Intra- and Inter-reasoning Graph Convolutional Network for Saliency Prediction on 360° Images

First A. Author, Fellow, IEEE, Second B. Author, and Third C. Author, Jr., Member, IEEE

Abstract—Cubic projection can be utilized to divide 360° images into multiple rectilinear images, with little distortion and semantic relevance. However, the existing saliency prediction models fail to integrate semantic information. In this paper, we address this by proposing an intra- and inter-reasoning graph convolutional network for saliency prediction on 360° images (SalReGCN360). The whole framework contains six sub-networks, each of which contains two branches. In the training phase, after utilizing Multiple Cubic Projection (MCP), six rectilinear images are simultaneously put into corresponding sub-networks. In one of the branches, the global features of a single rectilinear image are extracted by the intra-graph inference module. In the other branch, the semantic information of the six rectilinear images is integrated by the inter-graph inference module. Then, the feature maps are generated by fusing the two branches, and six corresponding rectilinear saliency maps are predicted. In the testing phase, the Multiple Sphere Rotation (MSR) projection method is applied to obtain multiple new equirectangular images, and the images are predicted and fused to generate the saliency map. Extensive experiments on two datasets show that the proposed model has improved, with different degrees in the four saliency evaluation metrics.

Index Terms—virtual reality, 360° image, saliency prediction, graph convolutional network

I. INTRODUCTION

With the continuous development of virtual reality film capture [1], human-computer interaction [2], content creation or editing [3] and other applications, saliency prediction on 360° images has been received more and more attention from researchers. Research on saliency prediction on 360° images is not only helpful for the design of key technologies, such as user interface and eye tracking of future VR systems, but is also helpful for researchers to better understand the visual behaviors of users in virtual environments. Although saliency prediction on 360° images has been gradually attracted, the cost of collecting fixation data of observers in virtual environment is relatively high. Therefore, a large, public saliency prediction dataset on 360° images has not yet been built. To a large extent, this limits the development of saliency prediction model on 360° images, especially those with a model framework based on neural networks.

In order to solve the problem, some models utilize cubic projection to project a 360° image into six rectilinear images, so as to increase the amount of data for model training. Each rectilinear image corresponds to a region in equirectangular image, so the scene contents in each image are semantically related. As shown in Figure I(a), when training the existing models by utilizing the rectilinear images and corresponding saliency maps, all the images are first randomly shuffled, then independently put into the network, and the saliency maps are then predicted. This training mode can only enable the model to predict the saliency on a single rectilinear image, but cannot integrate the semantic information of multiple rectilinear images.

This paragraph of the first footnote will contain the date on which you submitted your paper for review. It will also contain support information, including sponsor and financial support acknowledgment. For example, “This work was supported in part by the U.S. Department of Commerce under Grant BSI123456.” The next few paragraphs should contain the authors’ current affiliations, including current address and e-mail. For example, F. A. Author is with the National Institute of Standards and Technology, Boulder, CO 80305 USA (e-mail: author@boulder.nist.gov). S. B. Author, Jr., was with Rice University, Houston, TX 77005 USA. He is now with the Department of Physics, Colorado State University, Fort Collins, CO 80523 USA (e-mail: author@lamar.colostate.edu). T. C. Author is with the Electrical Engineering Department, University of Colorado, Boulder, CO 80309 USA, on leave from the National Research Institute for Metals, Tsukuba, Japan (e-mail: author@nrim.go.jp).
images, and establish the semantic relevance among images. Therefore, this will affect the prediction effect of the saliency model on the whole 360° image.

With the deep integration of data science and other fields, data in more and more fields can be represented in the form of a graph. In the field of computer vision, the various irregular objects and regions in the image have certain spatial and semantic associations. Therefore, by utilizing the characteristics of graph structure, the nodes on the graph can represent regions in the image, while the edges connecting nodes represent the correlation degree of different regions. Since the structure of graphs is irregular, all the nodes on the graph are not strictly arranged in a specified spatial order, and each node will have a different number of neighbors. Some important operations, such as convolution, can be easily operated in the image domain, but not on a graph. With the rise of graph-structured data, and the difficulty of modelling graph-structured data with traditional convolutional networks, Graph Convolutional Network (GCN) are gradually attracted. These are a kind of neural network architecture which can aggregate adjacent node information in a convolutional manner, and has a strong expressive ability to learn graph representation.

In this paper, we proposes a new saliency prediction model for 360° images based on GCN, which we call Intra- and Inter-reasoning Graph Convolutional Network (SalReGCN360). A brief overview of the SalReGCN360 framework is shown in Fig. 1(b). In order to integrate the semantic information of multiple rectilinear images obtained by cubic projection, these images are put into the network in parallel. The novel model framework contains six sub-networks, each of which contains two branches. In the training phase, the six rectilinear images are put into corresponding six sub-networks, after these images have been generated by the Multiple Cubic Projection (MCP) [4]. In one of the branches, the intra-graph inference module is extracts the global features of a single rectilinear image. In the other branch, the inter-graph inference module integrates the semantic information of six rectilinear images. Then the feature maps generated by the two branches are fused, and six corresponding saliency maps are predicted. In the testing phase, the Multiple Sphere Rotation (MSR) and Reversed Multiple Sphere Rotation (R-MSR) projection [5] is applied to effectively predict saliency on each region of a 360° image. In short, the main contributions of this paper are as follows:

1. A new graph convolutional network with six sub-networks is proposed. Each sub-network contains two branches, which are utilized to predict the saliency on 360° images of the single and multiple rectilinear images, respectively.

2. A new inter-graph inference module is proposed to integrate the semantic information of six rectilinear images and extract the contextual features of the whole 360° images.

3. An intra-graph inference module is adopted to extract the global features of a single rectilinear image, and predict the local saliency of 360° images more finely.

4. Extensive experiments on Salient360 and VR Saliency datasets verify that the proposed model is superior to the validated state-of-the-art methods.

II. RELATED WORK

For saliency prediction on 360° images, manual features and shallow networks are adopted in traditional methods. Ling et al. [6] extracted image features based on the dictionary of sparse representation, and then utilized image features and latitude bias enhancement to predict saliency on 360° images. The model proposed by Federica et al. [7] took advantage of the low-level features such as contrast, saturation and texture information, as well as some high-level features such as skin, face, and so on. Compared with only utilizing low-level manual features, the introduction of high-level semantic features was indeed beneficial and improved the prediction effectiveness of saliency regions. These methods can predict the saliency on 360° images with simple scenes, but not perform well with complex ones.

In order to further improve the performance of saliency prediction on 360° images, and considering that the performance of saliency prediction on traditional 2D images has been significantly improved with the help of deep learning, deep networks have been adopted in some saliency models. However, due to the lack of a large saliency prediction dataset for 360° images, almost all models based on deep convolutional network have to be pre-trained on the saliency prediction dataset on traditional 2D images. At present, researchers tend to combine transfer learning and data augmentation to solve the problem of using limited data to train the models effectively [8]. For 360° images, projection transformation is mainly adopted to enhance data. Equirectangular projection is the most commonly used projection method [9], and based on multiple equirectangular projections, saliency maps with different longitude boundaries have been uniformly integrated to generate a final saliency map [10]. However, the problem of severe distortion near the upper and lower boundaries of the images is still not solved.

There are other projection methods, such as cubic, octahedral projections, etc. [11]. SalNet360 utilized cubic projection to project a 360° image onto six faces of a cube, each of which corresponds to 90° view [12]. These image with local view are much less distorted, just like more traditional 2D images. However, this kind of projection method will cause obvious discontinuity when reversing the predicted saliency maps, using reverse projection. SalGAN360 proposed MCP on 360° image, to obtain multiple images with local view and less distortion [4]. MCP has also been used to solve the discontinuity problem of boundaries in fusion, but neither SalNet360 nor SalGAN360 are able to obtain the global information in the images, with local view.

Although the GCN has been widely applied in many fields of computer vision, its application to saliency prediction on 360° images is still in its infancy. SalGCN proposed by Haoran et al. [13] was the first to apply GCN to saliency prediction in 360° images. In this model, the constructed graphs have a spherical structure, which can directly convolve 360° images on the sphere, without first utilizing plane projection methods. However, due to the small size of the saliency prediction dataset on 360° images, SalGCN is prone to over-fitting, and so has poor generalized performance.
Combined with the application characteristics of GCN on 360° images, and the integration of semantic information of multiple rectilinear images, the proposed model contains six sub-networks. Each sub-network contains two branches, which extract the global features of a single rectilinear image, and integrate the semantic information of six rectilinear images.

III. PROPOSED APPROACH

In this section, the design of our proposed model is described, which contains six sub-networks corresponding to six rectilinear images generated by MCP. Each sub-network contains two branches, one of which is utilized to predict local saliency of 360° images in a refined way, and the other one integrates the semantic information of six rectilinear images. Then, the feature maps obtained by two branches are fused, and the final saliency maps are predicted by the model.

A. Mechanism of SalReGCN360

The framework of SalReGCN360 is illustrated in Figure 2. The network architecture of the proposed model used in training and testing phases is the same, but data is processed differently in each case. In the training phase, firstly, an equirectangular equirectangular saliency map is generated as follow:

\[ r_{S_i} = S_i(r) \quad (i = 1, 2, \ldots, 6). \]

The distortion of the region near the equator of the equirectangular images is much lower than that near the upper and lower boundaries, and the scene contents of the left and right boundaries are discontinuous. Therefore, by utilizing MSR, the positions of objects near the boundaries of images can be shifted to the equator, which enables the model to predict saliency on each region of the 360° images more accurately. Based on this principle, in the model testing phase, MSR is first used to generate multiple new equirectangular images showing the same contents but the positions of objects change. Then, six new equirectangular images \( E_i \quad (i = 1, 2, \ldots, 6) \) are taken as six sub-networks, respectively. In order to predict saliency on any direction of 360° images as much as possible, the positions of objects in six directions of 360° images, namely, front, back, left, right, up and down, are moved to the center of six equirectangular images, respectively. Similarly, for each sub-network, the equirectangular saliency map \( E_{S_i} \) is generated as follows:
Finally, six predicted equirectangular saliency maps are integrated as a fused saliency map by R-MSR. The details of the intra-graph and inter-graph inference modules are described in the next sections.

B. Intra-graph Inference Module

Although increasing the size of the convolutional kernel in the convolutional network can extract the features of a wider range of images, the receptive field with large size can only be achieved in the deep layer of the network if it is to cover the entire equirectangular images. In order to overcome the inherent limitations of convolutional network, and make the model capable of modeling the global relationship in a wide range of images, the intra-graph inference module is adopted based on GCN [15]. First, the feature maps in the coordinate space needs to be mapped to the graph representation in the interaction space, as shown in operation (a) of Figure 3. Given an input feature map \( X \in \mathbb{R}^{L \times C} \) (where \( L = W \times H \), \( W, H, C \) are denoted as the width, height, and the number of channels of the feature map), a projection function \( V = f(x) \in \mathbb{R}^{N \times C'} \) (where \( N \) is denoted as the number of nodes in graph, \( C' \) as the number of channels of each node, and each node can represent the feature information of a region composed of multiple pixels in the images) needs to be learned in coordinate space to make more friendly global inference about disconnected regions of image. The architecture of the intra-graph inference module is illustrated in Figure 4. Before building the projection function, which reduces the number of channels of the input feature map and enhances the capability of the projection function, we need to do the following convolution operations:

\[
X' = \phi(X; W_\phi), \\
B = \theta(X; W_\theta),
\]

where \( X' \in \mathbb{R}^{L' \times C'} \) denotes the feature map after dimensionality reduction, \( B = \mathbb{R}^{N \times L} \) are the learnable weights of the projection function, \( \phi(\cdot) \) and \( \theta(\cdot) \) are the convolution layers, and \( W_\phi \) and \( W_\theta \) are learnable \( 1 \times 1 \) convolution kernels. The projection function can then be represented as follow:

\[
V = f(X') = B X',
\]

where \( V = [v_1, \ldots, v_N] \in \mathbb{R}^{N \times C'} \) denotes a graph whose number of nodes is \( N \), and the number of channels of each node is \( C' \), \( v_i \in \mathbb{R}^{1 \times C'} \) as a feature representation of the \( i \)-th node.

After projecting the feature map from the coordinate space to the interaction space, a graph representation can be obtained. Then an efficient and differentiable GCN [16] can be applied to reason the graph by learning the weight of the edges connecting each pair of nodes. The weights of these edges corresponds to the degree of interaction of potential global feature information for each node. As shown in Figure 4, a two-layer graph convolution operation is carried out for the graph \( V \) containing \( N \) nodes:

\[
Z = (I - A_g) V W_g,
\]

where \( I \) denotes the unit matrix, \( A_g \) is the \( N \times N \) node adjacency matrix, which is used to propagate information between nodes, and \( W_g \) is the channel state update weight of the graph. The first layer of GCN implements Laplace smoothing [17] and propagates the feature information of nodes throughout the entire graph, and thus each node can receive the necessary information from all other nodes. In the training phase, the unit matrix can be utilized as a short connection to alleviate the difficulty of optimization. The adjacency matrix is initialized randomly, and the edge weights are learnt by gradient descent. The second layer of GCN updates the channel state information of each node using the linear transformation method. These two graph convolution operations are processed by two one-dimensional convolution layers along node-level and channel-level directions, respectively.

In order to make the above constructed graph compatible with the existing convolutional network architecture, the graph representation generated in the interaction space needs to be mapped back to the original coordinate space, after the relational inference of each region of 360° images is completed, as illustrated in the operation (b) of Figure 3. Only in this way can subsequent convolutional layers extract more useful features for saliency prediction. As shown in Figure 4, similar to the projection process in the first stage, given the graph representation \( Z \in \mathbb{R}^{N \times C'} \) generated by GCN in the reverse projection, it is necessary to learn a projection function \( Y = g(Z) \) that can project the graph into a feature map \( Y \in \mathbb{R}^{L' \times C'} \).

In fact, in order to reduce the calculation of the module, the projection weight matrix \( B = \mathbb{R}^{N \times L} \) generated in the first stage can be repeatedly applied to this reverse projection. Let \( D = B' = \mathbb{R}^{L' \times N} \), and the projection function \( Y = g(Z) \) can be calculated as follow:

\[
Y = g(Z) = D Z.
\]

After projecting the graph back to the feature map \( Y \) in the original coordinate space, a \( 1 \times 1 \) convolutional layer is added to match the number of channel of the convoluted feature map with that of the original feature map \( X \). In this way, the
C. Inter-graph Inference Module

The above mentioned intra-graph inference module is used to carry out global reasoning on each region, and extracts global features of a single rectilinear image. Since the rectilinear image is only the part of 360° images, each rectilinear image should have a semantic relationship with the others. Therefore, if the intra-graph inference module directly acts on a single rectilinear image and neglects the contextual information of others, then no matter how powerful the module performance is, it cannot effectively predict saliency in some regions of a 360° image. To solve this problem, a module which can extract the contextual features of multiple rectilinear images and carry out global reasoning among them, has been designed, based on the architecture of the intra-graph inference module. This is the inter-graph inference module, whose architecture is illustrated in Figure 6. Just like the intra-graph inference module, the operation of inter-graph inference module is also divided into three stages. First of all, six projection functions \( f_i(X) (i = 1, 2, ..., 6) \) are constructed to map the feature maps \( X_i \in R^{L \times C} (L = H \times W) \) of six sub-networks into the interaction space, respectively. Six graph representations \( V_i \) are obtained, which are shown in operation (a) of Figure 5. Before constructing the projection function, dimension reduction in channel direction of feature maps is needed, and a weight matrix \( B_i \in R^{N \times L} \) of the projection function is constructed:

\[
X'_i = \phi_i(X_i; W_{\phi}), \quad B_i = \theta_i(X_i; W_{\theta}).
\]

where \( X'_i \in R^{L \times C'} \) denotes the feature map after dimensionality reduction, \( \phi_i(\cdot) \) and \( \theta_i(\cdot) \) are the convolutional layers, \( W_{\phi} \) and \( W_{\theta} \) are learnable \( 1 \times 1 \) convolutional kernels. So, the projection function can be calculated as below:

\[
V_i = f_i(X'_i) = B_iX'_i,
\]

where \( V_i = [v_{i1}, ..., v_{iN}] \in R^{N \times C'} \), and \( v_{ij} \in R^{1 \times C'} \) denote the feature representation of the \( j \) th node of the graph, corresponding to the \( i \) th feature map.

The inter-graph inference module acts on global reasoning and regional interaction between rectilinear images, so that the node information of the graph can interact with that of other graphs. All graphs need to be concatenated in the node-level direction (operation (b) of Figure 5):

\[
V_{\text{all}} = \text{Concat}(V_1, ..., V_6) \in R^{6N \times C'}. \]

In this way, a large graph that integrates the node information of six graphs is generated in the interaction space, to represent the global information of a 360° image. Then, GCN can be used to extract the global and contextual features of 360° images, by learning the weights of edges connecting different nodes. As shown in Figure 6, in the interaction space, a two-layer graph convolution operation is carried out on the graph \( V_{\text{all}} \) containing \( 6N \) nodes:

\[
Z_{\text{all}} = ((I - A_g(\text{all}))V_{\text{all}})W_g(\text{all}),
\]

where \( I \) denotes the unit matrix, \( A_g(\text{all}) \), a \( 6N \times 6N \) nodes adjacency matrix which is utilized to propagate information between nodes, \( W_g(\text{all}) \) as the channel state update weight for each node. After the two-layer graph convolution operates in node-level and channel-level directions, each node can receive the contextual information of others, and update its own channel state information at the same time.

Before projecting the large graph generated in the interaction space back to the six feature maps in the original coordinate space, the large graph needs to be split into six sub-graphs in node-level direction (operation (c) of Figure 5). Each sub-graph \( Z_i \in R^{N \times C'} \) is represented as:

\[
Z_1, ..., Z_6 = \text{Split}(Z_{\text{all}}).
\]

In the reverse projection, similar to the projection process in the first stage, the projection functions \( Y_i = g_i(Z_i) \) that can map the sub-graphs into feature maps \( Y_i \in R^{L \times C'} \) need to be learnt. In addition, the projection weights matrix \( B_i \in R^{N \times L} \) generated in the first stage are applied to this reverse projection. Let \( D_i = B_i^T \in R^{L \times N} \), so that the projection function \( Y_i' = g_i(Z_i) \) can be calculated as below:

\[
Y_i' = g_i(Z_i) = D_iZ_i.
\]

Finally, the \( 1 \times 1 \) convolutional layer is applied to the generated feature map to increase the number of channels, and the element-wise addition is carried out with the original feature map \( X_i \). Thus, the original feature maps receive the interactive information with other feature maps, and the proposed model can predict the saliency distribution of 360° images more accurately.
IV. EXPERIMENTAL RESULTS

A. Saliency Datasets

The effectiveness of the proposed model is evaluated on two popular benchmark saliency datasets of 360° images, namely Salient360 (Salient360! Grand Challenge ICME2017) [18] and VR Saliency [19]. Salient360 contains 65 360° images with eye movement labels, 40 of which are utilized for model training and the others for model testing. VR Saliency contains 22 360° images with indoor and outdoor scenes, and provides corresponding saliency maps annotated from 169 subjects’ eye movement data.

B. Implementation Details

Due to the high requirement of hardware equipments in the experiment, the server memory could not meet the condition of model training. Therefore, in all subsequent experiments, the number of sub-networks in the proposed model framework is reduced from six to two. Before training the model on 360° images, we need to pre-train it using the SALICON dataset [20], which contains 15000 traditional 2D annotated images saliency data, 10000 of which are used to train the model and the remaining 5000 to validate the model’s performance. An additional 5000 images (with the labeling results not made public) are also provided for model testing. Since there is no semantic information integration of the rectilinear images during pre-training, the branches with inter-graph inference modules in the sub-networks have to be removed, and only the other branches with intra-graph inference modules are retained. The two sub-networks are trained independently, each of which utilizes the binary cross entropy function as the loss function.

For Salient360, in the fine-tuning and validating phase, 40 360° images are divided into 30 for model fine-tuning and 10 for validation. In the process of MCP, the cube on the outside of the sphere rotates 45 degrees in both horizontal and vertical directions, resulting in $2 \times 2 = 4$ rotations. Therefore, each 360° image can be divided into $4 \times 6 = 24$ rectilinear images, and a total of $30 \times 24 = 720$ rectilinear images can be generated for model fine-tuning with $10 \times 24 = 240$ for validation. In order to ensure that the inter-graph inference module can extract the contextual features of two adjacent rectilinear images, the images chosen from two adjacent cube faces are put into two sub-networks simultaneously. We take a set of rectilinear images shown in Figure 7 as an example. The rectilinear images A to F represent six faces of one of the cubes, in which image A is connected to B, D, E and F, and image B to C, E, F except for A. In a similar fashion, a single cube can generate 12 image groups, namely A-B, A-D, A-E, A-F, B-C, B-E, B-F, C-D, C-E, C-F and D-F. In this way, $30 \times 4 \times 12 = 1440$ image groups are generated for fine-tuning and $15 \times 4 \times 12 = 480$ for validation. In addition, the parameters of the backbone networks are fixed during fine-tuning, but that of the inter-graph inference module and the last two convolutional layers of decoder are initialized. The sum of the two binary cross entropy functions used in the two sub-networks is taken as the total loss function of the proposed model.

When performing MSR during the model testing phase, we keep the rotation angle 0° on the Y axis and rotate by -30°, 0°, 30° on the X axis, respectively. For each rotation angle on the X axis, we rotate 45° × $k (k = 0,...,7)$ on the Z axis. By doing so, $3 \times 8 = 24$ equirectangular images are generated, which is exactly the same number of rectilinear images generated by MCP. Therefore, a total of $25 \times 24 = 600$ equirectangular images are generated by MSR. Following that, two equirectangular images showing the same scene, but with objects in different positions, are taken as an image group. As shown in Figure 8, taking rectilinear image A in Figure 7 as an example, we firstly find an equirectangular image where objects on image A are located at the center of such that. Since saliency on the region with less distortion near the equator of equirectangular image can be predicted more accurately, we later find a set of equirectangular images where objects on image B, D, E, F (adjacent to image A in the cube) are located at the center of such these. Therefore, there are 4 testing image groups, and $12 \times 4 = 48$ image groups can be generated from an equirectangular image. For the entire testing dataset, a total of $48 \times 25 = 1200$ image groups can be generated. After 48 image groups are put into two sub-networks, and 48 corresponding saliency map groups are generated separately, the reverse rotation operation on $48 \times 2 = 96$ saliency maps with the same parameter configuration is performed by R-MSR. Finally, the saliency maps obtained by R-MSR are averaged to generate the fused saliency map.

Fig. 7. Six rectilinear images generated by cubic projection.

Fig. 8. Four equirectangular image groups generated by MSR.
equirectangular images and $12 \times 4 \times 22 = 1056$ image groups are generated under the same experimental setup as those in MSR and R-MSR for Salient360.

In the pre-training and fine-tuning phase, the Adam optimizer [21] is applied to optimize the proposed model. The learning rate is initialized as $1.5 \times 10^{-4}$ and the batch size is set to 6. SalReGCN360 was modeled in the PyTorch framework [22] and deployed on the GeForce GTX1080Ti GPU.

### C. Effectiveness of the Intra-graph Inference Module

In order to verify the effectiveness of the intra-graph inference module, all of the modules are removed in the proposed model. Table I shows the experimental quantitative results of whether SalReGCN360 has intra-graph inference module or not on the two datasets. It can be intuitively seen that the proposed model with these modules is superior to that without them on all metrics. In addition, the performance of SalReGCN360 on metrics KLD and CC is improved without them on all metrics. In particular, metrics NSS and AUC have a higher scores when the graph inference module is not as accurate as intra-graph inference module or not on the two datasets.

<table>
<thead>
<tr>
<th>Settings</th>
<th>Salient360</th>
<th>VR Saliency</th>
</tr>
</thead>
<tbody>
<tr>
<td>KLD</td>
<td>CC</td>
<td>NSS</td>
</tr>
<tr>
<td>w.o. Intra-graph IM</td>
<td>0.355</td>
<td>0.667</td>
</tr>
<tr>
<td>Intra-graph IM</td>
<td>0.324</td>
<td>0.684</td>
</tr>
</tbody>
</table>

### D. Effectiveness of the Inter-graph Inference Module

Similar to the setting of the above comparative experiment, all of the inter-graph inference modules in the proposed model were removed. Table II shows the experimental quantitative results of whether SalReGCN360 has inter-graph inference module or not on the two datasets. Similarly, the overall performance of the proposed model is improved by adding the inter-graph inference modules. In particular, metrics NSS and AUC have a higher scores, which strongly indicates that inter-graph inference module can effectively reduce the loss of saliency information in local regions of 360° images.

<table>
<thead>
<tr>
<th>Settings</th>
<th>Salient360</th>
<th>VR Saliency</th>
</tr>
</thead>
<tbody>
<tr>
<td>KLD</td>
<td>CC</td>
<td>NSS</td>
</tr>
<tr>
<td>w.o. Inter-graph IM</td>
<td>0.333</td>
<td>0.679</td>
</tr>
<tr>
<td>Inter-graph IM</td>
<td>0.324</td>
<td>0.684</td>
</tr>
</tbody>
</table>

### Effectiveness of the Proposed Model Framework

To verify the effectiveness of the proposed model framework, the rectilinear images can be intuitively seen that the proposed model with these modules is superior to that

<table>
<thead>
<tr>
<th>Method</th>
<th>KLD</th>
<th>CC</th>
<th>NSS</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>GBVS360 [10]</td>
<td>0.698</td>
<td>0.527</td>
<td>0.851</td>
<td>0.714</td>
</tr>
<tr>
<td>SalNet360 [12]</td>
<td>0.458</td>
<td>0.548</td>
<td>0.755</td>
<td>0.701</td>
</tr>
<tr>
<td>Maugey et al [23]</td>
<td>0.585</td>
<td>0.448</td>
<td>0.506</td>
<td>0.644</td>
</tr>
<tr>
<td>Startsev et al [24]</td>
<td>0.420</td>
<td>0.620</td>
<td>0.810</td>
<td>0.720</td>
</tr>
<tr>
<td>SJTU [25]</td>
<td>0.481</td>
<td>0.532</td>
<td>0.918</td>
<td>0.734</td>
</tr>
<tr>
<td>CDSR [6]</td>
<td>0.477</td>
<td>0.550</td>
<td>0.939</td>
<td>0.736</td>
</tr>
<tr>
<td>SalGCN [13]</td>
<td>0.428</td>
<td>0.589</td>
<td>0.945</td>
<td>0.736</td>
</tr>
<tr>
<td>SalGAN360 [4]</td>
<td>0.431</td>
<td>0.659</td>
<td>0.971</td>
<td>0.746</td>
</tr>
<tr>
<td>SalBiNet360 [26]</td>
<td>0.402</td>
<td>0.661</td>
<td>0.975</td>
<td>0.746</td>
</tr>
<tr>
<td>SalReGCN360</td>
<td>0.324</td>
<td>0.684</td>
<td>0.979</td>
<td>0.749</td>
</tr>
</tbody>
</table>

In the training phase, since there is only one region and the branch with the inter-graph inference modules are removed, leaving a sub-network with only the intra-graph inference modules. In the training phase, since there is only one sub-network in the model framework, the rectilinear images can only be put into the sub-network separately, and the model cannot integrate the semantic information of multiple images.

Similarly, in the testing phase, multiple equirectangular images obtained by MSR are also put into the sub-network separately, and the generated saliency maps are averaged to obtain the fused saliency maps. The comparative experimental results of the proposed model framework with a single and two sub-networks are shown in Table III. It can be found from the score changes of each metric that the framework with two sub-networks and inter-graph inference module can improve the performance of SalReGCN360 as a whole. This illustrates that the proposed model can perform better by constructing multiple sub-networks to integrate semantic information of multiple rectilinear images.

### E. Effectiveness of the Proposed Model Framework

The framework of SalReGCN360 mainly integrates the semantic information of multiple rectilinear images by putting several semantically related rectilinear images into corresponding sub-networks, so as to predict saliency on regions of 360° images more accurately. In order to verify the effectiveness of the proposed model framework, one of the sub-networks and the branch with the inter-graph inference modules are removed, leaving a sub-network with only the intra-graph inference modules. In the training phase, since there is only one sub-network in the model framework, the rectilinear images can only be put into the sub-network separately, and the model cannot integrate the semantic information of multiple images.

Similarly, in the testing phase, multiple equirectangular images obtained by MSR are also put into the sub-network separately, and the generated saliency maps are averaged to obtain the fused saliency maps. The comparative experimental results of the proposed model framework with a single and two sub-networks are shown in Table III. It can be found from the score changes of each metric that the framework with two sub-networks and inter-graph inference module can improve the performance of SalReGCN360 as a whole. This illustrates that the proposed model can perform better by constructing multiple sub-networks to integrate semantic information of multiple rectilinear images.

### Table I

<table>
<thead>
<tr>
<th>Settings</th>
<th>Salient360</th>
<th>VR Saliency</th>
</tr>
</thead>
<tbody>
<tr>
<td>KLD</td>
<td>CC</td>
<td>NSS</td>
</tr>
<tr>
<td>w.o. Intra-graph IM</td>
<td>0.355</td>
<td>0.667</td>
</tr>
<tr>
<td>Intra-graph IM</td>
<td>0.324</td>
<td>0.684</td>
</tr>
</tbody>
</table>

### Table II

<table>
<thead>
<tr>
<th>Settings</th>
<th>Salient360</th>
<th>VR Saliency</th>
</tr>
</thead>
<tbody>
<tr>
<td>KLD</td>
<td>CC</td>
<td>NSS</td>
</tr>
<tr>
<td>w.o. Inter-graph IM</td>
<td>0.333</td>
<td>0.679</td>
</tr>
<tr>
<td>Inter-graph IM</td>
<td>0.324</td>
<td>0.684</td>
</tr>
</tbody>
</table>

### Table III

<table>
<thead>
<tr>
<th>Settings</th>
<th>Salient360</th>
<th>VR Saliency</th>
</tr>
</thead>
<tbody>
<tr>
<td>KLD</td>
<td>CC</td>
<td>NSS</td>
</tr>
<tr>
<td>Single Sub-network</td>
<td>0.330</td>
<td>0.676</td>
</tr>
<tr>
<td>Two Sub-networks</td>
<td>0.324</td>
<td>0.684</td>
</tr>
</tbody>
</table>

latter in identifying eye movement points in 360° images. In this way, SalReGCN360 can learn which regions of 360° images obtain higher saliency values from a global perspective as well as reduce redundant saliency information to some extent, and thus improve the accuracy of model prediction.
F. Comparison to the State of The Art

To evaluate the effectiveness of SalReGCN360 in saliency prediction on 360° images, we compared four metric scores of the proposed model with those of several public and tested saliency prediction models on 360° images, based on Salient360 and VR Saliency datasets. The quantitative results of all models tested on Salient360 are shown in Table IV. SalReGCN360 outperforms the existing models in all metrics. In particular, our proposed model outperforms SalBiNet360 using the KLD and CC metrics, which indicates that the degree of fitting between the predicted saliency maps and the ground truths is further improved, further reducing the loss of saliency information. Quantitative results of VR Saliency dataset are shown in Table V. Although the performance of SalReGCN360 is not significantly improved on this dataset, it still outperforms the other existing saliency prediction models, including SalBiNet360.

In order to highlight the improved performance of SalReGCN360, Figure 9 illustrates the visual comparison of the saliency maps predicted by the proposed model, and those from SalBiNet360, SalGAN360 and SalGCN on Salient360. It can be seen that the salient regions predicted by SalReGCN360 and the other models are mainly concentrated in the equator of equirectangular images. However, the saliency predicted by...
other models is relatively course: that is, saliency is marked in many regions, but its magnitude has no clear limit. Moreover, there is much redundant saliency information, so it is difficult to clearly show which regions are more likely to attract the attention of observers. In contrast, our proposed model can finely predict the regions or objects with high saliency, and the boundaries between regions with and without saliency are shown clearly. In addition, under the effectiveness of MSR, SalReGCN360 is able to predict saliency on regions near the boundaries of the equirectangular images to some extent, so as to illustrate the characteristics of the saliency distribution of 360° images more comprehensively.

Visual comparisons with SalReGCN360 and SalBiNet360 on VR Saliency dataset are shown in Fig. 10. Compared with SalBiNet360, SalReGCN360 makes a greater contribution in reducing redundant saliency information and predicting saliency on each region of the 360° images more accurately. This is mainly illustrated by the clear reduction of redundant information on regions near the upper and lower boundaries of the 360° images, and in addition, the boundaries between regions marked with high and low saliency are definitely clear.

V. CONCLUSION

In this paper, a new saliency prediction model for 360° images, based on GCN, is proposed, which we named SalReGCN360. The model contains six sub-networks, each of which contains two branches. In one of the branches, the intra-graph inference module extracts global features of a single rectilinear image to predict the local saliency of 360° images more accurately. In the other branch, the inter-graph inference module extracts contextual features of 360° images by integrating semantic information of multiple rectilinear images. In the testing phase, MSR is applied to effectively predict saliency on each region of the 360° images. Experimental results have verified the effectiveness of each component and SalReGCN360 achieves the best performance on two publicly available datasets.

REFERENCES