# 360° Image Saliency Prediction by Embedding Self-Supervised Proxy Task

Zizhuang Zou, Mao Ye, Shuai Li, Xue Li, Frédéric Dufaux

# 360° Image Saliency Prediction by Embedding Self-Supervised Proxy Task

Zizhuang Zou, Mao Ye*, *Member, IEEE,* Shuai Li, *Member, IEEE,* Xue Li, *Member, IEEE*, and Frederic Dufaux, *Fellow, IEEE*

***Abstract*—The development of Metaverse industry produces many 360° images and videos. Transmitting these images or videos efficiently is the key to success of Metaverse. Since the subject's field of view is limited in Metaverse, from the perception perspective, bit rates can be saved by focusing video encoding on salient regions. On different ways of handling 360° image projections, the existing works either consider combining local and global projections or just use only global projection for saliency prediction, which results in slow detection speed or low accuracy. In this work, we address this problem by Embedding a self-supervised Proxy task in the Saliency prediction Network, dubbed as EPSNet. The main architecture follows an autoencoder with an encoder for feature extraction and a decoder for saliency prediction. The proxy task is combined with the encoder to enforce it to learn local and global information. It is designed to find the location of a certain local projection in the global projection via self-supervised learning. A cross-attention fusion mechanism is used to fuse the global and local features for location prediction. Then, the decoder is trained based on the sole global projection. In this way, the time-consuming local-global feature fusion is placed in the training stage only. Experiments on public dataset show that our method has achieved satisfactory results in terms of inference speed and accuracy. The dataset and code are available at https://github.com/zzz0326/EPSNet.**

***Index Terms*—360° image, saliency prediction, proxy task.**

## I. Introduction

OMNI-DIRECTIONAL (360°) image saliency detection are very useful for 360° image or video perception oriented transmission which will save bit rates in perception perspective [1], [2], [3], [4]. An accurate and quick saliency detection also means that a deeper understanding of the 360° image data which leads to advances in object detection [5], [6], semantic segmentation [7], [8], viewport prediction [9], [10], [11], etc. 360° saliency detection is an extremely challenging task because the subjects can only observe the area within a limited field of view when using the head-mounted displays due to the panoramic information it contains. As a result, when predicting the 360° image saliency, head and eye movement prediction is required [12].

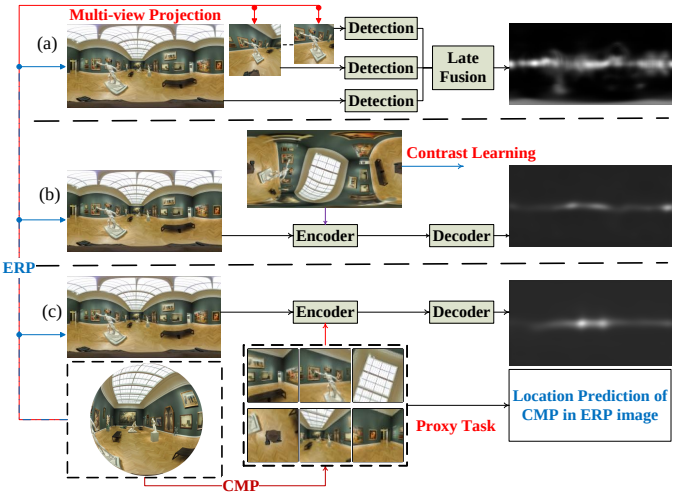


Fig. 1. Comparison between the previous method and our method **EPSNet**. (a) *Multiple projection fusion* first do multiple local or global projections, and then fuse the saliency detection results. (b) *Self-supervised learning approach* adapts contrast learning to train the encoder. (c) Our approach embeds a proxy task to train feature encoder.

There are generally three ways to predict 360° image saliency: *direct prediction*, *multiple projection fusion* and *self-supervised learning based method*. *Direct prediction* simply uses the 2D image saliency detection models [13], [14] to the global projected 2D image from the 360° image. Although the saliency detection methods for 2D image have been studied for a long time [15], these models cannot be directly applied to 360° image because of the discontinuity and distortion characteristics of the projected 2D image mentioned in [16]. Also the panoramic information requires additional head motion prediction, resulting in only limited performance.

The second row of methods uses *multiple projection fusion* of 360° image to combine local and global information to address the discontinuities and distortions [17], [18], [19], which is shown in Fig.1(a). Usually, 360° image is projected to a few local and global view 2D images, and then 2D image saliency detection method is applied to obtain the salient areas which are lately fused to get the final saliency regions. This type of method combines the experience of previous works on 2D saliency detection and the characteristics of 360° image which provides a decent performance. However, this category of methods need to first construct the 2D stored 360° image

This work was supported in part by the National Key R&D Program of China (2018YFE0203900) and National Natural Science Foundation of China (62276048).

Zizhuang Zou and Mao Ye are with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, P.R. China (e-mail:zouzizhuang@163.com, cvlab.uestc@gmail.com).

Shuai Li is with the School of Control Science and Engineering, Shandong University, Jinan 250000, P.R. China (e-mail:shuaili@sdu.edu.cn).

Xue Li is with the School of Information Technology and Electrical Engineering, University of Queensland, Brisbane, QLD 4072, Australia (e-mail: xueli@itee.uq.edu.au).

Frederic Dufaux is with Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des signaux et systèmes, France (email: frederic.dufaux@l2s.centralesupelec.fr).

*corresponding author

into a sphere, project it into a few plane image and then fuse predictions, resulting in a slow speed.

The last kind of methods utilizes the data characteristics of 360° images to design a *self-supervised learning method* [16]. With the limited saliency labeled omni-directional images with saliency, some data augmentation or 2D datasets have to be used to train the model, which does not make good use of the vast unlabeled 360° images. The last category of methods takes advantages of these unlabeled 360° images by using an autoencoder-style architecture consisting of an encoder and a decoder as shown in Fig.1(b). The encoder is trained with easily collected unlabeled 360° images by contrastive learning, i.e., the features of global projections of the same 360° image at different angles are close to each other and the projection features of different 360° images are far away. Then it is processed by a decoder with 2D image saliency model to promote inference speed. However, the existing work only enhances feature representations in the global field of view, **ignoring the local information similar to the user's view,** resulting in rather poor performance.

From the above analysis, it is natural to consider combining the approaches of *self-supervised learning* and *multiple projection fusion*. However, this combination is not trivial, because the simple ensemble of multiple projection approach further increases the inference time. As shown in Fig.1(a), the information of local projection is also contained in the global projection. To solve the above mentioned paradox, we propose to extract both global and local features from global projection. In this way, the prediction accuracy can be improved and redundant computation caused by multiple projection fusion can also be avoided.

Based on the above motivation and idea, we develop a new approach which Embeds a self-supervised Proxy task in the Saliency prediction Network, named as **EPSNet** as shown in Fig.1(c). Our EPSNet consists of an encoder and a decoder as the backbone. A 360° image is first projected by Equi-Rectangular Projection (ERP) and Cube Map Projection (CMP) [20] to obtain the global projection (ERP images) and local projection (CMP images), respectively. The encoder extracts the global features of ERP, with a proxy task, named as FindCMP. It aims to find the location of one face of CMP (local) in ERP (global) which is known in projections. Then a cross attention module fuses the local and global information to interact and learn from each other. The proxy task FindCMP enforces the encoder to learn both local and global features from an ERP image. In the end, a decoder is used to obtain the final saliency prediction. It is worth noting that the CMP projection is only used in the training process to supervise the feature learning from ERP, and not used in the inference, thus reducing the inference complexity.

Our contributions are in three-folds: (1) A new framework for 360° image saliency detection is proposed. It is the first one to use a proxy task to assist the 360° image saliency detection network learning local and global features, which only exists at training phase. Our technical design of the proxy task improves the prediction accuracy without reducing the inference speed. (2) We proposed a new cross attention feature fusion scheme for the proxy task. By predicting the category of six faces of a CMP image, it can obtain implicit global features to interact with the encoder to learn better features. (3) Despite its simple design, extensive experiments on public dataset prove that our **EPSNet** outperforms a wide variety of the state-of-the-art methods in terms of inference speed and accuracy.

## II. Related Work

### A. 360° Image Saliency Detection

As we mentioned before, there are three routes for 360° image saliency detection. The first approach extends 2D saliency model to the 2D global projection from 360° image directly. For example, SaltiNet [21] uses a 2D U-shaped structure to obtain 360° saliency results. As denoted by [18], 2D saliency model has strong center bias located in the center area of projected image, but 360° image does not obey this property. In addition, the discontinuity and distortion of projection also reduce the accuracy of 2D model.

The second approach *multiple projection fusion* usually uses the traditional 2D saliency model to the small field images obtained after projection, and then performs late fusion to get the final results. For example, SalGAN360 [17] uses fine-tunned 2D SalGan [13] to obtain the CMP and ERP saliency maps and then fuse them. ATSAL [19] divides the CMP surface into two categories: equator and pole, and then fuses saliency based on attentions. It is noteworthy that other state-of-the-art attention models can also be used in this kind of fusion. For example, HAN [22] constructs the attention with interdependencies between different layers, channels, and locations; SRGAT [23] divides the image into small pieces and uses the graph attention to relevance them. MV-SalGAN360 [18] introduces local fields of view with various sizes, and determine the learned weights for the corresponding positions according to the pixel density. In [24] expands the original CMP face is extended with the surrounding information so that the field of view of a single CMP face is greater than 90 degrees, which eventually makes the segmented face intersects at the boundary to construct a more complete sphere. All in all, the projection operation and multi-view prediction fusion slow down inference speed.

The lack of labeled 360° saliency samples promotes the last approach based on *self-supervised learning* method. Rethink+ [16] uses spherical rotation to get global projections from different angles to construct positive and negative sample pairs, and train the feature encoder through NCE loss [25]. Then, the features by self-supervised learning are used to train a decoder for saliency detection. This route brings fast single-view inference. But it does not use local features in the training process, so the accuracy is limited.

### B. Proxy Task Learning

Proxy task learning means building another task whose training sample and labels are based on the original training data. Training the proxy task model can learn some features that are helpful to the main task. For example, RotNet [19] rotates the original image to get 0, 90, 180, 270 degree rotated images, by the proxy task of classifying these 4 categories,

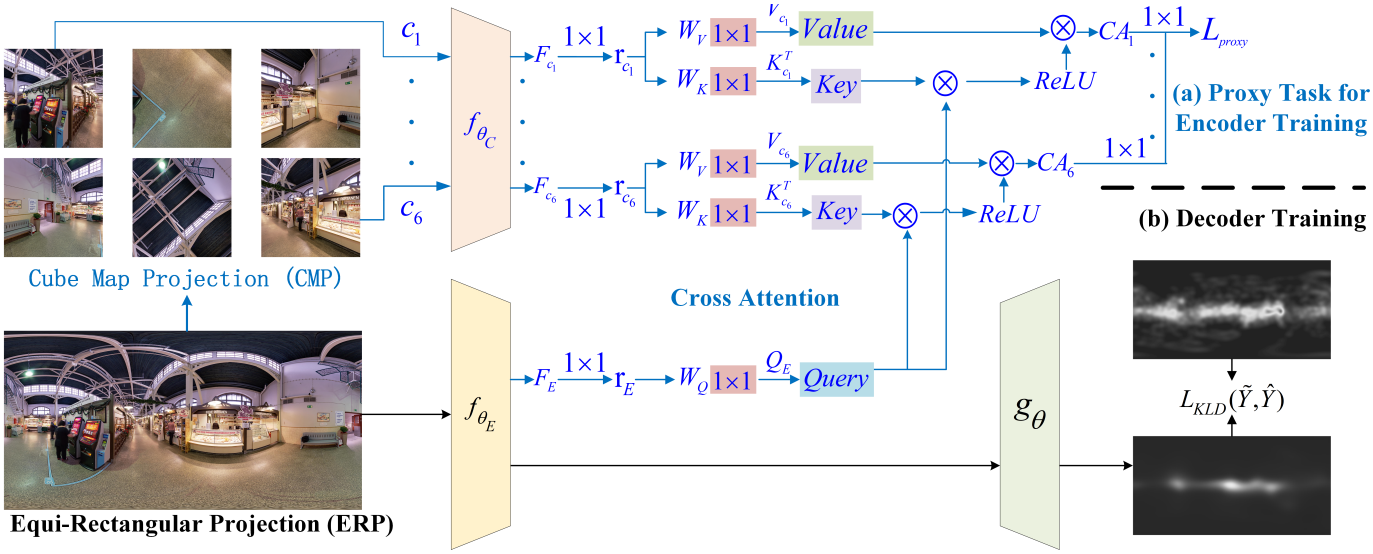


Fig. 2. Overview of our method. The training process of our model consists of two stages. (a) Encoder training based on proxy task. The encoder is trained using a proxy task that combines local and global features (blue). (b) Decoder training based on the trained encoder. KLD loss is used for training while the encoder parameters are frozen (black).

the additional enhanced features can be applied to image classification and object detection. In semantic segmentation area, [26] designs a proxy task of restoring grayscale images to color images, allowing the model to learn features to help image segmentation. [27] designs a series of tasks based on relationships between 2D images obtained by 360° images, and applied the learned features to all of the above 2D tasks.

These tasks are all designed for 2D related tasks and cannot be directly applied to 360° images because the objects in its projection format ERP have inequable geometric representations at different latitudes. Therefore, we propose a new proxy task that combines ERP with CMP images that are similar to the subject's field of view to help the model understand the geometric distortion in ERP format image.

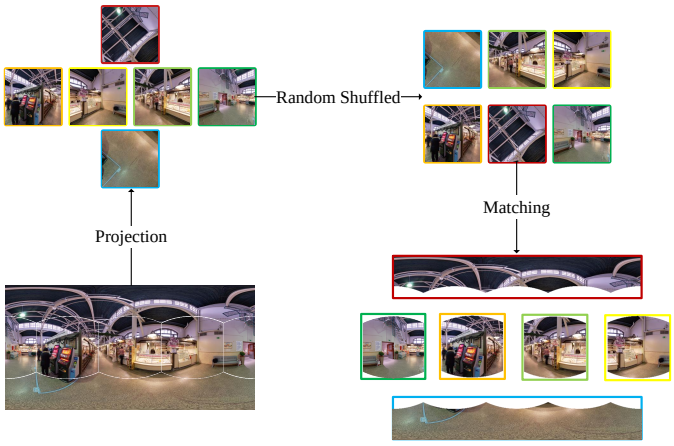


Fig. 3. Framework of the proposed FindCMP.

## III. The Proposed Method

**Problem statement.** Suppose that there exits a labeled dataset $D_s = \{(E_s^i, \hat{Y}_s^i)\}_{i=1}^{N_s}$ where $N_s$ represents the total number of labeled 360° images, $E_s^i$ is the $i$-th 360° image stored in ERP format and the corresponding label $\hat{Y}_s^i$ denotes its ground-truth saliency area. There also exists an unlabeled dataset $D_u = \{E_u^j\}_{j=1}^{N_u}$ where $N_u$ is the total number of unlabeled 360° images stored in ERP format. Our goal is to design a model which can efficiently and accurately predict the saliency areas of a 360° image based on the labeled dataset $D_s$ and unlabeled dataset $D_u$.

**Overview.** The proposed **EPSNet** is a two-stage method. As shown in Fig.2, the encoder $f_{\theta_E}$ and decoder $g_\theta$ are trained sequentially. First, a proxy task is employed to train the feature encoder based on the dataset $D_u$(Fig.2(a)), then the decoder is trained by freezing the encoder parameters based on the dataset $D_s$ (Fig.2(b)). At the inference phase, the proxy task is no longer needed; the saliency is predicted based on the ERP format input of a 360° image.

### A. Training Encoder with Proxy Task

Fig.3 demonstrates the process of the proposed proxy task. The unlabeled training dataset $D_u$ for proxy task is used for constructing the relationship between one face of CMP format image and its location in the corresponding ERP format image. It makes that $f_{\theta_E}$ can learn local feature similar to the user's field of view and the geometric characteristics contained in ERP. First, one ERP image is selected from dataset $D_u$ and converted to CMP format. The ERP format image is denoted as $E \in \mathcal{R}^{3\times w_e \times h_e}$, where $w_e$ and $h_e$ represent image width and height respectively. Each CMP has 6 faces denoted as $c_i \in \mathcal{R}^{3\times w_c \times h_c}$ where $i \in \{1, \cdots, 6\}$, $w_c$ and $h_c$ represent the width and height of one face of CMP image respectively. We divide the process of transforming ERP into CMP into two steps. The first step is placing ERP into a cube with the same side length as the sphere's diameter. Then the perspective projection is performed as $\mathcal{T}(E) = \sum_{i=1}^{6} C_i$ where $\mathcal{T}$ is the projection, and $C$ represents a CMP group obtained by $E$.

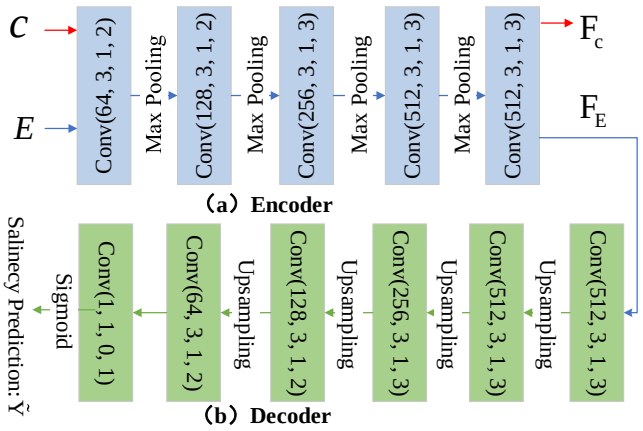


Fig. 4. The architecture of feature encoder and decoder for saliency prediction. The stride of all convolution layers is set to 1, and all but the last layer are activated by ReLU function.

Without rotating the sphere, the position of each CMP face on the ERP is fixed, and we can use this property to label the locations, i.e. assign each face $c_i$ with the label $\hat{P}$ of $1, 2, \cdots, 6$. Then, in order to avoid the trivial solution and prevent the model from generating shortcuts to stop learning, we randomly shuffle the order in a CMP group, and update $\hat{P}$ with the corresponding CMP face number. Finally, the shuffled CMP image will be matched with the ERP image to find the position of local information in the global information. In addition, the proposed agent task **FindCMP** also requires feature extraction network and feature fusion module in the matching phase. Subsequently, we will introduce these two parts and how to use the proxy task to train the encoder $f_{\theta_E}$ respectively.

**The feature extraction networks.** Two separate feature extraction networks $f_{\theta_E}$ and $f_{\theta_C}$ are constructed corresponding to the ERP and one face of CMP format images respectively. The network $f_{\theta_E}: E \mapsto F_E$ takes an ERP image $E$ as input and extracts the global feature $F_E$. In the same way, the local feature is extracted by $f_{\theta_C} : c \mapsto F_c$, where $c$ is a face in CMP. As shown in Fig.4(a), the above two feature extraction networks use VGG16 architecture [28] with the tail 5-layer structure removed.

With the local feature extraction network, $f_{\theta_C}$ accepts all the faces in a CMP, so it implicitly learns panorama information contained in the sphere as $f_{\theta_E}$, as well as local information contained in one face of a CMP. By interacting these two feature extraction networks, $f_{\theta_E}$ can also learn the local features. This argument is validated in the global structural integrity experiment in Section 4.

**Cross attention fusion.** The features $F_E$ and $F_c$ from these two encoders are further processed by a fully connected layer respectively to obtain features $r_E$ and $r_c$ in the same dimension. Then they are fused based on a cross attention module shown in Fig.2, which is denoted as the following,

$$Q_E = r_E W_Q, \tag{1}$$
$$V_{c_i} = r_{c_i} W_V, \tag{2}$$
$$K_{c_i}^T = (r_{c_i} W_K)^T, \tag{3}$$
$$CA_i = ReLU \circ (Q_E K_{c_i}^T) V_{c_i} \tag{4}$$

for $i = 1, \cdots, 6$, where $Q_E$, $V_{c_i}$, and $K_{c_i}^T$ are the query, value, and key used in cross attention. $\circ$ stands for the function nesting operator. The final $CA_i \in \mathcal{R}^{512}$ is obtained through ReLU activation function and used to predict the location of a face of CMP in ERP.

In the cross attention fusion process, query and key are interacted by dot product, which means that only data with the same position will be enhanced. By this mechanism, the global feature $F_E$ takes part in the location prediction. When the error is propagated back, the feature extraction network $f_{\theta_c}$ guides the feature extraction network $f_{\theta_E}$ to pay more attention to the local features.

**Remark.** After the cross attention fusion, ReLU is chosen as activation function instead of traditional Softmax in the Transformer [29]. Using softmax means that the query needs to be multiplied by the key of every face. This will cause the model to pay more attention to the information constructed by $f_{\theta_C}$. It is inconsistent with our objective of training $f_{\theta_E}$. On the contrary, ReLU can activate the corresponding input information independently, excluding the interference of other faces.

**Training loss.** Finally, the location prediction based on the vector $CA_i$ is as follows,

$$\tilde{P}(c_i) = FC \circ CA_i \tag{5}$$

where $\tilde{P}(c_i)$ is the location prediction. Then $f_{\theta_E}$ can be trained by the following objective function,

$$L_{proxy} = \sum_{i=1}^{6} \left( \tilde{P}(c_i) - \hat{P}(c_i) \right)^2. \tag{6}$$

**Remark.** In order to avoid $f_{\theta_E}$ paying too much attention to local information and causing the loss of global information, we adopt all the six faces in a CMP group for prediction at a time. The necessity of using all CMP faces will be discussed in the experiment section.

### *B. Decoder Training*

As shown in Fig.2(b), after completing the training of $f_{\theta_E}$. A decoder $g_\theta : F_E \mapsto \tilde{Y}$ is constructed to map the feature $F_E$ to a saliency map $\tilde{Y}$. In the following, the decoder is explained from three perspectives: the definition of saliency image, model building and training.

**Saliency map preparation.** The user's head and eye movements are recorded at fixed time intervals to obtain a series of points. Then the fixation map is obtained by the following equation:

$$S_{ij} = \begin{cases} 1 & \text{if (i, j) is recorded,} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where $S_{ij}$ is a matrix with the same resolution as the image. However, it is difficult for us to learn and predict a sparse

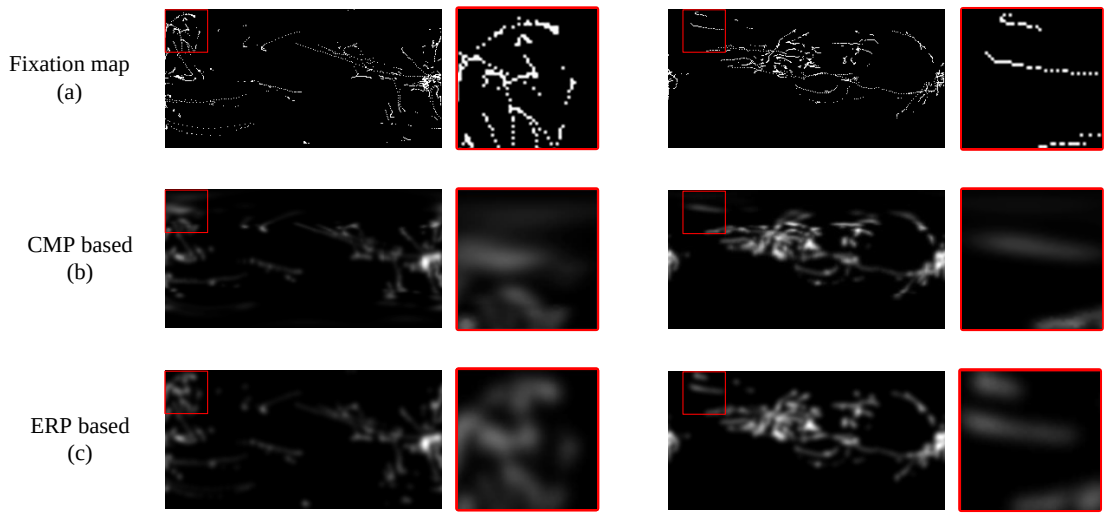


Fig. 5. Saliency map generation based on ERP and CMP. The red box represents the magnified area.

matrix with time series information. Accordingly, a Gaussian kernel is used to convolve the fixation map to obtain continuous regions, which is the saliency map in training. According to [30], we set the expansion angle of the Gaussian kernel to 5 degrees, and perform the following step:

$$\hat{Y} = G(S_E) \tag{8}$$

where $G$ is a Gaussian Kernel. We can see that in the above generation process, all fixations are treated equally. However, in ERP, the pixel density is different at different latitudes, i.e., the further away from the equator, the lower the density. This leads to a large number of fixations near the pole of the ERP, which actually only occupies a small part of the real field of view. It is inconsistent with our setting. To solve this problem, We took the method as follows,

$$\hat{Y} = \mathcal{T}_{back}(G(\mathcal{T}(S_E)) \tag{9}$$

where $\mathcal{T}$ converts ERP to CMP format to obtain a less distorted image to fit the generation process. $\mathcal{T}_{back}$ projects the resulting CMP saliency maps back into ERP format.

From Fig.5(a), we can see that there are some fixations near the pole area, and the CMP-based method tends to ignore those points (Fig.5(b)), while these fixations generate salient regions (Fig.5(c)) in ERP-based approach. The ERP format has lower pixel density in regions farther from the equator, which means that the pixel distances at the pole area are much closer than what is shown on the ERP. Therefore, the viewpoints in the red box may only occupy a small area or even a single point in the real field of view. It is obviously wrong for the ERP-based method to treat all pixels equally, so we adopt the CMP-based method that is closer to the real field of view to generate the saliency map.

**Decoder model.** The decoder model $g_\theta$ is shown in Fig.4(b) based on the 2D saliency prediction model SalGan [13]. Using the U-shaped structure to decode the large receptive field feature after multiple max pooling can make good use of the context to determine whether it belongs to the saliency region. With a simple decoder, we can prove that experimentally the obtained feature $F_E$ can predict a high-quality saliency image without the need of additional fusion at the decoding end under the premise of ensuring the inference speed.

**Algorithm 1** Asynchronous training algorithm

**Require**: Global encoder $f_{\theta_E}$, local encoder $f_{\theta_C}$, decoder $g_\theta$, unlabeled data set $D_u$, labeled data set $D_s$, projection operation $\mathcal{T}$, random shuffled $\mathcal{R}$.

**function** FindCMP($f_{\theta_E}$, $f_{\theta_C}$, $D_u$, $\mathcal{T}$, $\mathcal{R}$)
  **Data preparation:**
  for all $E_u^j \in D_u$ do
    $\mathcal{T}(E_u^j) = C_i$, $P = (1, 2, 3, 4, 5, 6)$
    $\mathcal{R}(C, P) = c, \hat{P}$
    **Encoder training:**
    Update $\theta_E, \theta_C$ by optimizing $L_{proxy}(\tilde{P}, \hat{P})$
  End for
  **return** $f_{\theta_E}$
**end function**
**function** Decoder training($f_{\theta_E}$, $g_\theta$, $D_s$)
  for all $(E_s^i, \hat{Y}_s^i) \in D_s$ do
    $g_\theta(f_{\theta_E}(E_s^i)) = \tilde{Y}$
    Update $g_\theta$ by optimizing $L_{KLD}(\tilde{Y}, \hat{Y})$
  End for
  **return** $g_\theta$
**end function**

**Training loss.** The Kullback-Leibler Divergence (KLD) loss is used to measure the distance between the predicted value and the real value as follows,

$$L_{KLD}(\tilde{Y}, \hat{Y}) = \sum_{i=1}^{w_e \times w_e} \hat{Y}_i \log\left(\varepsilon + \frac{\hat{Y}_i}{\varepsilon + \tilde{Y}_i}\right), \tag{10}$$

where the predicted distribution is $\tilde{Y} \in [0,1]^{w_e \times w_e}$, and the real value is $\hat{Y} \in [0,1]^{w_e \times w_e}$. The constant $\varepsilon$ ($\varepsilon$=1e-50) prevents the value blows up when the predicted value is close to 0.

As most of areas in saliency map are 0, KLD loss is more dependent on the area with large values $\hat{Y}_i$. It will make the

TABLE I
PERFORMANCE COMPARISON ON SALIENT360! DATASET. THE BEST SCORES ARE MARKED IN BOLD AND SECOND BEST IN RED.

| Model | | Venue | AUC-J ↑ | NSS ↑ | CC ↑ | SIM ↑ | KLD ↓ |
|---|---|---|---|---|---|---|---|
| Direct prediction | UNISAL[14] | ECCV20 | 0.7563± .022 | 0.9114±.096 | 0.6782±.029 | 0.6499±.024 | 1.4305±.29 |
| | SalGan[13] | arXiv17 | 0.7609±.022 | 0.8942±.081 | 0.658±.026 | 0.6301±.019 | 0.5282±.055 |
| | SaltiNet[21] | ICCVW17 | 0.7460±.027 | 0.6781±.073 | 0.6301±.024 | 0.7895±.011 | 0.1447±.013 |
| Multiple projection fusion | MV-SalGAN360[18] | TMM20 | **0.8028±.018** | **1.1680±.088** | **0.8106±.023** | 0.7342±.028 | 0.6635±.249 |
| | ATSAL[19] | ICPRW21 | 0.7255±.024 | 0.8207±.108 | 0.5107±.037 | 0.5928±.014 | 1.2795±.257 |
| Self-supervised approach | Rethink[16] | ICCV21 | 0.7565±.029 | 0.7927±.094 | 0.6720±.031 | 0.8021±.011 | 0.1239±.013 |
| | Rethink+ | ICCV21 | 0.7570±.028 | 0.7882±.092 | 0.6637±.03 | 0.7999±.011 | 0.1262±.011 |
| | EPSNet | Ours | 0.7607±.029 | 0.8642±.106 | 0.7141±.031 | **0.8096±.01** | **0.1125±.012** |

model focus on the viewpoint position, which is close to the requirement of using saliency image for viewpoint prediction in 360° data to reduce the bitrate and be more in line with the actual needs.

### C. Overall function

The overall saliency model will be trained by the following formula:

$$L_T = L_{proxy} + L_{KLD}, \tag{11}$$

asynchronous training of the model consists of training the encoder using the proxy task FindCMP with $L_{proxy}$ and supervised learning of the decoder with $L_{KLD}$. Due to the lack of labeled saliency images, we process the unlabeled images and get the corresponding input and labels, and train the encoder. The model can learn the geometric information of the input format ERP from the designed proxy task, so as to better predict the saliency images. This process is described in Algorithm 1.

## IV. EXPERIMENT

In this section, we will verify the effectiveness of the saliency encoder trained on FindCMP. First, we compare the frozen encoder-trained model with other saliency detection models. Then we illustrate the gap in inference speed between different methods. Finally, we perform ablation experiments to demonstrate the effectiveness of cross attention, the necessity of implicit panoramic views when training the encoder, and the anti-interference ability of the proposed proxy task against center bias.

### A. Experiment Setup

**Datasets.** Three public datasets, unlabeled ERP images provided by [16], VR-EyeTracking [31], and Salient360! [32] are used for encoder training, decoder training, and evaluation, respectively. To train the encoder, we selected 8000 gravity alignment ERP images [33] from the unlabeled dataset as the training set to ensure consistency with ERP images in the decoder dataset. After completing the training of the encoder, the limited labeled image saliency dataset is used for the next training step. We intercept 4081 images in the VR-EyeTracking [31] video dataset in units of 1 second to obtain images with similar information but different fixation locations, which simulates the situation where the saliency collection time of images is much longer than that of videos, and used the provided fixation files of eye and head movements to generate fixation maps. The resulting fixation maps generate the saliency maps with the Gaussian Kernel. After completing the training of the entire model, the Salient360! dataset (85 training/26 testing) is used to evaluate the model, and since the labels of its test set are not public, we choose to test on the training set (not trained on this dataset).

**Evaluation metrics.** To be consistent with the measurement methods of the previous saliency model, all the experiments are compared under the following five metrics: Normalized Scanpath Saliency (NSS), Kullback-Leibler Divergence (KLD), Similarity (SIM), Linear Correlation (CC), and AUC-Judd (AUC-J). The detail metric explanation can be found in [31]. All the evaluations are compared at 320×160 resolution.

**Implementation details.** The models in this paper are built based on Pytorch and RTX3090. All feature encoders are trained with the following parameters: the batch size is 80 as Rethink [16], the learning rate is set as 1e-4, adam [34] optimizer is used, and 100 epochs are trained in the self-supervised training set. In decoder training, we only train the decoder for 20 epochs instead of the 100 epochs in Rethink to prevent over-learning of video saliency. Except the batch is set to 16, all the other settings are the same with encoder training. It is worth noting that for the fair comparison with the Rethink model, we freeze the weights of the encoders and train the decoder on the VR-EyeTracking [31] using the same parameters.

### B. Comparison with state-of-the-art methods

**Compared methods.** We compare the proposed model with six models, including three direct prediction models, UNISAL [14], SalGan [13], and SaltiNet [21]; and two models designed based on multiple projection fusion, i.e., MV-SalGAN360 [18], ATSAL [35]. In addition, we also compare with the self-supervised learning based approach Rethink [16]. To compare our proxy task with contrastive learning in Rethink, we build a variant Rethink+ that uses an encoder trained by contrastive learning and combined with EPSNet decoder. All the decoders of these self-supervised based models are trained on the processed VR-EyeTracking [31] image dataset with the same parameters and frozen encoder weights.

**Quantitative comparison.** Table I shows the experimental results on 25 images randomly selected from the training set on salient360!. We can get the following observations. First, our

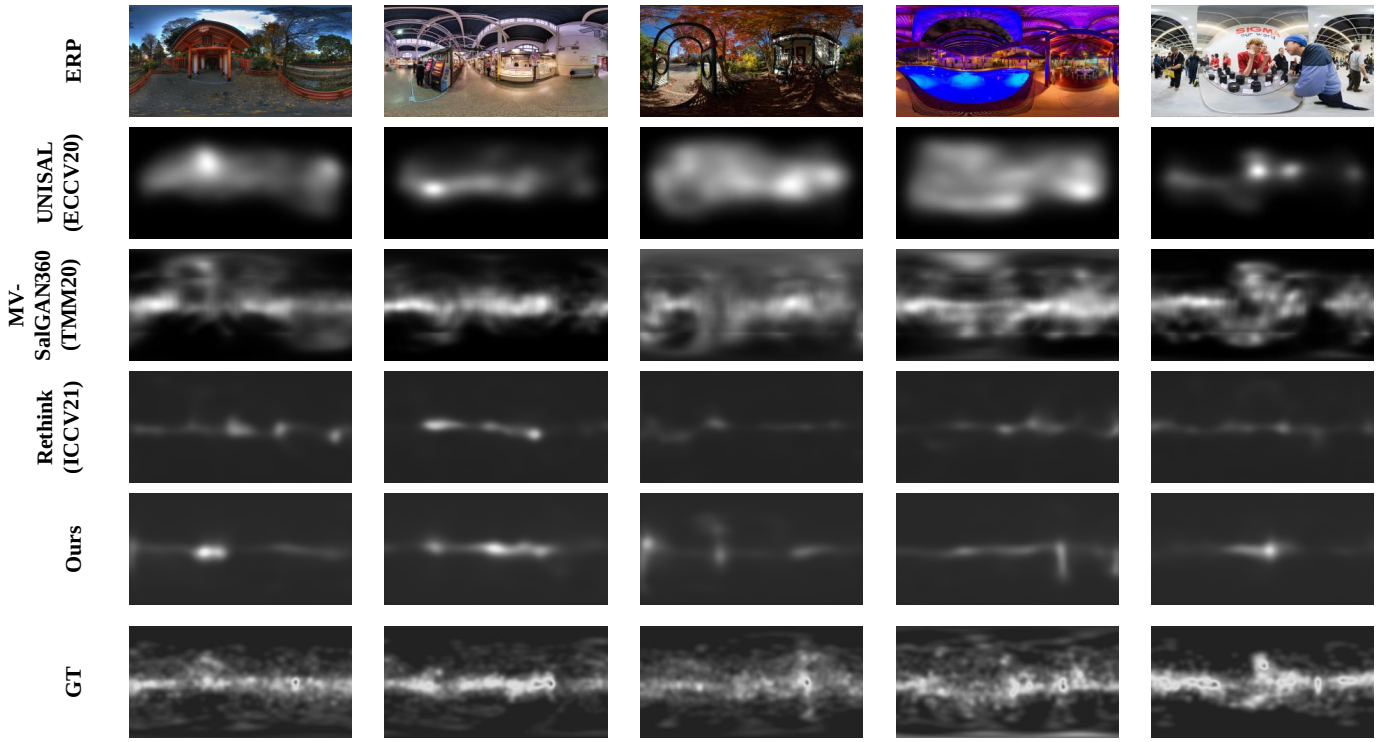


Fig. 6. Visualizations of saliency predictions on Salinet360! dataset using different models.

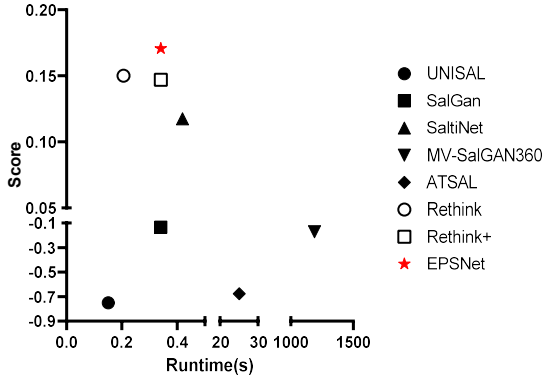


Fig. 7. Comprehensive comparison of runtime and integral performance of all indicators.

method EPSNet achieves the best performance on two indicators KLD and SMI, compared with all SOTA approaches. The multiple projection fusion approach MV-SalGAN360 achieves the best performance on other three indicators, i.e., **CC**, **ACU-J** and **NSS**. However, by combining Table III with Table I, it can be found that this method consumes much more inference time than other methods (over 3000 times of our method). By taking whether the inference time is greater than 1 second as the threshold, existing methods can be divided into two categories: long time group and short time group for further comparison. It can be found in short time group that our EPSNet also reaches the best on **CC** indicator, and the gap between the best value on **ACU-J** is small. For the indicator **NSS**, our EPSNet ranks third. The reason is that **NSS** cares more about whether the salient region is covered or not, but ignores the overflowing non-salient region. And the 2D direct prediction methods have a larger prediction area, resulting in better performance on this indicator. Second, compared with the self-supervised learning based approach, EPSNet obtains better results because it can extract more better local features. However, by comparing Rethink with Rethink+, we can find that the effect of Rethink+ has decreased, which is caused by the mismatch between the simple feature representation and the complex decoder structure. The model ATSAL does not work well because it is designed for video prediction. It treats the input image as the first frame of the video, and many salient regions are not predicted. All of the above mentioned facts prove that our proxy task strategy works.

In order to further verify the effectiveness of our method in terms of comprehensive performance. We apply the Coefficient of Variation (CV) method to get the weight of each indicator [36], and carry out the weighting operation to get the integral score. First of all, the CVs of five indicators are calculated through the following formula,

$$CV = \frac{S}{M} \tag{12}$$

where $S$ and $M$ stand for standard deviation and the mean of each indicator. The **KLD** indicator is given a negative number to keep consistent with the other four indicators, that is, the greater the better. The weight to each indicator is obtained by normalizing these five CVs. The final score is obtained by multiplying the weight with the corresponding indicator. As shown in Fig.7, EPSNet achieves the best score under the condition of acceptable inference speed.

To show that the proposed model EPSNet can have stable performance, we conduct a confidence interval study for

TABLE II
ABLATION STUDY ON FUSION ARCHITECTURE AND THE INPUT FACE NUMBER. THE NUMBER STANDS FOR HOW MANY FACES OF A CMP IS INPUT AND THE FULLY CONNECT MEANS FUSING THE FEATURES BASED ON A FULLY CONNECTED LAYER.

| | Fully Connect | | | | | Cross Attention | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Number | AUC-J ↑ | NSS ↑ | CC ↑ | SIM ↑ | KLD ↓ | AUC-J ↑ | NSS ↑ | CC ↑ | SIM ↑ | KLD ↓ |
| 1 | 0.758 | 0.857 | 0.700 | 0.808 | 0.116 | 0.755 | 0.793 | 0.658 | 0.799 | 0.130 |
| 2 | 0.753 | 0.787 | 0.658 | 0.800 | 0.128 | 0.756 | 0.811 | 0.665 | 0.801 | 0.127 |
| 3 | 0.756 | 0.816 | 0.675 | 0.802 | 0.125 | 0.756 | 0.815 | 0.689 | 0.805 | 0.120 |
| 4 | 0.756 | 0.848 | 0.707 | 0.808 | 0.115 | 0.758 | 0.816 | 0.682 | 0.804 | 0.121 |
| 5 | 0.754 | 0.790 | 0.669 | 0.801 | 0.125 | 0.758 | 0.830 | 0.688 | 0.806 | 0.118 |
| 6 | 0.760 | 0.801 | 0.668 | 0.801 | 0.127 | 0.761 | 0.864 | 0.714 | 0.810 | 0.113 |

TABLE III
COMPARISON OF INFERENCE TIME ON SALIENT360!.

| Model | Runtime(s) |
|---|---|
| MV-SalGAN360[18] | 1189.530 |
| ATSAL[19] | 25.058 |
| SaltiNet[21] | 0.419 |
| Rethink[16] | 0.206 |
| UNISAL[14] | 0.153 |
| Our Model | 0.341 |

TABLE IV
ABLATION STUDY ON DIFFERENT CONFIGURATIONS OF THE EPSNET. THE RESULTS FROM TOP TO BOTTOM REPRESENT THAT THE DECODER TRAINING SET IS DIRECTLY USED FOR TRAINING WITHOUT FREEZING THE ENCODER WEIGHT, SOFTMAX IS USED IN CROSS ATTENTION, ROTATED ERP IMAGES BEFORE THE PROCESS OF PROXY TASK DATA PROCESSING, GENERATED SALIENCY MAP IN ERP-BASED APPROACH, AND THE PROPOSED METHOD.

| | AUC-J ↑ | NSS ↑ | CC ↑ | SIM ↑ | KLD ↓ |
|---|---|---|---|---|---|
| Direct train | 0.755 | 0.810 | 0.680 | 0.801 | 0.124 |
| Softmax | 0.757 | 0.831 | 0.699 | 0.804 | 0.119 |
| Rotation | 0.753 | 0.793 | 0.663 | 0.798 | 0.131 |
| ERP-based | 0.758 | 0.825 | 0.697 | 0.806 | 0.117 |
| Proposed | 0.761 | 0.864 | 0.714 | 0.810 | 0.113 |

performance comparisons on different indicators. For the $95\%$ confidence level, they are shown in Table I. We can see that in the last three rows of Table I, EPSNet has a similar or smaller confidence interval with the rethink model while improving performance, which proves that our model has reliable improvement in the test set.

**Qualitative results.** Fig. 6 visualizes the saliency results of three kinds of approaches and ours on four images (ERP) in the Salient360! dataset. Compared with the labeled saliency maps (GT), EPSNet is better than Rethink. It can observed that EPSNet predictions are more complete near the equator, and it is significantly more sensitive in non-equatorial regions. The 2D direct prediction model UNISAL has obvious spillover. It is contrary to the purpose of saving bitrate by transmitting the saliency region. Although MV-SalGAN360 can predict most of the saliency regions, due to the multi-view saliency prediction, the fusion of the saliency prediction results in local-view images makes many non-salient regions marked as bright spots.

**Inference time.** The inference time comparisons between different models are shown in Table III. For fair comparison, we choose i5-9400 CPU for inference in the Windows environment. UNISAL achieves the best result because of its simply architecture, while our model has slightly lower inference speed than the Rethink model due to its more complex decoder. Since MV-SalGAN360 performs multi-view late fusion, it needs a longer prediction time. From the comprehensive evaluation of efficiency and accuracy, our method achieves the best performance.

### C. Further Analysis

In this part, we will conduct exhaustive analysis from five perspectives: the analysis of ablation; the integrity of CMP; the choice of different loss functions; model fine tuning; and significance test.

**Ablation Experiment.** We verify the rationality of the proposed method by restoring or destroying each design in this paper. In the first and second rows of Table IV, removing the proxy task or using the original softmax to activate the query and key operation results in cross attention will result in poor performance. The former proves the effectiveness of FindCMP, while the latter reflects the rationality of ReLU. In the third row, we destroy the attribute of gravity alignment by randomly rotating the original ERP image in the encoder training set, and use these irregular images as a new training set. The third row in Table IV shows that the performance is degraded, proving the importance of screening gravity aligned ERP images in the encoder training set. When the saliency map is generated in the ERP way, the obtained saliency map is taken as the label and the result is shown Table IV (the fourth line noted as ERP-based). Due to the distortion in such ERP-based approach, it can be seen that the performance decreases compared with the CMP-based approach.

In order to verify the effectiveness of cross attention and input of all faces of a CMP, we conduct an ablation experiment shown in Table II. When feature fusion is based on a fully connected layer, the performance is not directly related to the number of input faces; while for cross attention fusion, the more faces are input, the better the performance is. Deep learning expects the performance of the model can be improved with more data, but the performance of full connected layer fluctuation indicates that it is not suitable for our proxy task. Our cross attention explicitly compute correlations between the local features and global features and organically combined them. Finally, the encoder can better learn the local features

TABLE V
IMPLICIT GLOBAL STRUCTURAL INTEGRITY.

| Rate | AUC-J ↑ | NSS ↑ | CC ↑ | SIM ↑ | KLD ↓ |
|---|---|---|---|---|---|
| 100% | 0.759 | 0.825 | 0.700 | 0.806 | 0.118 |
| 75% | 0.755 | 0.813 | 0.685 | 0.804 | 0.122 |
| 50% | 0.757 | 0.792 | 0.673 | 0.801 | 0.124 |
| 25% | 0.757 | 0.815 | 0.681 | 0.804 | 0.122 |
| 0% | 0.761 | 0.864 | 0.714 | 0.810 | 0.113 |

TABLE VI
ABLATION STUDY ON OBJECTIVE FUNCTION VARIANTS. K, C, N, AND B REPRESENT KLD, CC, NSS, AND BCE LOSSES RESPECTIVELY. THE LOSS FUNCTION IN THE THIRD ROW MEANS THE COMBINATION OF THE LOSSES KLD, CC AND NSS WHERE THE NUMBER IS THE RATIO. THE SECOND ROW IS SIMILAR.

| Loss | AUC-J ↑ | NSS ↑ | CC ↑ | SIM ↑ | KLD ↓ |
|---|---|---|---|---|---|
| K | 0.761 | 0.864 | 0.714 | 0.810 | 0.113 |
| K-C-N | 0.761 | 0.788 | 0.726 | 0.801 | 0.120 |
| 10K-2C-N | 0.757 | 0.810 | 0.696 | 0.806 | 0.117 |
| B | 0.761 | 0.849 | 0.704 | 0.810 | 0.113 |

of 6 faces, and the global features are organically integrated. Therefore, the more faces, the better the performance.

**Implicit** global structural integrity. At the proxy task, we assume that the panorama information can be implicitly extracted for the CMP face feature extraction network, so it can be well integrated with the ERP feature extraction network. When a sphere is projected into the CMP format, six separate faces are obtained. The first to fourth faces are at the equator and the fifth to sixth planes are at the polar. When the six faces are input into the CMP feature extractor at the same time, it can obtain the information of the whole sphere. We do an experiment to show the performance when some polar faces are replaced by its equator faces. In the training set, the replacement is done with a probability, i.e., the implicit global structural integrity is corrupted. We show the performances with conversion probability from 100% to 0% in Table V. It can be seen that the performance is the best without any conversion, verifying the importance of the implicit global structural integrity. The case of all conversion (100%) is also relatively good, because the samples are stored in a head up manner, and the saliency is mainly concentrated at the equator. The performance degrades the most when the conversion rate is between 25% and 75%, because they have neither global information nor focus on the equator.

**Objective function variants.** In previous studies, various loss functions and combinations have been applied to saliency detection. In order to find the optimal one, we use the four loss functions mentioned in [18] during the training process of the decoder, and their results are presented in Table VI. Among these functions, CC aims to calculate the correlation between saliency maps, with the same penalty for false positives and false negatives, which is inconsistent with our need to find salient regions as much as possible. NSS expects the model generate a result with a high value at the viewpoint to make the value of its value larger. Binary Cross Entropy (BCE) is similar to KLD, and also expects a similar distribution between the values and labels. In the comparison of first three rows, as the ratio of KLD increases, the performance of the model improves to a certain extent, which shows the adaptation of KLD for saliency detection. It is worth noting that the loss function in the second row is the same as the loss function used by MV-SalGAN360, and our method still maintains great advantages in KLD. The performance of BCE is close to KLD, because it also targets the strong saliency regions. However, when the label value is far from 0.5, the size of the loss function can well reflect the degree of similarity with $\hat{Y}$, and the model can be trained according to the size of the difference. However, we use saliency images instead of viewpoint maps for training, and inevitably will encounter values close to 0.5. At this time, if $\tilde{Y}$ is close to 1, training will fail with the gradient exploding. Finally, we choose the KLD as our loss function because it best meets the saliency requirement and performs best.

TABLE VII
STUDY ON BATCH NORMALIZATION. C, F AND B STAND FOR CROSS ATTENTION FUSION, FULLY CONNECTED LAYER BASED FUSION, AND BATCH NORMALIZATION, RESPECTIVELY.

| C | F | B | AUC-J↑ | NSS↑ | CC↑ | SIM↑ | KLD↓ |
|---|---|---|---|---|---|---|---|
| ✓ | ✗ | ✓ | 0.756 | 0.846 | 0.703 | 0.809 | 0.115 |
| ✓ | ✗ | ✗ | 0.761 | 0.864 | 0.714 | 0.810 | 0.113 |
| ✗ | ✓ | ✓ | 0.760 | 0.801 | 0.668 | 0.801 | 0.127 |
| ✗ | ✓ | ✗ | 0.758 | 0.806 | 0.677 | 0.804 | 0.121 |

**The study of batch normalization.** In many works, Batch Normalization (BN) is used to help model training. To verify its effectiveness in our model, we use this operation in two types of fusion models. After the cross attention model completes the fusion of features to obtain $CA$, only one fully connected layer is used to predict the result, so we perform a BN operation before this fully connected layer. For the case using fully connected layer to feature fusion, a total of three fully connected layers are used, and the BN operation is applied on the outputs of the first and second layers. It is worth noting that all training is done with total 6 CMP faces as input. In the first two columns of Table VII respect to cross-attention and fully-connected layer fusion respectively, we can see that after adding the BN operation, the performance of the model has declined, indicating that this operation is not suitable for our method. This is because BN operation will produce similar image features which reduces the pertinence of the proxy tasks, resulting in performance degradation.

**Fine tuning.** After completing EPSNet training, we no longer freeze the encoded weights, and fine-tune the model using VR-EyeTracking [31] dataset to see if the performance can be further improved. All the experimental results are shown in Table VIII. It can be observed from the rows (2-5) with 20 epochs, all indicators are decreased, and the performance of the model has steadily been improved as the learning rate decreases. This shows that the previously trained EPSNet has learned satisfactory features, further training will lead the model to loss of ability to capture local information.

In the rest of this table, we can see that without freezing the encoder weights, more training epochs actually lead to worse

TABLE VIII
FINE TUNING RESULTS WITH NO FREEZING ENCODER WEIGHT. THE FIRST ROW CONTAINS THE ORIGINAL RESULTS BEFORE FINETUNE. THE RESULTS IN ROWS (2-5) ARE TRAINED IN THE FIRST 20 EPOCHS USING THE LEARNING RATE 1E-4, THEN IN THE FURTHER 20 EPOCHS USING THE CORRESPONDING LEARNING RATES IN THE TABLE. THE LOWER HALF OF THE TABLE CONTAINS THE RESULTS AFTER DIFFERENT FINETUNE TRAINING EPOCHS USING THE LEARNING RATE 1E-4.

| Learning rate | AUC-J↑ | NSS↑ | CC↑ | SIM↑ | KLD↓ |
|---|---|---|---|---|---|
| Fix | 0.761 | 0.864 | 0.714 | 0.810 | 0.113 |
| 5e-5 | 0.754 | 0.782 | 0.654 | 0.797 | 0.130 |
| 1e-5 | 0.758 | 0.798 | 0.657 | 0.798 | 0.129 |
| 5e-6 | 0.758 | 0.809 | 0.667 | 0.800 | 0.126 |
| 1e-6 | 0.760 | 0.832 | 0.686 | 0.804 | 0.121 |
| Epoch | AUC-J↑ | NSS↑ | CC↑ | SIM↑ | KLD↓ |
| 20 | 0.758 | 0.820 | 0.688 | 0.805 | 0.120 |
| 15 | 0.759 | 0.847 | 0.704 | 0.809 | 0.117 |
| 10 | 0.756 | 0.832 | 0.696 | 0.806 | 0.118 |

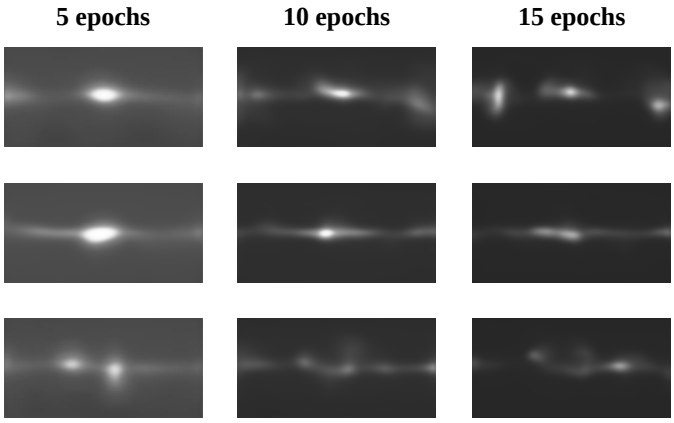


Fig. 8. Saliency maps at different training epochs.

results. To explain this phenomenon, we show the saliency maps with different epochs in Fig. 8. It can be observed that when training for only five epochs, the model predicts large saliency regions with a grey background. When the number of training increases, the saliency area gradually becomes smaller and the background becomes black. This is related to the loss function KLD, since KLD drives the model to learn those regions with the largest distance from the label image. It causes the model to ignore the background areas with small gaps when the number of training is small, resulting in a gray background. After the training of 20 epochs, the performance of the model is declined. This phenomena shows that the features learned by our proposed proxy task is injured by direct training.

**Significance test.** In order to verify the reliability of the improvements on the indicators in the comparative experiments, we compare EPSNet with the rest of the models in the main text using Wilcoxon signed-rank test. The experimental results are shown in Table IX. It can be observed that all values are less than 5% except for 2D image and video models on some indicators, proving that our changes on these models are robust. For the cases of larger values greater than 5%, this is due to the reason that the 360° saliecny map is predicted by directly using a 2D or video model. Due to the large difference between the 2D and 3D images, there is a large gap between the prediction results, resulting in high indicator values.

TABLE IX
WILCOXON SIGNED-RANK RESULTS COMPARED WITH EPSNET.

| Model | AUC-J | NSS | CC | SIM | KLD |
|---|---|---|---|---|---|
| UNISAL[14] | 0.128 | 0.563 | 0.042 | 0.000 | 0.000 |
| SalGan[13] | 0.600 | 0.510 | 0.003 | 0.000 | 0.000 |
| SaltiNet[21] | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| MV-SalGAN360[18] | 0.011 | 0.000 | 0.000 | 0.000 | 0.000 |
| ATSAL[19] | 0.001 | 0.150 | 0.000 | 0.000 | 0.000 |
| Rethink[16] | 0.015 | 0.001 | 0.005 | 0.012 | 0.006 |
| Rethink+ | 0.011 | 0.002 | 0.001 | 0.001 | 0.001 |

## V. CONCLUSION

We proposed a novel 360° saliency detection framework EPSNet embedded with a proxy task. The proxy task Find-CMP can use large unlabeled 360° images to self-supervise train a feature encoder which can extract local and global features from an ERP format image. Then these features are input to a decoder to predict saliency map. Compared with the previous complex multiple projection and fusion process, EPSNet is fast and also accurate. In contrast to the self-supervised approach and 2D extension models based on only ERP images, EPSNet can extract much better local features, so the saliency prediction accuracy is better. Experiments demonstrate the effectiveness and efficiency of our method. Moreover, not only with saliency detection, theoretically, the framework can also be applied to other 360°-related tasks, such as object detection and semantic segmentation.

## REFERENCES

[1] Y. Li, W. Liao, J. Huang, D. He, and Z. Chen, "Saliency based perceptual hevc," in *2014 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*. IEEE, 2014, pp. 1–5.
[2] A. Polakovič, R. Vargic, and G. Rozinaj, "Adaptive multimedia content delivery in 5g networks using dash and saliency information," in *2018 25th International Conference on Systems, Signals and Image Processing (IWSSIP)*. IEEE, 2018, pp. 1–5.
[3] R. Yang, M. Xu, Z. Wang, Y. Duan, and X. Tao, "Saliency-guided complexity control for hevc decoding," *IEEE Transactions on Broadcasting*, vol. 64, no. 4, pp. 865–882, 2018.
[4] M. Paul, "Efficient multiview video coding using 3-d coding and saliency-based bit allocation," *IEEE Transactions on Broadcasting*, vol. 64, no. 2, pp. 235–246, 2018.
[5] W. Yang, Y. Qian, J.-K. Kämäräinen, F. Cricri, and L. Fan, "Object detection in equirectangular panorama," in *2018 24th international conference on pattern recognition (icpr)*. IEEE, 2018, pp. 2190–2195.
[6] P. Zhao, A. You, Y. Zhang, J. Liu, K. Bian, and Y. Tong, "Spherical criteria for fast and accurate 360 object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, pp. 12 959–12 966.
[7] K. Yang, J. Zhang, S. Reiß, X. Hu, and R. Stiefelhagen, "Capturing omni-range context for omnidirectional segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1376–1386.
[8] K. Yang, X. Hu, L. M. Bergasa, E. Romera, and K. Wang, "Pass: Panoramic annular semantic segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 10, pp. 4171–4185, 2019.
[9] C. Wu, R. Zhang, Z. Wang, and L. Sun, "A spherical convolution approach for learning long term viewport prediction in 360 immersive video," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, pp. 14 003–14 040.
[10] A. Yaqoob and G.-M. Muntean, "A combined field-of-view prediction-assisted viewport adaptive delivery scheme for 360° videos," *IEEE Transactions on Broadcasting*, vol. 67, no. 3, pp. 746–760, 2021.

[11] Z. Jiang, X. Zhang, Y. Xu, Z. Ma, J. Sun, and Y. Zhang, "Reinforcement learning based rate adaptation for 360-degree video streaming," *IEEE Transactions on Broadcasting*, vol. 67, no. 2, pp. 409–423, 2021.
[12] M. Xu, C. Li, S. Zhang, and P. Le Callet, "State-of-the-art in 360 video/image processing: Perception, assessment and compression," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 1, pp. 5–26, 2020.
[13] J. Pan, C. C. Ferrer, K. McGuinness, N. E. O'Connor, J. Torres, E. Sayrol, and X. Giro-i Nieto, "Salgan: Visual saliency prediction with generative adversarial networks," *arXiv preprint arXiv:1701.01081*, 2017.
[14] R. Droste, J. Jiao, and J. A. Noble, "Unified image and video saliency modeling," in *European Conference on Computer Vision*. Springer, 2020, pp. 419–435.
[15] A. Borji, "Saliency prediction in the deep learning era: An empirical investigation," *arXiv preprint arXiv:1810.03716*, vol. 10, 2018.
[16] Y. A. D. Djilali, T. Krishna, K. McGuinness, and N. E. O'Connor, "Rethinking 360deg image visual attention modelling with unsupervised learning." in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 414–15 424.
[17] F.-Y. Chao, L. Zhang, W. Hamidouche, and O. Deforges, "Salgan360: Visual saliency prediction on 360 degree images with generative adversarial networks," in *2018 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 2018, pp. 01–04.
[18] F.-Y. Chao, L. Zhang, W. Hamidouche, and O. Déforges, "A multi-fov viewport-based visual saliency model using adaptive weighting losses for 360° images," *IEEE Transactions on Multimedia*, vol. 23, pp. 1811–1826, 2020.
[19] Y. Dahou, M. Tliba, K. McGuinness, and N. O'Connor, "Atsal: An attention based architecture for saliency prediction in 360° videos," in *International Conference on Pattern Recognition*. Springer, 2021, pp. 305–320.
[20] C. Wu, H. Zhao, and X. Shang, "Octagonal mapping scheme for panoramic video encoding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 9, pp. 2402–2406, 2018.
[21] M. Assens Reina, X. Giro-i Nieto, K. McGuinness, and N. E. O'Connor, "Saltinet: Scan-path prediction on 360 degree images using saliency volumes," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 2331–2338.
[22] B. Niu, W. Wen, W. Ren, X. Zhang, L. Yang, S. Wang, K. Zhang, X. Cao, and H. Shen, "Single image super-resolution via a holistic attention network," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*. Springer, 2020, pp. 191–207.
[23] Y. Yan, W. Ren, X. Hu, K. Li, H. Shen, and X. Cao, "Srgat: Single image super-resolution with graph attention network," *IEEE Transactions on Image Processing*, vol. 30, pp. 4905–4918, 2021.
[24] H.-T. Cheng, C.-H. Chao, J.-D. Dong, H.-K. Wen, T.-L. Liu, and M. Sun, "Cube padding for weakly-supervised saliency prediction in 360 videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1420–1429.
[25] M. Gutmann and A. Hyvarinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2010, pp. 297–304.
[26] G. Larsson, M. Maire, and G. Shakhnarovich, "Colorization as a proxy task for visual understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6874–6883.
[27] J. Li, J. Liu, Y. Wong, S. Nishimura, and M. S. Kankanhalli, "Self-supervised representation learning using 360 data," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 998–1006.
[28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
[30] F.-Y. Chao, C. Ozcinar, C. Wang, E. Zerman, L. Zhang, W. Hamidouche, O. Deforges, and A. Smolic, "Audio-visual perception of omnidirectional video for virtual reality applications," in *2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 2020, pp. 1–6.
[31] Y. Xu, Y. Dong, J. Wu, Z. Sun, Z. Shi, J. Yu, and S. Gao, "Gaze prediction in dynamic 360 immersive videos," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5333–5342.
[32] J. Gutierrez, E. J. David, A. Coutrot, M. P. Da Silva, and P. Le Callet, "Introducing un salient360! benchmark: A platform for evaluating visual attention models for 360 contents," in *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 2018, pp. 1–3.
[33] L. He, B. Jian, Y. Wen, H. Zhu, K. Liu, W. Feng, and S. Liu, "Rethinking supervised depth estimation for 360° panoramic imagery," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2022, pp. 5169–5177.
[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[35] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," *arXiv preprint arXiv:1803.07728*, 2018.
[36] Y. Sun, X. Liang, and C. Xiao, "Assessing the influence of land use on groundwater pollution based on coefficient of variation weight method: A case study of shuangliao city," *Environmental Science and Pollution Research*, vol. 26, pp. 34 964–34 976, 2019.

**Zizhuang Zou** received his B.S. degree in the software engineering from Yangtze University in 2021 and is now a master student of University of Electronic Science and Technology of China. His research interests include 360 degree image and video, and image processing.

**Mao Ye** (Member, IEEE) received the B.S. degree from Sichuan Normal University, Chengdu, China, in 1995, and the M.S degree from University of Electronic Science and Technology of China, Chengdu, China, in 1998 and Ph.D. degree from Chinese University of Hong Kong, China, in 2002, all in mathematics. He has been a short-time visiting scholar at University of Queensland, and University of Pennsylvania. He is currently a professor and director of CVLab with University of Electronic Science and Technology of China, Chengdu, China. His research interests include machine learning and computer vision. In these areas, he has published over 90 papers in leading international journals or conference proceedings. He has served on the editorial board of ENGINEERING APPLICATIONS OF ARTIFICIAL INTELLIGENCE. He was a co-recipient of the Best Student Paper Award at the IEEE ICME 2017.

**Shuai Li** (Member, IEEE) is currently with the School of Control Science and Engineering, Shandong University (SDU), China, as a Professor and QiLu Young Scholar. He was with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, China, as an Associate Professor from 2018-2020. He received his Ph.D. degree from the University of Wollongong, Australia, in 2018. His research interests include image/video coding, 3D video processing and computer vision. He was a co-recipient of two best paper awards at the IEEE BMSB 2014 and IIH-MSP 2013, respectively.

**Xue Li** (Member, IEEE) received the Ph.D. degree from the Queensland University of Technology, Brisbane, QLD, Australia, in 1997. He is currently a professor with the School of Information Technology and Electrical Engineering, University of Queensland (UQ), Brisbane, QLD, Australia. He also holds a professor title with the School of Medicine, Griffith University, Gold Coast, QLD, Australia. His major areas of research interests and expertise include machine learning, health data analytics, data mining, social computing, and intelligent web information systems.

**Fréderic Dufaux** (Fellow, IEEE) received the M.Sc. degree in physics and the Ph.D. degree in electrical engineering from EPFL in 1990 and 1994, respectively. He is a CNRS Research Director with the CNRS, Centrale Supélec, Laboratoire des Signaux et Systèmes (L2S, UMR 8506), Université Paris-Saclay, where he is the Head of the Telecom and Networking Hub. He has authored or coauthored three books, more than 200 research publications and 20 patents issued or pending. His research interests include image and video coding, 3D video, high dynamic range imaging, visual quality assessment, video surveillance, privacy protection, image and video analysis, multimedia content search and retrieval, and video transmission over wireless network.,Dr. Dufaux was the Chair of the IEEE SPS Multimedia Signal Processing Technical Committee in 2018 and 2019. He is a member of the IEEE SPS Technical Directions Board. He was the Vice General Chair of ICIP 2014, the General Chair of MMSP 2018, and the Technical Program Co-Chair of ICIP 2019 and ICIP 2021. He is also a Founding Member and the Chair of the EURASIP Technical Area Committee on Visual Information Processing.