptive Integration (VFAI) module and deploy the VFAI module at different levels in the VFNet to raise the capacity of feature encoding of our VFNet. Compared with state-of-the-art fully-supervised methods, our VFNet with fewer labeled training data achieves competitive performance demonstrated by extensive experiments on two publicly available datasets.

1. Introduction

Visual saliency prediction, of which the goal is to predict the location of the human visual attention over an image, has been studied extensively in the last decades [1–8]. Saliency prediction is an essential step for many computer vision tasks such as image classification [9], video compression [10], object detection [11], object segmentation [12,13], and image captioning [14]. Many 2D image saliency prediction methods based on Convolutional Neural Networks (CNNs) have reached good performance in the normal Field of View (FoV) because of the sufficient labeled data for training in existing datasets [15,16]. Compared with the training data of 2D image saliency prediction models, however, the labeled data of omnidirectional images (or 360° images) in the publicly available datasets [20,21] can not meet the need for the training of 360° image saliency prediction methods. To mitigate the insufficiency of labeled data, most existing CNN-based omnidirectional image (ODI) saliency prediction methods adopt a time-consuming and computationally expensive training strategy, *i.e.*, pre-training the methods on a 2D image saliency prediction dataset [15] and then fine-tuning them on the small ODI dataset. Nevertheless, different from the typical 2D images with a limited FoV, ODI displays the spatial information of all

* Corresponding author at: Shanghai Institute for Advanced Communication and Data Science, Shanghai University, Shanghai 200444, PR China.
E-mail addresses: huangmengke@shu.edu.cn (M. Huang), ligongyang@shu.edu.cn (G. Li), liuzhisjtu@163.com (Z. Liu), yong_wu@shu.edu.cn (Y. Wu), chen.gong@njust.edu.cn (C. Gong), linchao.zhu@uts.edu.au (L. Zhu), yi.yang@uts.edu.au (Y. Yang).

https://doi.org/10.1016/j.imavis.2022.104590
0262-8856/

directions of the real 3D visual world on the entire viewing sphere and presents challenges to CNN-based ODI saliency prediction methods.

First, due to geometric distortion in its top and bottom regions caused by the EquiRectangular Projection (ERP) [22], 360° equirectangular image, which displays the 3D global semantic information of an ODI on a 2D plane, has visual differences from typical 2D images. Differently, 360° cubemap images display the 360° scene by projecting an ODI onto six viewport images (faces) of a cube, *i.e.*, the CubeMap Projection (CMP) [22], and introduce less distortion into these viewport images (with 90° FoV). However, CMP leads to the lack of global information and the discontinuities in boundaries between the faces of the projected cube. Obviously, the above two representations, *i.e.*, 360° equirectangular and cubemap images, of ODIs are different from those of the typical 2D images. If ODI saliency prediction methods overly depend on typical 2D image data, they may neglect the specific visual cues of ODIs and produce inaccurate saliency maps, such as [19,23,24]. As shown in Fig. 1, for example, saliency maps of SalNet360 [19] highlight ambiguous regions in ODIs.

Second, few existing ODI saliency prediction methods explore the features of 360° equirectangular and cubemap viewport images extracted by CNNs at different levels cooperatively. Most existing CNN-based ODI saliency prediction methods [17–19] only employ the final saliency results predicted from a certain representation of ODI. Due to ignoring the useful semantic features of other representations of ODIs at intermediate levels in CNN, these methods may predict unfaithful saliency distribution such as the results of SalGAN360 [17] and MV-SalGAN360 [18] shown in Fig. 1.

Third, because observers usually do not pay attention to the north and south pole regions of ODIs which correspond to the top and bottom viewports of 360° cubemap images, the visual attention of observers on different viewports are not uniform and this imbalanced visual attention distribution differs in diverse ODIs, as illustrated in Fig. 2. For predicting visual saliency in the whole ODIs, thus, it may be reasonable to adaptively aggregate the CNN features of each cubemap viewport according to the imbalanced distribution of eye fixations.

To solve the problems mentioned above, in this paper, we propose a novel end-to-end semi-supervised Viewport Feature Network (VFNet) for ODI saliency prediction. The semi-supervised framework is inspired by Ouali et al. [25], and is proposed for alleviating the insufficiency of labeled ODI saliency prediction data. For enforcing consistency of the predictions of unlabeled ODIs between main and auxiliary saliency inference networks in the VFNet, we perturb features by different forms of perturbations and our proposed DropView, before injecting these features into auxiliary saliency inference networks. In these saliency inference networks, we deploy trainable adaptive weights to combine the saliency outputs predicted at different levels. For deeply correlating the CNN features of different representations of ODIs and fitting the non-uniform distribution of eye fixations over ODIs, we design a Viewport Feature Adaptive Integration (VFAI) module. We apply VFAI modules to multiple levels of the shared feature encoding network in the VFNet for integrating features of 360° equirectangular and cubemap viewports. The training of labeled and unlabeled ODIs in the VFNet are respectively supervised by labels and generated pseudo labels. The inference stage only relies on the feature encoding network, viewport feature adaptive integration modules and the main saliency inference network in the VFNet.

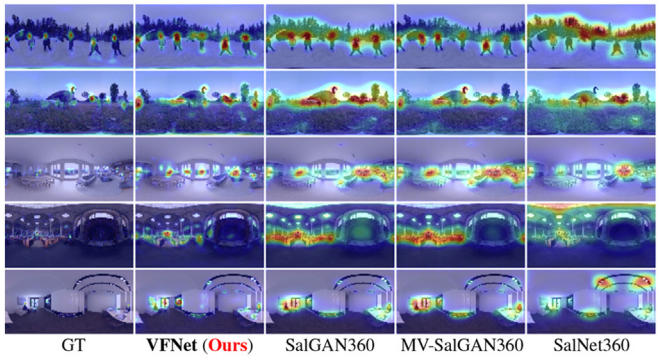

Fig. 1. Saliency maps of our VFNet and three state-of-the-art supervised methods including SalGAN360 [17], MV-SalGAN360 [18] and SalNet360 [19] on ODIs. The saliency maps of our semi-supervised VFNet are visually closer to Ground Truth (GT) than those of three fully-supervised methods.

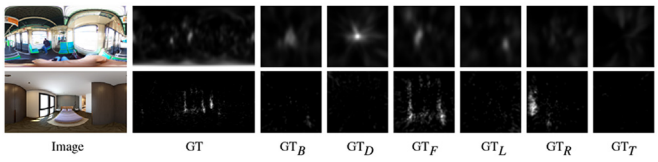

Fig. 2. Examples of non-uniform eye fixation distribution of different cubemap viewports from the Salient360!2017 [20] and Saliency-in-VR [21] datasets, respectively. GT is the ground truth saliency map of 360° equirectangular image, and GT_B, GT_D, GT_F, GT_L, GT_R, and GT_T respectively represent the ground truth saliency maps from back, down, front, left, right and top viewports of the corresponding 360° cubemap images.

The contributions of this paper are summarized as follows:

- We propose a novel semi-supervised ODI saliency prediction method, *i.e.*, VFNet, which considers 360° equirectangular and cubemap features simultaneously. Although VFNet is only trained on few labeled and some unlabeled ODIs, compared with the state-of-the-art ODI saliency prediction methods pre-trained on a large amount of labeled 2D image saliency prediction data, our VFNet achieves competitive performance on two publicly available datasets under four evaluation metrics.
- We propose a new form of perturbation, *i.e.*, DropView, to fully utilize the unlabeled ODIs. DropView focuses on randomly dropping the features of a given cubemap viewport. It improves the generalization capability of the network by avoiding spatial over-reliance on a certain region, which corresponds to a certain cubemap viewport, in the 360° equirectangular features.
- We propose a novel Viewport Feature Adaptive Integration (VFAI) module to fit the non-uniform distribution of visual attention over different viewports of ODIs and integrate 360° equirectangular and cubemap features adaptively. By deploying the attention mechanism, the VFAI module is able to keep the most valuable 360° equirectangular and cubemap viewport features for the final saliency prediction.

2. Related work

2.1. ODI saliency prediction

1) 2D image saliency prediction methods adaptation. The research of 2D visual saliency prediction has made great progress in the past decades [2–6]. In contrast, because of the gradual prevalence of affordable 360° cameras and the development of Virtual Reality (VR) applications, saliency prediction of ODIs has just begun to gain increasing attention in recent years. However, directly applying 2D image saliency prediction methods to ODIs may suffer from geometric distortion and discontinuities caused by ERP and CMP. Thus, several methods have explored to properly extend existing 2D image saliency prediction methods to ODIs.

In [26], a fused saliency map post-processing method is proposed for ODI saliency prediction. This method applies a 2D saliency prediction method to ODIs directly and linearly combines the saliency maps predicted from several shifted ODIs for mitigating the center prior limitation. Furthermore, Lebreton et al. [23] propose a projected saliency framework which integrates the saliency maps predicted by existing 2D image saliency methods, *i.e.*, GBVS [3] and BMS [4], and utilizes the interactions between a given viewport and its neighbouring regions.

In [22], to discover the attention-catching regions of ODIs, a double cube projection is introduced to project the ODI onto two cubes. This method combines the cubemap saliency maps predicted by traditional 2D saliency prediction methods via a weighted average. To mitigate the discontinuities of border, Startsev et al. [24] incorporate CMP with ERP to alleviate border effects caused by directly applying a 2D saliency method to 360° equirectangular images.

Although these methods extended from 2D saliency prediction methods have achieved promising performance to some extent, some attributes of typical 2D images built-in these methods may not satisfy the features of ODI saliency prediction, *e.g.*, the non-uniform eye fixation distribution over the different viewports of ODIs.

2) ODI saliency prediction methods. Benefiting from the development of deep learning, most tailored ODI saliency prediction methods are able to learn exclusive features of ODIs from training data. Monroy et al. [19] propose a two-stage CNN-based method which includes a backbone network and a saliency refinement network. The method is trained on 2D images and patches randomly sampled from ODIs. Chao et al. [17] propose a 360° saliency prediction method fined-tuned on 2D image saliency prediction method SalGAN [27] by fusing 360° equirectangular saliency map and the corresponding 360° cubemap saliency maps. In [18], a multi-FoV viewport-based 360° visual saliency predictor is built by combining three different types of viewport saliency maps, and then trained by an adaptively weighted loss.

These tailor-made methods mentioned above explore to integrate the saliency maps predicted from different representations of ODIs, they ignore the interaction of intermediate features of different representations of ODIs, which are extracted by CNNs. Moreover, most of these methods required the network to be pre-trained on the labeled data-heavy 2D image dataset [15] for saliency prediction, and this time-consuming strategy may lead the learned features to overfit typical 2D images.

In this work, we pay attention to mine the exclusive features of ODIs extracted by CNN. To match the non-uniform eye fixations distribution of ODIs, we explore to modulate the features extracted from 360° equirectangular and cubemap images inside CNN collaboratively. To reduce the over-reliance on 2D images and alleviate the lack of labeled data of ODIs, we only utilize labeled and unlabeled ODIs to train our VFNet via semi-supervised learning.

2.2. Semi-supervised learning

For reducing the huge cost in annotating training data, many efforts based on semi-supervised learning have been exploited in various computer vision tasks [28–32]. In this paper, we adopt the consistency regularization method as our semi-supervised learning framework. This method is under the cluster assumption, *i.e.*, the predictions of unlabeled samples should not have significant variances when a form of perturbation is applied to these examples.

To enforce consistency over different perturbed features, several works have provided different effective solutions. Laine et al. [33] propose a Π-Model which imposes consistency over perturbed inputs by data augmentation and dropout for more stable predictions over unlabeled data. Furthermore, a temporal ensembling method proposed in [33] enforces consistency by averaging previous predictions through moving weighting. Mean teacher model [34] explores to average the model weights instead of predictions for consistency regularization. Different from adding perturbations to input, Ouali et al. [25] demonstrate that perturbations injected into hidden representations (outputs of encoding network) can make the class boundaries of cluster assumption maintain in the low-density region and improve the stability of semi-supervised learning for semantic segmentation.

Inspired by the experiments in [25], we also apply different perturbations to the intermediate features, and then send the perturbed features into auxiliary saliency inference networks in the VFNet. Furthermore, we apply the same type of perturbation to features at multiple levels in VFNet to achieve saliency prediction with improved stability.

2.3. Different forms of perturbations

For perturbing the inputs, features or predictions in consistency regularization, different forms of perturbations have been adopted in previous methods. The most common form is Dropout [35], which avoids the overfitting of the network by dropping the features in the network randomly, leading to a breakthrough in image classification [36]. This work has inspired a series of regularization methods for the training of the neural network.

In [37], the subset of weights within the network are randomly set to zero instead of randomly setting the selected activations to zero within each layer, which is proposed in [35]. Similarly, Larsson et al. [38] explore to set an entire layer in the neural network to zero rather than only a particular activation unit in training. As for CNNs, Tompson et al. [39] introduced a structural noise, *i.e.*, SpatialDropout, to regularize the spatially correlated features activations by dropping entire channels from a feature map. Furthermore, Ghiasi et al. [40] proposed a structured dropout, *i.e.*, DropBlock, to drop all the units in contiguously spatial regions of feature maps together during training, which obtains encouraging performance in several computer vision tasks.

Although many efforts have been made to improve dropout in 2D visual tasks, few methods have explored to achieve the dropout effect in 360° visual tasks. Inspired by DropBlock [40], we design a novel dropout form, namely DropView, and inject this perturbation of 3D spatial correlation to the CNN features of ODIs. Concretely, before the auxiliary saliency inference network, we randomly drop a given cubemap viewport feature and project the dropped cubemap viewport features to equirectangular features by the cubemap-to-equirectangular projection. In this way, we enforce the network to learn features from the most important viewports to improve the representation ability of the network.

3. Methodology

3.1. Network overview

As shown in Fig. 3, the proposed VFNet consists of a shared Feature Encoding (FE) network, Viewport Feature Adaptive Integration (VFAI) modules, Perturbation Modules (PMs), the Main Saliency Inference (MSI) network, and Auxiliary Saliency Inference (ASI) networks.

1) Shared feature encoding network. Considering the computational efficiency, we employ relatively shallow ResNet-34 [41] based Feature Pyramid Network (FPN) [42] for feature encoding and preliminary integration of semantic information. As shown in Fig. 3, the 360° equirectangular image and the corresponding 360° cubemap images are encoded through the shared FE network simultaneously. Following the design of FPN, outputs of the shared FE network are equirectangular features $\boldsymbol{F}_E^l$ and six cubemap features $\boldsymbol{F}_C^l = \{\boldsymbol{F}_C^{lf} | f \in \{B, D, F, L, R, T\}\}$ which correspond to the back, down, front, left, right and top viewport features of the cube, where $l \in \{1, 2, 3, 4\}$ is the l-th stage and the same as FPN.

2) Viewport feature adaptive integration module. Based on the observation of the imbalanced visual saliency in different 360° cubemap viewports shown in Fig. 2, adaptively emphasizing or suppressing features belonging to different viewports may benefit the whole network to predict visual saliency stably. Hence, we aim at correlating 360° equirectangular and cubemap viewport features and adaptively integrating the features in different viewports by attention mechanism in the Viewport Feature Adaptive Integration (VFAI) module. As shown in Fig. 3, VFAI module is the core component and is equipped to different semantic levels in the shared FE network. We introduce the VFAI

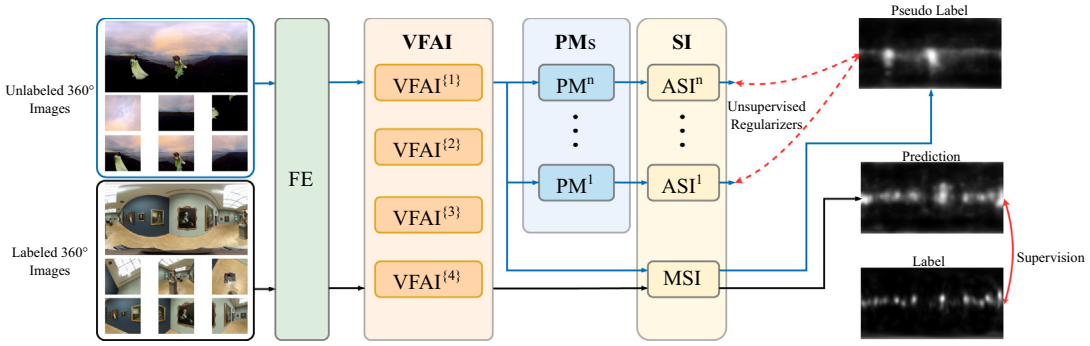

Fig. 3. Pipeline of the proposed VFNet. Our VFNet contains four key stages: Feature Encoding (FE), Viewport Feature Adaptive Integration (**VFAI**), Perturbation Modules (**PMs**) and Saliency Inference (**SI**). First, FE network extracts features from labeled and unlabeled 360° equirectangular images with their corresponding 360° cubemap images simultaneously. Then, VFAI modules adaptively integrate these extracted features by FE network at different levels. Next, these integrated features at different levels in VFAI modules are perturbed in PMs block by different perturbations. Finally, Main Saliency Inference (MSI) network generates saliency maps for labeled and unlabeled images from the features integrated in VFAI modules directly. Besides, Auxiliary Saliency Inference (ASI) networks produce the saliency maps of unlabeled images from the perturbed features.

module in detail in Section 3.2, and examine the effectiveness of the VFAI module and its components in Section 4.5.

3) DropView perturbation. To improve the robustness of the VFNet for small changes, following [25], the consistency training strategy in our semi-supervised learning framework enforces the consistency of results between the MSI network and ASI networks over different perturbations as illustrated in Fig. 3. Focusing on ODIs, particularly, we propose DropView for perturbing spatial correlation of the features between different 360° cubemap viewports. Furthermore, we apply the same type of perturbation to output features of the VFAI modules at different levels for improving the effectiveness of perturbations. We provide the details of the DropView perturbation in Section 3.3, and examine the effectiveness of this module and its components in Section 4.5.

4) Saliency inference networks. The saliency inference (SI) networks are designed for adaptively combining the saliency information of features at different levels of the network. These saliency inference blocks in SI networks are represented as $\mathrm{SI}^{\{l\}}$. For capturing the most valuable saliency information at different levels, we introduce trainable adaptive weights to weight the saliency maps produced at different levels. The final result of saliency prediction is obtained by combining the weighted saliency information at different levels through up-sampling operations. This adaptively weighting structure is applied in both MSI and ASI networks and the structure is elaborated in Section 3.4.

5) Hybrid objective function. Motivated by previous works [43,44], we employ a hybrid bootstrapping loss containing Kullback–Leibler Divergence (KLD) [45], Pearson's Correlation Coefficient (CC) [46] and Normalized Scanpath Saliency (NSS) [47] in the supervised learning. Aiming at unlabeled ODIs in the training phase, we adopt the summation of KLD and Mean Squared Error (MSE) as the unsupervised consistency regularizers over predictions of the MSI network and these ASI networks as shown in Fig. 3. The total objective function of the training stage is the summation of the loss of supervision and the weighted unsupervised regularizer. Following [33], the weight of unsupervised regularizer ramps up starting from zero to a fixed weight along an exponential curve. We present the formulation and ablation study of the hybrid objective function in Sections 3.5 and 4.5, respectively.

3.2. Viewport feature adaptive integration module

Viewport Feature Adaptive Integration (VFAI) module is the key component bridging feature encoding and saliency inference stages. This module is responsible for the feature integration of different viewports in the network. In the following, we elaborate VFAI module $\mathrm{VFAI}^{\{l\}}$ at the l-th level of the VFNet shown in Fig. 4.

The sizes of output equirectangular features $\boldsymbol{F}_E^l$ and six cubemap features $\boldsymbol{F}_C^l$ at the l-th ($l \in \{1,2,3,4\}$) level of FE network are $c \times h_l \times w_l$

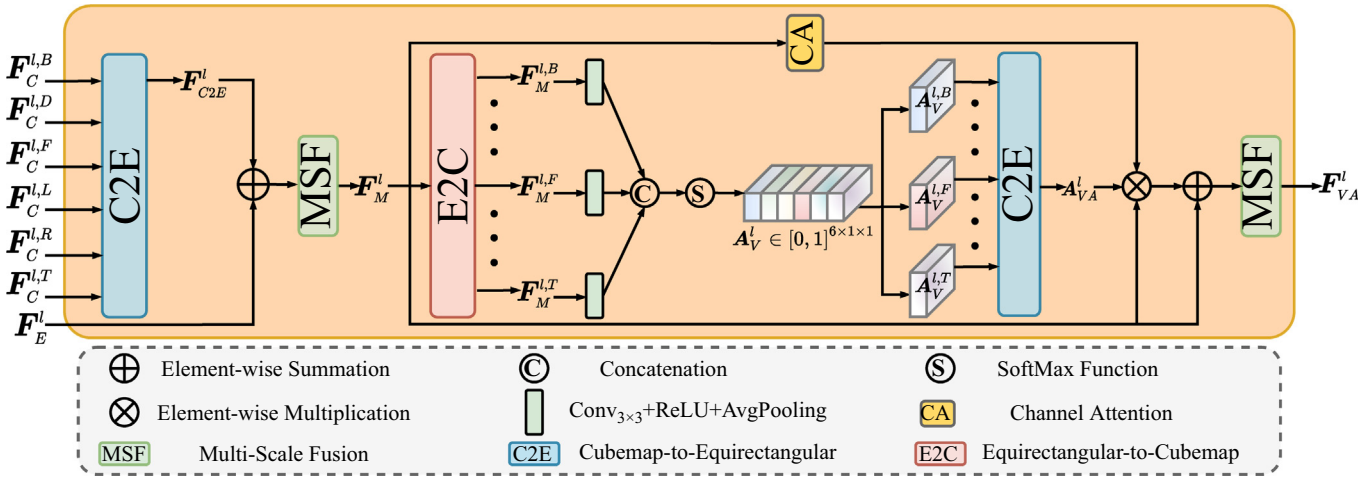

Fig. 4. Structure of the VFAI module. The VFAI modules are equipped at different levels in the VFNet. The input features of the l-th VFAI module in the VFNet are extracted from 360° equirectangular image and its 360° cubemap images by FE network at the l-th level.

and $c \times \frac{h_l}{2} \times \frac{w_l}{4}$, respectively, and the number c of channel is 64. In the VFAI$^{\{l\}}$, we first conduct Cubemap-to-Equirectangular (C2E) projection $\mathcal{P}_{C\to E}(\cdot)$ to obtain the projected equirectangular features $\boldsymbol{F}^l_{C2E}$ which have the same size with $\boldsymbol{F}^l_E$. The element-wise summation "$\oplus$" between $\boldsymbol{F}^l_{C2E}$ and $\boldsymbol{F}^l_E$ is used for integrating these two types of features. For capturing more usable global and local information, we apply Multi-Scale Fusion operation MSF$(\cdot)$, which resembles the atrous spatial pyramid pooling module [48], to the features combined by $\boldsymbol{F}^l_E$ and $\boldsymbol{F}^l_{C2E}$. The operations can be formulated as:

$$\begin{aligned} \boldsymbol{F}^l_{C2E} &= \mathcal{P}_{C\to E}(\boldsymbol{F}^l_C), \\ \boldsymbol{F}^l_M &= \mathrm{MSF}(\boldsymbol{F}^l_E \oplus \boldsymbol{F}^l_{C2E}). \end{aligned} \tag{1}$$

As the structure illustrated in Fig. 4, we use Equirectangular-to-Cubemap (E2C) projection $\mathcal{P}_{E\to C}(\cdot)$ to project the interacted $\boldsymbol{F}^l_M$ to six cubemap features $\{\boldsymbol{F}^{l,f}_M | f \in \{B, D, F, L, R, T\}\}$ in the VFAI$^{\{l\}}$. To extract the dominant features of every cubemap viewport, we apply the 3×3 convolutional layer with ReLU activation function to compress the channel of each $\boldsymbol{F}^{l,f}_M$ to 1, and the spatial-wise global average pooling operation GAP$(\cdot)$ is equipped following the ReLU activation function. Next, we concatenate these extracted features along channel dimension and obtain the viewport attention of each cubemap viewport feature by channel-wise SoftMax activation function *softmax*$(\cdot)$, namely

$$\boldsymbol{A}^l_V = softmax(\mathrm{Cat}(\{\mathrm{GAP}(\mathrm{Conv}^{\rho}_{3\times3}(\boldsymbol{F}^{l,f}_M))\})), \tag{2}$$

where $\mathrm{Conv}^{\rho}_{3\times3}(\cdot)$ is the 3×3 convolutional layer with ReLU activation function $\rho(\cdot)$, Cat$(\cdot)$ is the channel-wise concatenation operation, and $\boldsymbol{A}^l_V \in [0,1]^{6\times1\times1}$ is the attention of six different cubemap viewports.

To expand the attention of six different cubemap viewports to the sizes of cubemap viewport features, we further exploit the broadcasting operation to $\boldsymbol{A}^l_V$ along the height and width of features. These six expanded attention maps $\{\boldsymbol{A}^{l,f}_V\}$ are then re-projected to the equirectangular attention map $\boldsymbol{A}^l_{VA}$ by C2E projection. *i.e.*,

$$\boldsymbol{A}^l_{VA} = \mathcal{P}_{C\to E}(\{\boldsymbol{A}^{l,f}_V\}). \tag{3}$$

We apply Channel Attention [49] operation CA$(\cdot)$ in the VFAI module shown in Fig. 4 and define it as:

$$\mathrm{CA}(\boldsymbol{F}^l_M) = \mathrm{Conv}^{\sigma}_{1\times1}(\mathrm{GAP}(\boldsymbol{F}^l_M) \oplus \mathrm{GMP}(\boldsymbol{F}^l_M)), \tag{4}$$

where $\mathrm{Conv}^{\sigma}_{1\times1}(\cdot)$ is the 1×1 convolutional layer with the sigmoid activation function $\sigma(\cdot)$ and GMP$(\cdot)$ is the spatial-wise global maximum pooling operation.

At the end of the VFAI$^{\{l\}}$, we multiply $\boldsymbol{F}^l_M$ with its channel attention and the equirectangular viewport attention map $\boldsymbol{A}^l_{VA}$, and we also adopt residual connection and MSF operation to obtain the final integrated features $\boldsymbol{F}^l_{VA}$ at different levels. This adaptation process can be computed as:

$$\boldsymbol{F}^l_{VA} = \mathrm{MSF}(\mathrm{CA}(\boldsymbol{F}^l_M) \otimes \boldsymbol{A}^l_{VA} \otimes \boldsymbol{F}^l_M \oplus \boldsymbol{F}^l_M), \tag{5}$$

where "$\otimes$" is the element-wise multiplication.

3.3. DropView perturbation

We adopt several existing perturbation functions, such as DropOut [35], SpatialDropout [39], DropBlock [40], FeatureDrop [25] and FeatureNoise [25], to features of unlabeled ODIs in the VFNet for consistency training, here we also propose a new DropView perturbation for the unlabeled ODIs in the consistency regularization framework shown in Fig. 3. We provide the details of DropView and visualize the processing of the features in this module in Fig. 5 for a clear illustration.

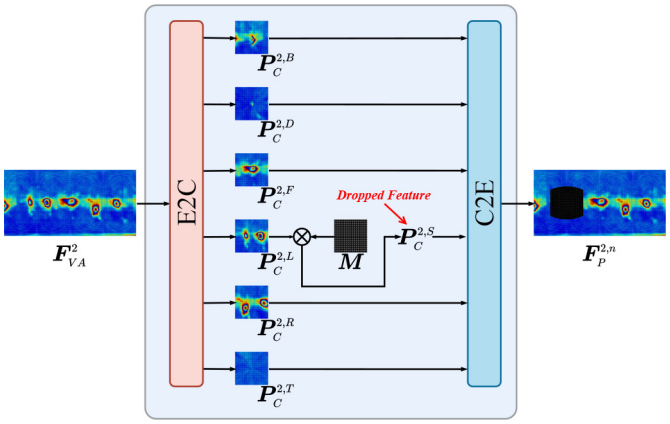

Fig. 5. Visualization of the DropView. In this example, the output feature $\boldsymbol{F}^2_{VA}$ of VFAI$^{\{2\}}$ module is processed by the DropView. By E2C projection, equirectangular feature $\boldsymbol{F}^2_{VA}$ is projected to six cubemap viewport features $\boldsymbol{P}^{2,B}_C, \boldsymbol{P}^{2,D}_C, \boldsymbol{P}^{2,F}_C, \boldsymbol{P}^{2,L}_C, \boldsymbol{P}^{2,R}_C$ and $\boldsymbol{P}^{2,T}_C$. Then, the feature of the left cubemap viewport $\boldsymbol{P}^{2,L}_C$ is randomly dropped out by the random mask $\boldsymbol{M}$ to generate the dropped cubemap viewport feature $\boldsymbol{P}^{2,S}_C$. Finally, $\boldsymbol{F}^{2,n}_P$ represents the feature perturbed by the DropView.

In the proposed DropView, the features generated by VFAI modules $\boldsymbol{F}^l_{VA}$ are projected to $\{\boldsymbol{P}^{l,f}_C\}$ by E2C projection. Then, a random mask $\boldsymbol{M}$, which has the same size as each $\boldsymbol{P}^{l,f}_C$, is generated, and every element of this random mask is zero. Next, the randomly selected feature $\boldsymbol{P}^{l,S}_C$ is dropped by multiplying the random mask $\boldsymbol{M}$, *i.e.*, every element in the dropped $\boldsymbol{P}^{l,S}_C$ is set to zero. The elements in $\boldsymbol{P}^{l,R}_C$ which is the set of unselected features remain their original values. Here, the probability of random selection p_d obeys the discrete uniform distribution $p_d \sim \mathrm{DU}(1,6)$ $(p(d,6) = \frac{1}{6}, d = 1, 2, \ldots, 6)$, *i.e.*, every cubemap viewport feature $\boldsymbol{P}^{l,f}_C$ has equal probability to be dropped. Finally, we re-project the randomly dropped and remaining cubemap viewport features to 360° equirectangular features $\boldsymbol{F}^{l,n}_P$ by C2E projection. Hence, values of the region in equirectangular feature corresponding to the dropped cubemap viewport feature are set to zero as shown in Fig. 5. The above operations can be formulated as:

$$\boldsymbol{F}^{l,n}_P = \mathcal{P}_{C\to E}(\{\boldsymbol{P}^{l,S}_C \otimes \boldsymbol{M}\} \cup \boldsymbol{P}^{l,R}_C). \tag{6}$$

In this way, the spatial correlation of different cubemap viewports of $\boldsymbol{F}^{l,n}_P$ is perturbed before weighted integration by SI Network. Thus, the VFNet can concentrate on general features instead of the features of a certain cubemap viewport, and the representation ability of VFNet can be improved.

3.4. Saliency inference networks

As shown in Fig. 3, MSI network predicts the saliency maps of both labeled and unlabeled ODIs from $\boldsymbol{F}^l_{VA}$ directly. The saliency maps of labeled ODIs predicted by MSI network are under the supervision revealed by labels. Besides, the saliency maps of unlabeled ODIs predicted by MSI network will conduct the consistency training with the results predicted by ASI networks. Notably, the results of ASI networks are predicted from the perturbed features $\boldsymbol{F}^{l,n}_P$. We elaborate SI networks in Fig. 6.

Concretely, we first denote the summation of trainable adaptive scalar weights $\boldsymbol{w}^l$, *i.e.*,

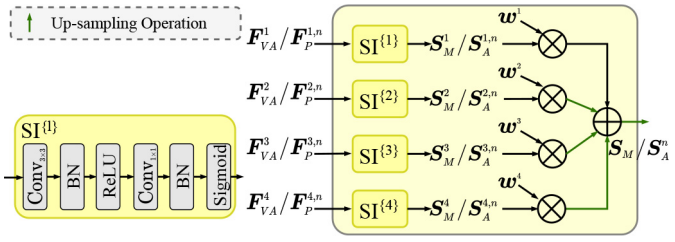

Fig. 6. Details of the SI Network. The structure of MSI and ASI networks is identical, which consists of four SI branches. All SI branches share a common structure and predict the saliency information at different levels of VFNet. After weighting the saliency prediction at different levels by the four trainable weights $\boldsymbol{w}^1, \boldsymbol{w}^2, \boldsymbol{w}^3$ and $\boldsymbol{w}^4$, the final saliency prediction is obtained by up-sampling the weighted saliency information and element-wise summation.

$$\sum_{l=1}^{4} \boldsymbol{w}^l = 1, \tag{7}$$

where $0 \leqslant \boldsymbol{w}^l \leqslant 1$. To extract saliency information at every level, the SI branches $\mathrm{SI}^{\{l\}}(\cdot)$ are then equipped for prediction. Every $\mathrm{SI}^{\{l\}}(\cdot)$ consists of a 3×3 convolutional layer, a ReLU activation function, a 1×1 convolutional layer and a sigmoid activation function sequentially. Besides, two batch normalization layers are inserted between convolutional layers and activation functions, respectively. Here, the amount of output channel of the 1×1 convolutional layer before the sigmoid activation function is one and is consistent with that of the final saliency map. The details of $\mathrm{SI}^{\{l\}}(\cdot)$ have been illustrated in Fig. 6. These operations can be expressed as:

$$\begin{aligned} \boldsymbol{S}_M^l &= \mathrm{SI}^{\{l\}}(\boldsymbol{F}_{VA}^l), \\ \boldsymbol{S}_A^{l,n} &= \mathrm{SI}^{\{l\}}(\boldsymbol{F}_P^{l,n}), \end{aligned} \tag{8}$$

where $\boldsymbol{S}_M^l$ and $\boldsymbol{S}_A^{l,n}$ are the saliency predictions of the l-th level of the MSI network and the n-th ASI network, correspondingly.

At the end of SI networks, the saliency information at different levels is weighted by the trainable weights learned from the data, and the final saliency maps are obtained by adding these weighted saliency maps and up-sampling operation $\mathrm{UP}(\cdot)$, which is

$$\begin{aligned} \boldsymbol{S}_M &= \mathrm{UP}\left(\sum_{l=1}^{4} \mathrm{UP}\left(\boldsymbol{w}^l \otimes \boldsymbol{S}_M^l\right)\right), \\ \boldsymbol{S}_A^n &= \mathrm{UP}\left(\sum_{l=1}^{4} \mathrm{UP}\left(\boldsymbol{w}^l \otimes \boldsymbol{S}_A^{l,n}\right)\right), \end{aligned} \tag{9}$$

where $\boldsymbol{S}_M$ is the final saliency prediction from the MSI network, $\boldsymbol{S}_A^n$ is the final saliency prediction from the n-th ASI network, and $n \in \{1,2,3,4,5,6\}$ indicates the ASI networks which process the $\boldsymbol{F}_P^{l,n}$ perturbed by DropOut [35], SpatialDropout [39], DropBlock [40], FeatureDrop [25], FeatureNoise [25] and our DropView perturbation accordingly.

3.5. Hybrid objective function

1) Bootstrapping loss function of supervision. To handle the training of labeled ODIs in the FE network, VFAI modules and MSI network, we adopt KLD [45], CC [46] and NSS [47] to establish the loss of supervision.

Specifically, the loss $\mathcal{L}_L$ can be expressed as

$$\mathcal{L}_L = \ell_{kld}(\boldsymbol{S}_{M,L}, \boldsymbol{G}^S) - \ell_{cc}(\boldsymbol{S}_{M,L}, \boldsymbol{G}^S) - \ell_{nss}(\boldsymbol{S}_{M,L}, \boldsymbol{G}^F), \tag{10}$$

where $\ell_{kld}(\cdot)$ is the KLD loss function, $\ell_{cc}(\cdot)$ is the CC loss function, $\ell_{nss}(\cdot)$ is the NSS loss function, $\boldsymbol{S}_{M,L}$ is the saliency map of the labeled ODI predicted by the MSI network, and $\boldsymbol{G}^S$ and $\boldsymbol{G}^F$ are fixation density map and binary fixation location map of the corresponding label, respectively.

To keep the training stability on difficult saliency information, we follow the work [43,44] and modify $\mathcal{L}_L$ to the bootstrapped loss, *i.e.*,

$$\mathcal{L}_L^B = \mathrm{I}(\boldsymbol{S}_{M,L} < \lambda) \cdot \mathcal{L}_L, \tag{11}$$

where $\mathrm{I}(\cdot)$ is the indicator function. In this function, the loss of pixel values of the predicted saliency map over λ would be set to zero. Therefore, in every iteration, we only focus on the loss over the pixels with saliency values less than λ. Then less salient pixels are gradually taken into consideration when the training processes. Here, we set λ to 0.7.

2) Unsupervised regularizer. In the training phase, we utilize unlabeled ODIs to improve the representation capability of the network. Since there are no fixation ground truths of the unlabeled data, we exploit unsupervised regularizers to constrain the training between the saliency maps of unlabeled ODIs predicted by MSI network and ASI networks.

Specifically, we combine KLD $\ell_{kld}(\cdot)$ and MSE $\ell_{mse}(\cdot)$ to train the unlabeled data. The unsupervised regularizer can be denoted as:

$$\mathcal{L}_U^n = \ell_{kld}(\boldsymbol{S}_A^n, \boldsymbol{S}_{M,U}) + \ell_{mse}(\boldsymbol{S}_A^n, \boldsymbol{S}_{M,U}), \tag{12}$$

where $\mathcal{L}_U^n$ is the unsupervised regularizer of the n-th ASI network, and $\boldsymbol{S}_{M,U}$ is the saliency map of unlabeled ODIs predicted by MSI network which serves as a pseudo label for the unlabeled data in the consistency regularization. The total unsupervised regularizers $\mathcal{L}_U$ can be expressed as

$$\mathcal{L}_U = \sum_{n=1}^{6} \mathcal{L}_U^n. \tag{13}$$

3) Total objective function. We train our semi-supervised VFNet in an end-to-end manner. For mitigating the initial noisy predictions of the MSI network, we exploit ω_U to weight the unsupervised term, *i.e.*,

$$\mathcal{L}_T = \mathcal{L}_L^B + \omega_U \cdot \mathcal{L}_U. \tag{14}$$

Following [33,25], ω_U ramps up starting from zero along an exponential curve to a fixed value η_U. In this paper, we set η_U to one, and the ramp-up period is the first eight epochs.

4. Experiments

4.1. Datasets

1) Training datasets. We train the proposed VFNet on the following two publicly available datasets.

Salient360!2017 [20] is one of the most popular ODI saliency prediction dataset. It includes 20 ODIs for head movement, 40 ODIs for head and eye movement. The images in this dataset are with various resolutions, and the fixation density labels are blurred by a Gaussian of 3.34° visual angle. For comparing with other methods fairly, we follow previous works [18,17] and adopt these 40 ODIs with head and eye movement visual fixation labels as the labeled data to train our VFNet.

360-SOD [53] includes 500 ODIs for 360° saliency detection and provides binary pixel-wise segmentation ground truths. However, there are no fixation location and density labels in this dataset, and the amount of data in [53] is much smaller than that of the 2D image saliency prediction dataset [15]. We utilize the whole dataset, which contains scene-similar ODIs, as the unlabeled training data of the proposed VFNet method.

2) Evaluation datasets. We evaluate the proposed VFNet and other state-of-the-art methods on the following two publicly available datasets in this paper.

Salient360!2017 [20] also contains 25 ODIs including both "head", and "head and eye movement" eye fixations as evaluation data. For a fair comparison, we adopt these 25 ODIs to evaluate the performance of our VFNet and the compared saliency prediction methods.

Saliency-in-VR [21] consists of 22 ODIs including indoor and outdoor scenes. In this dataset, fixations are captured under the "VR", "VR seated", and "desktop" condition. We follow the compared methods and only employ the fixation label of the VR condition. We utilize this whole dataset for evaluating the performance of various saliency prediction methods. Here, the fixation density labels in this dataset are generated by convolving a Gaussian with 1° visual angle.

4.2. Implementation details

1) Training protocol. We implement the proposed VFNet by PyTorch [54] framework with an NVIDIA Titan RTX GPU. At the training phase, all the training 360° equirectangular images and their corresponding 360° cubemap images are respectively resized to 512×256 and 128×128, and the 360° equirectangular images are augmented by randomly flipping. The initial parameters of the shared FE network are adopted from the pre-trained ResNet-34 model [41] trained on ImageNet [55]. The normal distribution [56] is employed to initialize the parameters of all the newly added convolutional layers. We utilize the Stochastic Gradient Descent (SGD) algorithm for training our VFNet in an end-to-end way.

The batch sizes of labeled and unlabeled data are set to four, and the initial learning rate is set to 0.001. We utilize the momentum of 0.9 and the weight decay of 0.0001. Besides, we adopt the 'poly' policy described in [57] to adjust the learning rate. Finally, the training converges after ~ 40 epochs.

2) Testing protocol. At the testing stage, all the perturbations and corresponding ASI networks are removed and the saliency maps are predicted by MSI network.

4.3. Evaluation metrics

We adopt four most common saliency prediction metrics to evaluate all methods, which include Kullback–Leibler Divergence (KLD) [45], Pearson's Correlation coefficient (CC) [46], Normalized Scanpath Saliency (NSS) [47], and the Area Under the receiver operating characteristic Curve (AUC) variant from Judd et al. [58] (AUC_J). The evaluation tool[1] and default configurations are from Gutiérrez et al. [59].

We adopt the above metrics to quantitatively evaluate the performance of our method and other compared methods from different aspects.

4.4. Comparison with state-of-the-art methods

1) Compared methods. We compare our method with six state-of-the-art ODI saliency prediction methods including BMS360 [23], GBVS360 [23], SalNet360 [19], SalGAN360 [17], MV-SalGAN360 [18] and MC-AEB [52], and four state-of-the-art visual saliency methods for 2D images including MLNet [2], SAM [6], TranSalNet [50] and SalFBNet [51].

Specifically, BMS360 [23], GBVS360 [23] are the ODIs saliency prediction methods extended from traditional 2D saliency prediction methods BMS [4] and GBVS [3]. SalNet360 [19], SalGAN360 [17], MV-SalGAN360 [18] and MC-AEB [52] are the CNN-based ODI saliency prediction methods. The CNN-based methods (SalNet360, SalGAN360 and MV-SalGAN360) are pre-trained on the SALICON [15] dataset which contains 10,000 typical 2D images for training, 5,000 typical 2D images for validation, and 5,000 typical 2D images for testing. Compared with the entire 360-SOD [53] dataset fully adopted as unlabeled data to train our VFNet, the amount of the training data of SALICON (10,000) is much larger than that of the 360-SOD [53] dataset (500). For a fair comparison, we fine-tune the default trained 2D image saliency prediction methods MLNet, ResNet-based SAM, TranSalNet and SalFBNet on the training set of Salient360!2017 [20] dataset, *i.e.*, the 40 ODIs with head and eye movement eye fixation labels. As for the methods for ODIs, we use the saliency maps and the parameters provided by the authors.

2) Quantitative comparison. We conduct quantitative performance comparison of our method and the methods [2,6,17–19,23] mentioned above on Salient360!2017 [20] and Saliency-in-VR [21] dataset in terms of four metrics in Table 1.

Concretely, on the Salient360!2017 dataset, our method shows similar performance to the best methods in terms of KLD, CC, NSS and AUC_J. The SAM method fine-tuned on the Salient360!2017 dataset achieves the best performance in terms of KLD and CC. However, in terms of NSS and AUC_J, it has an obvious percentage gap with our method. SalGAN360 shows the best performance in the remaining metrics, NSS and AUC_J, while our method performs better in terms of KLD and CC, and the percentage gain compared with SalGAN360 reaches 4.6% for KLD and 1.7% for CC. Our semi-supervised method shows balanced performance on these four different metrics. On the Saliency-in-VR dataset, our method achieves the best performance in terms of KLD, CC and NSS. Compared with the second best method on this dataset, the percentage gain of our method reaches 1.5% for KLD, 1.9% for CC, and 12.7% for NSS. Meanwhile, our method performs almost similar to the best method in terms of AUC_J.

Overall, our semi-supervised method shows comparable performance on Salient360!2017 dataset and performs better on the Saliency-in-VR dataset, it achieves the optimal ranking on these two datasets.

3) Visual comparison. In Fig. 7, we present visual comparisons of the investigated methods for several typical scenes in ODI saliency prediction. For the first easy scene, most methods can predict the visually salient regions of the image including our semi-supervised VFNet. For the second scene, our VFNet can predict the visual saliency accurately although the non-uniform eye fixations distribute over different viewports. For example, our saliency map in the 3^{rd} row not only enhances the salient regions in the ODI but also suppresses the regions with few eye fixations. For the last scene, the eye fixations are distributed in multiple viewports. Our semi-supervised method can still effectively highlight the visually salient regions as those fully-supervised methods. In addition, in the top and bottom regions of 360° equirectangular images, our method can suppress the background information stably, such as the paintings and display case in the 6^{th} row.

4.5. Ablation studies

In this subsection, we provide comprehensive ablation studies of our VFNet on Salient360!2017 [20] and Saliency-in-VR [21] datasets, and verify the contribution of every key component in our method. Specifically, we investigate 1) the importance of the VFAI module, 2) the necessity of the DropView in the consistency training, 3) the effectiveness of the multi-level form of perturbation, and 4) the usefulness of bootstrapping hybrid loss of supervision. At each time, we change one component and re-train the corresponding variants with the same training set and hyper-parameters in Sections 4.1 and 4.2.

1. The importance of the VFAI module. To study the importance of VFAI module in VFNet, we explore two straightforward variants as baselines: 1) replacing VFAI module with element-wise summation and convolution operations (*i.e.*, *AC*) to fuse the equirectangular and projected equirectangular features, and 2) removing the viewport feature adaptive integration operation but maintaining Multi-Scale Fusion (MSF) and Channel Attention (CA) (*i.e.*, *MC*) in VFAI module.

As shown in Table 2, the performance of both variants is severely damaged. Furthermore, the comparison between the proposed VFNet and the variant *MC* indicates that cubemap viewport feature adaptive integration is critical in the VFAI module which fits the non-uniform distribution of eye fixations over ODIs by adaptively and further improves the performance of VFNet on all metrics on both two datasets.

[1] https://salient360.ls2n.fr/.

Table 1
Quantitative performance comparison of our method and other state-of-the-art methods on Salient360!2017 [20] and Saliency-in-VR [21] datasets. ↑ (↓) indicates the larger (smaller) is better. The top three results are marked in **red**, **blue** and **green** accordingly.

Models	Salient360!2017 [20]				Saliency-in-VR [21]				AvgRank ↓
	KLD ↓	CC ↑	NSS ↑	AUC_J ↑	KLD ↓	CC ↑	NSS ↑	AUC_J ↑	
$MLNet_{16}$ [2]	0.653	0.583	0.585	0.664	1.795	0.490	1.854	0.831	7.3
SAM_{18} [6]	**0.341**	**0.684**	0.803	0.707	1.165	0.499	1.795	0.848	4.0
$TranSalNet_{22}$ [50]	0.664	0.398	0.779	0.711	1.398	0.402	1.433	0.821	8.6
$SalFBNet_{22}$ [51]	2.087	0.483	0.774	0.676	2.473	0.430	1.559	0.804	8.8
$BMS360_{18}$ [23]	0.531	0.494	0.868	**0.735**	1.224	0.441	1.596	0.849	5.0
$GBVS360_{18}$ [23]	0.554	0.442	0.739	0.703	1.321	0.408	1.558	0.834	7.8
$SalNet360_{18}$ [19]	0.473	0.534	0.802	0.714	1.354	0.436	1.539	0.831	6.5
$SalGAN360_{18}$ [17]	**0.407**	0.634	**0.965**	**0.746**	**1.114**	**0.518**	**1.912**	**0.863**	**2.3**
$MV\text{-}SalGAN360_{21}$ [18]	0.450	**0.675**	**0.922**	0.732	**1.072**	**0.517**	**1.900**	**0.864**	**2.6**
$MC\text{-}AEB_{22}$ [52]	0.527	0.591	0.887	0.719	-	-	-	-	-
VFNet (Ours)	**0.361**	**0.651**	**0.921**	**0.733**	**1.057**	**0.537**	**2.039**	**0.859**	**2.1**

2. The necessity of the DropView in consistency training. To validate the necessity of the DropView in consistency training, we remove the DropView and only use DropOut [35], SpatialDropout [39], DropBlock [40], FeatureDrop [25] and FeatureNoise [25] as the perturbations.

The performance of *w/o DV* shown in Table 2 illustrates that the proposed DropView can further improve the performance of VFNet on both two datasets by perturbing the spatial correlation of different viewports of ODIs.

3. The effectiveness of the multi-level form of perturbation. To explore the effectiveness of injecting a given type of perturbation module into input features of the given ASI network at different levels, we offer a variant which conducts trainable weighted summation described in Section 3.4 on the output features of different levels in VFAI modules and then perturbs these added features before the ASI networks, *i.e.*, *w/o MLP*. From Table 2, we see that the performance is degraded when the multi-level form of perturbation is removed. It confirms perturbing multi-level features by a given perturbation function in one specific ASI network is reasonable.

4. The usefulness of bootstrapping hybrid loss of supervision. In MSI network, we adopt bootstrapping hybrid loss function for training with the labeled data. To investigate its usefulness, we conduct a normal form of the hybrid loss, *i.e.*, *w/o BH*, in MSI network and train the variant. As reported in Table 2, the performance degradation of *w/o BH* validates that the bootstrapping hybrid loss function policy benefits predicting visual saliency in ODIs. Besides, we also explore the labeled data only for training our VFNet, *i.e.*, we only maintain the FE network, VFAI modules and MSI network, and then train this variant (*w/o UL*) with the 40 360° equirectangular images in Salient360!2017 dataset. As shown in Table 2, the performance of *w/o UL* is also degraded on Salient360!2017 and Saliency-in-VR datasets. Specifically, we observe that *w/o UL* has a smaller percentage gap with VFNet (Ours) on Salient360!2017 dataset than on Saliency-in-VR dataset. This demonstrates training with a small amount of training data may cause overfitting on a given dataset and damage the generalizability of method cross datasets.

5. Conclusion

In this paper, we propose a novel and effective semi-supervised Viewport Feature Network (VFNet) for omnidirectional image (ODI) saliency prediction. VFNet is equipped with consistency regularization framework for training with the labeled and unlabeled data in an end-to-end way. In particular, the proposed DropView provides a new perturbation form, which is imposed on the intermediate features of unlabeled ODIs, for enhancing the representation capacity of the VFNet. Furthermore, the proposed Viewport Feature Adaptive Integration (VFAI) module is a vital medium for feature encoding and saliency inference in the VFNet. This module is in charge of interacting 360° equirectangular and cubemap features and adaptively integrating the features based on the distribution of eye fixations in omnidirectional scenes. Comprehensive experiments, including comparison analyses and ablation studies, demonstrate that our VFNet is competitive with

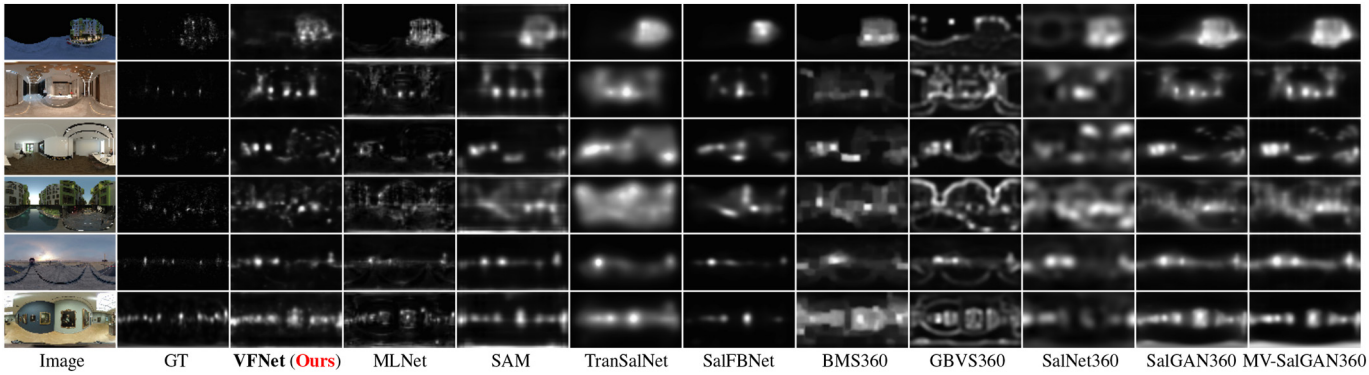

Fig. 7. Visual comparisons of our VFNet with nine latest ODI saliency prediction methods, including MLNet [2], SAM [6], TranSalNet [50], SalFBNet [51], BMS360 [23], GBVS360 [23], SalNet360 [19], SalGAN360 [17], and MV-SalGAN360 [18] on Salient360!2017 [20] and Saliency-in-VR [21] datasets.

Table 2
Ablation analyses for the importance of VFAI module, the necessity of DropView in consistency training, the effectiveness of the multi-level form of perturbation and the usefulness of bootstrapping hybrid loss of supervision. The best result in each column is in **bold**.

Models	Salient360!2017 [20]				Saliency-in-VR [21]			
	KLD↓	CC↑	NSS↑	AUC_J↑	KLD↓	CC↑	NSS↑	AUC_J↑
VFNet (Ours)	**0.361**	**0.651**	**0.921**	**0.733**	**1.057**	**0.537**	**2.039**	**0.859**
AC	0.418	0.631	0.869	0.721	1.193	0.500	1.840	0.850
MC	0.371	0.642	0.872	0.718	1.126	0.503	1.832	0.856
w/o DV	0.363	**0.651**	0.874	0.722	1.137	0.511	1.860	0.857
w/o MLP	0.377	0.639	0.858	0.719	1.110	0.512	1.893	0.857
w/o BH	0.394	0.623	0.860	0.714	1.159	0.499	1.870	0.849
w/o UL	0.371	0.649	**0.921**	0.728	1.124	0.503	1.809	0.855

state-of-the-art ODI saliency prediction methods based on fully-supervised learning, and shows a balanced performance on different datasets.

CRediT authorship contribution statement

Mengke Huang: Conceptualization, Methodology, Software, Validation, Writing-original-draft. **Gongyang Li:** Conceptualization, Investigation, Writing-review-editing. **Zhi Liu:** Conceptualization, Writing-review-editing, Supervision, Resources. **Yong Wu:** Formal-analysis, Visualization. **Chen Gong:** Writing-review-editing. **Linchao Zhu:** Writing-review-editing. **Yi Yang:** Resources, Supervision.

Data availability

Data will be made available on request.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 62171269, and in part by the China Postdoctoral Science Foundation under Grant 2022M722037.

References

[1] L. Itti, C. Koch, E. Niebur, A model of saliency-based visual attention for rapid scene analysis, IEEE Trans. Pattern Anal. Mach. Intell. 20 (11) (1998) 1254–1259.
[2] M. Cornia, L. Baraldi, G. Serra, R. Cucchiara, A deep multi-level network for saliency prediction, in: Proc. Int. Conf. Pattern Recognit. (ICPR), 2016, pp. 3488–3493.
[3] J. Harel, C. Koch, P. Perona, Graph-based visual saliency, in: Proc. Annu. Conf. Neural Inf. Process. Syst. (NeurIPS), 2006, pp. 545–552.
[4] J. Zhang, S. Sclaroff, Saliency Detection: A boolean map approach, in: Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2013, pp. 153–160.
[5] E. Vig, M. Dorr, D. Cox, Large-scale optimization of hierarchical features for saliency prediction in natural images, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2014, pp. 2798–2805.
[6] M. Cornia, L. Baraldi, G. Serra, R. Cucchiara, Predicting human eye fixations via an lstm-based saliency attentive model, IEEE Trans. Image Process. 27 (10) (2018) 5142–5154.
[7] Z. Wang, Z. Liu, W. Wei, H. Duan, SalED: Saliency prediction with a pithy encoder-decoder architecture sensing local and global information, Image Vis. Comput. 109 (2021), 104149.
[8] X. Luo, Z. Liu, W. Wei, L. Ye, T. Zhang, L. Xu, J. Wang, Few-shot personalized saliency prediction using meta-learning, Image Vis. Comput. 124 (2022), 104491.
[9] M. Meng, J. Wei, J. Wu, Learning multi-part attention neural network for zero-shot classification, IEEE Trans. Cogn. Develop. Syst. 14 (2) (2022) 414–423.
[10] H. Hadizadeh, I.V. Bajić, Saliency-aware video compression, IEEE Trans. Image Process. 23 (1) (2014) 19–33.
[11] F. Zhao, Q. Kong, Y. Zeng, B. Xu, A brain-inspired visual fear responses model for uav emergent obstacle dodging, IEEE Trans. Cogn. Develop. Syst. 12 (1) (2020) 124–132.
[12] G. Li, Z. Liu, R. Shi, Z. Hu, W. Wei, Y. Wu, M. Huang, H. Ling, Personal fixations-based object segmentation with object localization and boundary preservation, IEEE Trans. Image Process. 30 (2021) 1461–1475.
[13] G. Li, Z. Liu, R. Shi, W. Wei, Constrained fixation point based segmentation via deep neural network, Neurocomputing 368 (2019) 180–187.
[14] M. Cornia, L. Baraldi, G. Serra, R. Cucchiara, Paying More Attention to Saliency: Image captioning with saliency and context attention, ACM Trans. Multimedia Comput. Commun. Appl. 14 (2) (2018) 1–21.
[15] M. Jiang, S. Huang, J. Duan, Q. Zhao, SALICON: Saliency in context, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2015, pp. 1072–1080.
[16] A. Borji, L. Itti, CAT2000: A large scale fixation dataset for boosting saliency research, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshop, 2015.
[17] F.-Y. Chao, L. Zhang, W. Hamidouche, O. Déforges, SalGAN360: Visual saliency prediction on 360 degree images with generative adversarial networks, in: Proc. Int. Conf. Multimedia Expo. (ICME) Workshop, 2018, pp. 1–4.
[18] F.-Y. Chao, L. Zhang, W. Hamidouche, O. Déforges, A multi-fov viewport-based visual saliency model using adaptive weighting losses for 360° images, IEEE Trans. Multimedia 23 (2021) 1811–1826.
[19] R. Monroy, S. Lutz, T. Chalasani, A. Smolic, SalNet360: Saliency maps for omni-directional images with cnn, Signal Process.-Image Commun. 69 (2018) 26–34.
[20] Y. Rai, J. Gutiérrez, P. Le Callet, A dataset of head and eye movements for 360 degree images, in: Proc. ACM Multimedia Syst., 2017, pp. 205–210.
[21] V. Sitzmann, A. Serrano, A. Pavel, M. Agrawala, D. Gutierrez, B. Masia, G. Wetzstein, Saliency in VR: How do people explore virtual environments? IEEE Trans. Vis. Comput. Graph. 24 (4) (2018) 1633–1642.
[22] T. Maugey, O. Le Meur, Z. Liu, Saliency-based navigation in omnidirectional image, in: Proc. Int. Workshop Multimedia Signal Process. (MMSP), 2017, pp. 1–6.
[23] P. Lebreton, A. Raake, BMS360, ProSal: Extending existing saliency prediction models from 2d to omnidirectional images, Signal Process.-Image Commun. 69 (2018) 69–78.
[24] M. Startsev, M. Dorr, 360-aware saliency estimation with conventional image saliency predictors, Signal Process.-Image Commun. 69 (2018) 43–52.
[25] Y. Ouali, C. Hudelot, M. Tami, Semi-supervised semantic segmentation with cross-consistency training, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2020, pp. 12671–12681.
[26] A. De Abreu, C. Ozcinar, A. Smolic, Look around you: Saliency maps for omnidirectional images in vr applications, in: Proc. Int. Conf. on Qual. Multimedia Experience (QoMEX), 2017, pp. 1–6.
[27] J. Pan, C. Canton Ferrer, K. McGuinness, N.E. O'Connor, J. Torres, E. Sayrol, X. Giro-i-Nieto, SalGAN: Visual saliency prediction with generative adversarial networks, arXiv e-prints arXiv:1701.01081.
[28] L. Zhu, Y. Yang, Label independent memory for semi-supervised few-shot video classification, IEEE Trans. Pattern Anal. Mach. Intell. 44 (1) (2022) 273–285.
[29] Y. Chen, J. Huang, Z. Zhu, X. Long, G. Qingyi, Boosting semi-supervised face recognition with raw faces, Image Vis. Comput. 125 (2022), 104512.
[30] J. Zhang, Y. Peng, SSDH: Semi-supervised deep hashing for large scale image retrieval, IEEE Trans. Circuits Syst. Video Technol. 29 (1) (2019) 212–225.
[31] L. Qi, L. Wang, J. Huo, Y. Shi, Y. Gao, Progressive cross-camera soft-label learning for semi-supervised person re-identification, IEEE Trans. Circuits Syst. Video Technol. 30 (9) (2020) 2815–2829.
[32] C. Jia, Z. Ding, Y. Kong, Y. Fu, Semi-supervised cross-modality action recognition by latent tensor transfer learning, IEEE Trans. Circuits Syst. Video Technol. 30 (9) (2020) 2801–2814.
[33] S. Laine, T. Aila, Temporal ensembling for semi-supervised learning, in: Proc. Int. Conf. Learn. Represent. (ICLR), 2017.
[34] A. Tarvainen, H. Valpola, Mean teachers are better role models: weight-averaged consistency targets improve semi-supervised deep learning results, in: Proc. Annu. Conf. Neural Inf. Process. Syst. (NeurIPS), 2017, pp. 1195–1204.
[35] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (56) (2014) 1929–1958.
[36] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proc. Annu. Conf. Neural Inf. Process. Syst. (NeurIPS), 2012, pp. 1097–1105.
[37] L. Wan, M. Zeiler, S. Zhang, Y.L. Cun, R. Fergus, Regularization of neural networks using dropconnect, in: Proc. Int. Conf. Mach. Learn. (ICML), 2013, pp. 1058–1066.
[38] G. Larsson, M. Maire, G. Shakhnarovich, FractalNet: Ultra-deep neural networks without residuals, in: Proc. Int. Conf. Learn. Represent. (ICLR), 2017.
[39] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, C. Bregler, Efficient object localization using convolutional networks, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2015, pp. 648–656.
[40] G. Ghiasi, T.-Y. Lin, Q.V. Le, DropBlock: A regularization method for convolutional networks, in: Proc. Annu. Conf. Neural Inf. Process. Syst. (NeurIPS), 2018, pp. 10750–10760.
[41] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 770–778.
[42] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2017, pp. 936–944.
[43] T. Pohlen, A. Hermans, M. Mathias, B. Leibe, Full-resolution residual networks for semantic segmentation in street scenes, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2017, pp. 3309–3318.
[44] Q. Xie, Z. Dai, E. Hovy, T. Luong, Q. Le, Unsupervised data augmentation for consistency training, in: Proc. Annu. Conf. Neural Inf. Process. Syst. (NeurIPS), 2020, pp. 6256–6268.

[45] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, F. Durand, What do different evaluation metrics tell us about saliency models? IEEE Trans. Pattern Anal. Mach. Intell 41 (3) (2019) 740–757.
[46] O. Le Meur, P. Le Callet, D. Barba, Predicting visual fixations on video based on low-level visual features, Vision Res. 47 (19) (2007) 2483–2498.
[47] R.J. Peters, A. Iyer, L. Itti, C. Koch, Components of bottom-up gaze allocation in natural images, Vision Res. 45 (18) (2005) 2397–2416.
[48] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in: Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 833–851.
[49] S. Woo, J. Park, J.-Y. Lee, I.S. Kweon, CBAM: Convolutional block attention module, in: Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 3–19.
[50] J. Lou, H. Lin, D. Marshall, D. Saupe, H. Liu, TranSalNet: Towards perceptually relevant visual saliency prediction, Neurocomputing 494 (2022) 455–467.
[51] G. Ding, N. Imamoğlu, A. Caglayan, M. Murakawa, R. Nakamura, SalFBNet: Learning pseudo-saliency distribution via feedback convolutional networks, Image Vis. Comput. 120 (2022), 104395.
[52] R. Zhang, C. Chen, J. Zhang, J. Peng, A.M.T. Alzbier, 360-degree visual saliency detection based on fast-mapped convolution and adaptive equator-bias perception, Vis. Comput. (2022) 1–18, https://doi.org/10.1007/s00371-021-02395-w.
[53] J. Li, J. Su, C. Xia, Y. Tian, Distortion-adaptive salient object detection in 360° omnidirectional images, IEEE J. Sel. Top. Signal Process. 14 (1) (2020) 38–48.
[54] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An imperative style, high-performance deep learning library, in: Proc. Annu. Conf. Neural Inf. Process. Syst. (NeurIPS), 2019, pp. 8026–8037.
[55] J. Deng, W. Dong, R. Socher, L. Li, K. Li, F.-F. Li, ImageNet: A large-scale hierarchical image database, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2009, pp. 248–255.
[56] K. He, X. Zhang, S. Ren, J. Sun, Delving Deep into Rectifiers: Surpassing human-level performance on imagenet classification, in: Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2015, pp. 1026–1034.
[57] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, IEEE Trans. Pattern Anal. Mach. Intell 40 (4) (2018) 834–848.
[58] T. Judd, K. Ehinger, F. Durand, A. Torralba, Learning to predict where humans look, in: Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2009, pp. 2106–2113.
[59] J. Gutiérrez, E.J. David, A. Coutrot, M.P. Da Silva, P.L. Callet, Introducing UN Salient360! Benchmark: A platform for evaluating visual attention models for 360° contents, in: Proc. Int. Conf. Qual. Multimedia Experience (QoMEX), 2018, pp. 1–3.