Edge-aware Graph Representation Learning and Reasoning for Face Parsing

Gusi Te[1], Yinglu Liu[2], Wei Hu[1*], Hailin Shi[2], and Tao Mei[2]

[1] Wangxuan Institute of Computer Technology, Peking University, Beijing, China
{tegusi, forhuwei}@pku.edu.cn
[2] JD AI Research, Beijing, China
{liuyinglu1,shihailin,tmei}@jd.com

Abstract. Face parsing infers a pixel-wise label to each facial component, which has drawn much attention recently. Previous methods have shown their efficiency in face parsing, which however overlook the correlation among different face regions. The correlation is a critical clue about the facial appearance, pose, expression, *etc.*, and should be taken into account for face parsing. To this end, we propose to model and reason the region-wise relations by learning graph representations, and leverage the edge information between regions for optimized abstraction. Specifically, we encode a facial image onto a global graph representation where a collection of pixels ("regions") with similar features are projected to each vertex. Our model learns and reasons over relations between the regions by propagating information across vertices on the graph. Furthermore, we incorporate the edge information to aggregate the pixel-wise features onto vertices, which emphasizes on the features around edges for fine segmentation along edges. The finally learned graph representation is projected back to pixel grids for parsing. Experiments demonstrate that our model outperforms state-of-the-art methods on the widely used Helen dataset, and also exhibits the superior performance on the large-scale CelebAMask-HQ and LaPa dataset. The code is available at https://github.com/tegusi/EAGRNet.

Keywords: Face parsing, graph representation, attention mechanism, graph reasoning

1 Introduction

Face parsing assigns a pixel-wise label to each semantic component, such as facial skin, eyes, mouth and nose, which is a particular task in semantic segmentation. It has been applied in a variety of scenarios such as face understanding, editing, synthesis, and animation [1,2,3].

The region-based methods have been recently proposed to model the facial components separately [4,5,6], and achieved state-of-the-art performance on the

[*] Corresponding author: Wei Hu (forhuwei@pku.edu.cn). This work was in collaboration with JD AI Research during Gusi Te's internship there.

current benchmarks. However, these methods are based on the individual information within each region, and the correlation among regions is not exploited yet to capture long range dependencies. In fact, facial components present themselves with abundant correlation between each other. For instance, eyes, mouth and eyebrows will generally become more curvy when people smile; facial skin and other components will be dark when the lighting is weak, and so on.

The correlation between the facial components is the critical clue in face representation, and should be taken into account in the face parsing. To this end, we propose to learn graph representations over facial images, which model the relations between regions and enable reasoning over non-local regions to capture long range dependencies. To bridge the facial image pixels and graph vertices, we project a collection of pixels (a "region") with similar features to each vertex. The pixel-wise features in a region are aggregated to the feature of the corresponding vertex. In particular, to achieve accurate segmentation along the edges between different components, we propose the edge attention in the pixel-to-vertex projection, assigning larger weights to the features of edge pixels during the feature aggregation. Further, the graph representation learns the relations between facial regions, *i.e.*, the graph connectivity between vertices, and reasons over the relations by propagating information across all vertices on the graph, which is able to capture long range correlations in the facial image. The learned graph representation is finally projected back to the pixel grids for face parsing. Since the number of vertices is significantly smaller than that of pixels, the graph representation also reduces redundancy in features as well as computational complexity effectively.

Specifically, given an input facial image, we first encode the high-level and low-level feature maps by the ResNet backbone [7]. Then, we build a projection matrix to map a cluster of pixels with similar features to each vertex. The feature of each vertex is taken as the weighted aggregation of pixel-wise features in the cluster, where features of edge pixels are assigned with larger weights via an edge mask. Next, we learn and reason over the relations between vertices (*i.e.*, regions) via graph convolution [8,9] to further extract global semantic features. The learned features are finally projected back to a pixel-wise feature map. We test our model on Helen, CelebAMask-HQ and LaPa datasets, and surpass state-of-the-art methods.

Our main contributions are summarized as follows.

- We propose to exploit the relations between regions for face parsing by modeling on a region-level graph representation, where we project a collection of pixels with similar features to each vertex and reason over the relations to capture long range dependencies.
- We introduce edge attention in the pixel-to-vertex feature projection, which emphasizes on features of edge pixels during the feature aggregation to each vertex and thus enforces accurate segmentation along edges.
- We conduct extensive experiments on Helen, CelebAMask-HQ and LaPa datasets. The experimental results show our model outperforms state-of-the-art methods on almost every category.

2 Related Work

2.1 Face Parsing

Face parsing is a division of semantic segmentation, which assigns different labels to the corresponding regions on human faces, such as nose, eyes, mouth and *etc.*. The methods of face parsing could be classified into global-based and local-based methods.

Traditionally, hand crafted features including SIFT [10] are applied to model the facial structure. Warrell *et al.* describe spatial relationship of facial parts with epitome model [11]. Kae *et al.* combine Conditional Random Field (CRF) with a Restricted Boltzmann Machine (RBM) to extract local and global features [12]. With the rapid development of machine learning, CNN has been introduced to learn more robust and rich features. Liu *et al.* import CNN-based features into the CRF framework to model individual pixel labels and neighborhood dependencies [13]. Luo *et al.* propose a hierarchical deep neural network to extract multi-scale facial features [14]. Zhou *et al.* adopt adversarial learning approach to train the network and capture high-order inconsistency [15]. Liu *et al.* design a CNN-RNN hybrid model that benefits from both high quality features of CNN and non-local properties of RNN [6]. Zhou *et al.* present an interlinked CNN that takes multi-scale images as input and allows bidirectional information passing [16]. Lin *et al.* propose a novel RoI Tanh-Warping operator preserving central and peripheral information. It contains two branches with the local-based for inner facial components and the global based for outer facial ones. This method shows high performance especially on hair segmentation [4].

2.2 Attention Mechanism

Attention mechanism has been proposed to capture long-range information [17], and applied to many applications such as sentence encoding [18] and image feature extraction [19]. Limited by the locality of convolution operators, CNN lacks the ability to model global contextual information. Furthermore, Chen *et al.* propose Double Attention Model that gathers information spatially and temporally to improve complexity of traditional non-local modules [20]. Zhao *et al.* propose a point-wise spatial attention module, relaxing the local neighborhood constraint [21]. Zhu *et al.* also present an asymmetric module to reduce abundant computation and distillate features [22]. Fu *et al.* devise a dual attention module that applies both spatial and channel attention in feature maps [23]. To research underlying relationship between different regions, Chen *et al.* project original features into interactive space and utilize GCN to exploit high order relationship [24]. Li *et al.* devise a robust attention module that incorporates the Expectation-Maximization algorithm [25].

2.3 Graph Reasoning

Interpreting images from the graph perspective is an interesting idea, since an image could be regarded as regular pixel grids. Chandra *et al.* propose Condi-

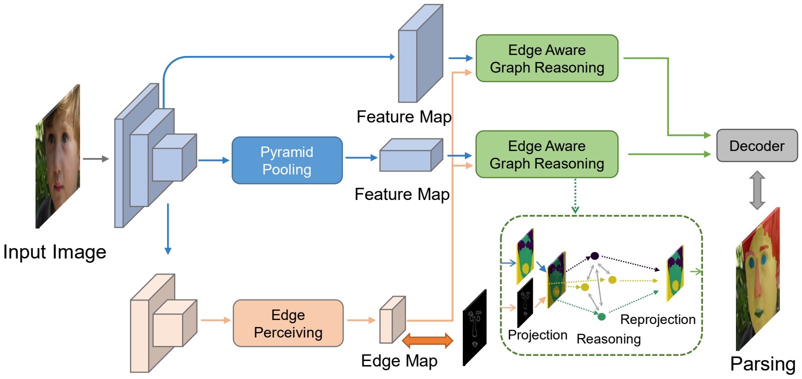

Fig. 1. The overview of the proposed face parsing framework.

tional Random Field (CRF) based method on image segmentation [26]. Besides, graph convolution network (GCN) is imported into image segmentation. Li *et al.* introduce graph convolution to the semantic segmentation, which projects features into vertices in the graph domain and applies graph convolution afterwards [27]. Furthermore, Lu *et al.* propose Graph-FCN where semantic segmentation is reduced to vertex classification by directly transforming an image into regular grids [28]. Pourian *et al.* propose a method of semi-supervised segmentation [29]. The image is divided into community graph and different labels are assigned to corresponding communities. Te *et al.* propose a computation-efficient and posture-invariant face representation with only a few key points on hypergraphs for face anti-spoofing beyond 2D attacks [30]. Zhang *et al.* utilize graph convolution both in the coordinate space and feature space [31].

3 Methods

3.1 Overview

As illustrated in Fig. 1, given an input facial image, we aim to predict the corresponding parsing label and auxiliary edge map. The overall framework of our method consists of three procedures as follows.

- **Feature and Edge Extraction.** We take ResNet as the backbone to extract features at various levels for multi-scale representation. The low-level features contain more details but lack semantic information, while the high-level features provide rich semantics with global information at the cost of image details. To fully exploit the global information in high-level features, we employ a spatial pyramid pooling operation to learn multi-scale contextual information. Further, we construct an edge perceiving module to acquire an edge map for the subsequent module.

- **Edge Aware Graph Reasoning.** We feed the feature map and edge map into the proposed Edge Aware Graph Reasoning (EAGR) module, aiming to learn intrinsic graph representations for the characterization of the relations between regions. The EAGR module consists of three operations: graph projection, graph reasoning and graph reprojection, which projects the original features onto vertices in an edge-aware fashion, reasons the relations between vertices (regions) over the graph and projects the learned graph representation back to pixel grids, leading to a refined feature map with the same size.
- **Semantic Decoding.** We fuse the refined features into a decoder to predict the final result of face parsing. The high-level feature map is upsampled to the same dimension as the low-level one. We concatenate both feature maps and leverage 1 × 1 convolution layer to reduce feature channels, predicting the final parsing labels.

3.2 Edge-Aware Graph Reasoning

Inspired by the non-local module [19], we aim to build the long-range interactions between distant regions, which is critical for the description of the facial structure. In particular, we propose edge-aware graph reasoning to model the long-range relations between regions on a graph, which consists of edge-aware graph projection, graph reasoning and graph reprojection.

Edge-Aware Graph Projection We first revisit the typical non-local modules. Given a feature map $\mathbf{X} \in \mathbb{R}^{HW \times C}$, where H and W refer to the height and width of the input image respectively and C is the number of feature channels. A typical non-local module is formulated as:

$$\widetilde{\mathbf{X}} = \text{softmax}\left(\theta(\mathbf{X})\varphi^{\top}(\mathbf{X})\right)\gamma(\mathbf{X}) = \mathbf{V}\gamma(\mathbf{X}), \tag{1}$$

where θ, φ and γ are convolution operations with 1×1 kernel size. $\mathbf{V} \in \mathbb{R}^{HW \times HW}$ is regarded as the attention maps to model the long-range dependencies. However, the complexity of computing $\mathbf{V}$ is $\mathcal{O}(H^2W^2C)$, which does not scale well with increasing number of pixels HW. To address this issue, we propose a simple yet effective edge-aware projection operation to eliminate the redundancy in features.

Given an input feature map $\mathbf{X} \in \mathbb{R}^{HW \times C}$ and an edge map $\mathbf{Y} \in \mathbb{R}^{HW \times 1}$, we construct a projection matrix $\mathbf{P}$ by mapping $\mathbf{X}$ onto vertices of a graph with $\mathbf{Y}$ as a prior. Specifically, we first reduce the dimension of $\mathbf{X}$ in the feature space via a convolution operation φ with 1×1 kernel size, leading to $\varphi(\mathbf{X}) \in \mathbb{R}^{HW \times T}$, $T < C$. Then, we duplicate the edge map Y to the same dimension of $\varphi(\mathbf{X})$ for ease of computation. We incorporate the edge information into the projection, by taking the Hadamard Product of $\varphi(\mathbf{X})$ and $\mathbf{Y}$. As the edge map $\mathbf{Y}$ encodes the probability of each pixel being an edge pixel, the Hadamard Product operation essentially assigns a weight to the feature of each pixel, with larger weights to features of edge pixels. Further, we introduce an average *pooling* operation $\mathcal{P}(\cdot)$

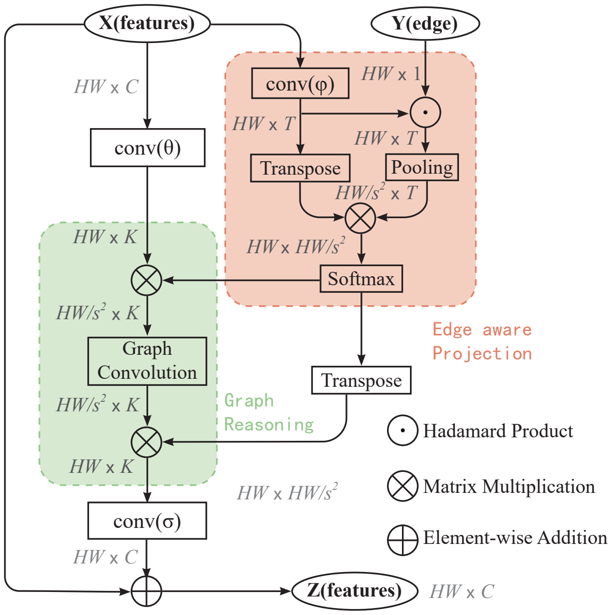

Fig. 2. Architecture of the Edge Aware Graph Reasoning module.

with stride s to obtain anchors of vertices. These anchors represent the centers of each region of pixels, and we take the multiplication of $\varphi(\mathbf{X})$ and anchors to capture the similarity between anchors and each pixel. We then apply a softmax function for normalization. Formally, the projection matrix takes the form:

$$\mathbf{P} = \mathrm{softmax}\left(\mathcal{P}(\varphi(\mathbf{X}) \odot \mathbf{Y}) \cdot \varphi(\mathbf{X})^{\top}\right), \tag{2}$$

where $\odot$ denotes the Hadamard product, and $\mathbf{P} \in \mathbb{R}^{HW/s^2 \times HW}$.

In Eq. (2), we have two critical operations: the edge attention and the pooling operation. The edge attention emphasizes the features of edge pixels by assigning larger weights to edge pixels. Further, we propose the pooling operation in the features, whose benefits are in twofold aspects. On one hand, the pooling leads to compact representations by averaging over features to remove the redundancy. On the other hand, by pooling with stride s, the computation complexity is reduced from $\mathcal{O}(H^2W^2C)$ in non-local modules to $\mathcal{O}(H^2W^2C/s^2)$.

With the acquired projection matrix $\mathbf{P}$, we project the pixel-wise features $\mathbf{X}$ onto the graph domain, *i.e.*,

$$\mathbf{X}_G = \mathbf{P}\theta(\mathbf{X}), \tag{3}$$

where θ is a a convolution operation with 1×1 kernel size so as to reduce the dimension of $\mathbf{X}$, resulting in $\theta(\mathbf{X}) \in \mathbb{R}^{HW \times K}$. The projection aggregates pixels with similar features as each anchor to one vertex, thus each vertex essentially represents a region in the facial images. Hence, we bridge the connection between pixels and each region via the proposed edge aware graph projection, leading to the features of the projected vertices on the graph $\mathbf{X}_G \in \mathbb{R}^{HW/s^2 \times K}$ via Eq. (3).

Graph Reasoning Next, we learn the connectivity between vertices from $\mathbf{X}_G$, *i.e.*, the relations between regions. Meanwhile, we reason over the relations by propagating information across vertices to learn higher-level semantic information. This is elegantly realized by a single-layer Graph Convolution Network (GCN). Specifically, we feed the input vertex features $\mathbf{X}_G$ into a first-order approximation of spectral graph convolution. The output feature map $\hat{\mathbf{X}}_G \in \mathbb{R}^{HW/s^2 \times K}$ is

$$\hat{\mathbf{X}}_G = \mathrm{ReLU}\left[(\mathbf{I} - \mathbf{A})\mathbf{X}_G\mathbf{W}_G\right] = \mathrm{ReLU}\left[(\mathbf{I} - \mathbf{A})\mathbf{P}\theta(\mathbf{X})\mathbf{W}_G\right], \tag{4}$$

where $\mathbf{A}$ denotes the adjacent matrix that encodes the graph connectivity to learn, $\mathbf{W}_G \in \mathbb{R}^{K \times K}$ denotes the weights of the GCN, and ReLU is the activation function. The features $\hat{\mathbf{X}}_G$ are acquired by the vertex-wise interaction (multiplication with $(\mathbf{I} - \mathbf{A})$) and channel-wise interaction (multiplication with $\mathbf{W}_G$).

Different from the original one-layer GCN [32] in which the graph $\mathbf{A}$ is handcrafted, we randomly initialize $\mathbf{A}$ and learn from vertex features. Moreover, we add a residual connection to reserve features of raw vertices. Based on the learned graph, the information propagation across all vertices leads to the finally reasoned relations between regions. After graph reasoning, pixels embedded within one vertex share the same context of features modeled by graph convolution. We set the same number of output channels as the input to keep consistency, allowing the module to be compatible with the subsequent process.

Graph Reprojection In order to fit into existing framework, we reproject the extracted vertex features in the graph domain to the original pixel grids. Given the learned graph representation $\hat{\mathbf{X}}_G \in \mathbb{R}^{HW/s^2 \times K}$, we aim to compute a matrix $\mathbf{V} \in \mathbb{R}^{HW \times HW/s^2}$ that maps $\hat{\mathbf{X}}_G$ to the pixel space. In theory, $\mathbf{V}$ could be taken as the inverse of the projection matrix $\mathbf{P}$. However, it is nontrivial to compute because $\mathbf{P}$ is not a square matrix. To tackle this problem, we take the transpose matrix $\mathbf{P}^\top$ as the reprojection matrix [27], in which $\mathbf{P}_{ij}^\top$ reflects the correlation between vertex i and pixel j. The limitation of this operation is that the row vectors in $\mathbf{P}^\top$ are not normalized.

After reprojection, we deploy a 1×1 convolution operation σ to increase the feature channels in consistent with the input features $\mathbf{X}$. Then, we take the summation of the reprojected refined features and the original feature map as the final features. The final pixel-wise feature map $\mathbf{Z} \in \mathbb{R}^{HW \times C}$ is thus computed by

$$\mathbf{Z} = \mathbf{X} + \sigma(\mathbf{P}^\top \hat{\mathbf{X}}_G). \tag{5}$$

3.3 The Loss Function

To further strengthen the effect of the proposed edge aware graph reasoning, we introduce the boundary-attention loss (BA-Loss) inspired by [33] besides the traditional cross entropy loss for predicted parsing maps and edge maps. The BA-loss computes the loss between the predicted label and the ground truth only at edge pixels, thus improving the segmentation accuracy of critical edge pixels that are difficult to distinguish. Mathematically, the BA-loss is written as

$$\mathcal{L}_{\mathrm{BA}} = \sum_{i=1}^{HW} \sum_{j=1}^{N} [e_i = 1]\, y_{ij} \log p_{ij}, \tag{6}$$

where i is the index of pixels, j is the index of classes and N is the number of classes. e_i denotes the edge label, y_{ij} denotes the ground truth label of face parsing, and p_{ij} denotes the predicted parsing label. $[\cdot]$ is the Iverson bracket, which denotes a number that is 1 if the condition in the bracket is satisfied, and 0 otherwise.

The total loss function is then defined as follows:

$$\mathcal{L} = \mathcal{L}_{\mathrm{parsing}} + \lambda_1 \mathcal{L}_{\mathrm{edge}} + \lambda_2 \mathcal{L}_{\mathrm{BA}}, \tag{7}$$

where $\mathcal{L}_{\mathrm{parsing}}$ and $\mathcal{L}_{\mathrm{edge}}$ are classical cross entropy losses for the parsing and edge maps. λ_1 and λ_2 are two hyper-parameters to strike a balance among the three loss functions.

3.4 Analysis

Since non-local modules and graph-based methods have drawn increasing attention, it is interesting to analyze the similarities and differences between previous works and our method.

Comparison with non-local modules Typically, a traditional non-local module models *pixel-wise* correlations by feature similarities. However, the high-order relationship between regions are not captured. In contrast, we exploit the correlation among distinct regions via the proposed graph projection and reasoning. The features of each vertex embed not only local contextual anchor aggregated by average pooling in a certain region but also global features from the overall pixels. We further learn and reason over the relations between regions by graph convolution, which captures high-order semantic relations between different facial regions.

Also, the computation complexity of non-local modules is expensive in general as discussed in Section 3.2. Our proposed edge-aware pooling addresses the issue by extracting significant anchors to replace redundant query points. Also, we do not incorporate pixels within each facial region during the sampling process while focusing on edge pixels, thus improving boundary details. The intuition is that pixels within each region tend to share similar features.

Comparison with graph-based models In comparison with other graph-based models, such as [24,27], we improve the graph projection process by introducing locality in sampling in particular. In previous works, each vertex is simply represented by a weighted sum of image pixels, which does not consider edge information explicitly and brings ambiguity in understanding vertices. Besides, with different inputs of feature maps, the pixel-wise features often vary greatly but the projection matrix is fixed after training. In contrast, we incorporate the edge information into the projection process to emphasize on edge pixels, which preserves boundary details well. Further, we specify vertex anchors locally based on the average pooling, which conforms with the rule that the location of facial components keeps almost unchanged after face alignment.

4 Experiments

4.1 Datasets and Metrics

The Helen dataset includes 2,330 images with 11 categories: background, skin, left/right brow, left/right eye, upper/lower lip, inner mouth and hair. Specifically, we keep the same train/validation/test protocol as in [34]. The number of the training, validation and test samples are 2,000, 230 and 100, respectively. The CelebAMask-HQ dataset is a large-scale face parsing dataset which consists of 24,183 training images, 2,993 validation images and 2,824 test images. The number of categories in CelebAMask-HQ is 19. In addition to facial components, the accessories such as eyeglass, earring, necklace, neck, and cloth are also annotated in the CelebAMask-HQ dataset. The LaPa dataset is a newly released challenging dataset for face parsing, which contains 11 categories as Helen, covering large variations in facial expression, pose and occlusion. It consists of 18,176 training images, 2,000 validation images and 2,000 test images.

During training, we use the rotation and scale augmentation. The rotation angle is randomly selected from $(-30^\circ, 30^\circ)$ and the scale factor is randomly selected from $(0.75, 1.25)$. The edge mask is extracted according to the semantic label map. If the label of a pixel is different with its 4 neighborhoods, it is regarded as a edge pixel. For the Helen dataset, similar to [4], we implement face alignment as a pre-processing step and the results are re-mapped to the original image for evaluation.

We employ three evaluation metrics to measure the performance of our model: pixel accuracy, mean intersection over union (mIoU) and F1 score. Directly employing the accuracy metric ignores the scale variance amid facial components, while the mean IoU and F1 score are better for evaluation. To keep consistent with the previous methods, we report the overall F1-score on the Helen dataset, which is computed over the merged facial components: brows (left+right), eyes (left+right), nose, mouth (upper lip+lower lip+inner mouth). For the CelebAMask-HQ and LaPa datasets, the mean F1-score over all categories excluding background is employed.

Table 1. Ablation study on the Helen dataset.

Model	Baseline	Edge	Graph	Reasoning	BA-loss	mIoU	F1-score	Accuracy
1	✓					76.5	91.4	85.9
2	✓	✓				77.5	92.0	86.2
3	✓		✓	✓		77.3	92.3	85.8
4	✓	✓	✓	✓		77.8	92.4	84.6
5	✓	✓	✓		✓	77.3	92.3	86.7
6	✓	✓	✓	✓	✓	78.2	92.8	87.3

4.2 Implementation Details

Our backbone is a modified version of the ResNet-101 [7] excluding the average pooling layer, and the Conv1 block is changed to three 3×3 convolutional layers. For the pyramid pooling module, we follow the implementation in [35] to exploit global contextual information. The pooling factors are $\{1, 2, 3, 6\}$. Similar to [36], the edge perceiving module predicts a two-channel edge map based on the outputs of Conv2, Conv3 and Conv4 in ResNet-101. The outputs of Conv1 and the pyramid pooling serve as the low-level and high-level feature maps, respectively. Both of them are fed into the EAGR module separately for graph representation learning.

As for the EAGR module, we set the pooling size to 6×6. To pay more attention on the facial components, we just utilize the central 4×4 anchors for graph construction. The feature dimensions K and T are set to 128 and 64, respectively.

Stochastic Gradient Descent (SGD) is employed for optimization. We initialize the network with a pretrained model on ImageNet. The input size is 473×473 and the batch size is set to 28. The learning rate starts at 0.001 with the weight decay of 0.0005. The batch normalization is implemented with In-Place Activated Batch Norm [37].

4.3 Ablation study

On different components We demonstrate the effectiveness of different components in the proposed EAGR module. Specifically, we remove some components and train the model from scratch under the same initialization. The quantitative results are reported in Table 1. *Baseline* means the model only utilizes the ResNet backbone, pyramid pooling and multi-scale decoder without any EGAR module, and *Edge* represents whether edge aware pooling is employed. *Graph* represents the EAGR module, while *Reasoning* indicates the graph reasoning excluding graph projection and reprojection. We observe that *Edge* and *Graph* lead to improvement over the baseline by 1% in mIoU respectively. When both components are taken into account, we achieve even better performance. The boundary-attention loss (BA-loss) also leads to performance improvement.

We also provide subjective results of face parsing from different models in Fig. 3. Results of incomplete models exhibit varying degrees of deficiency around

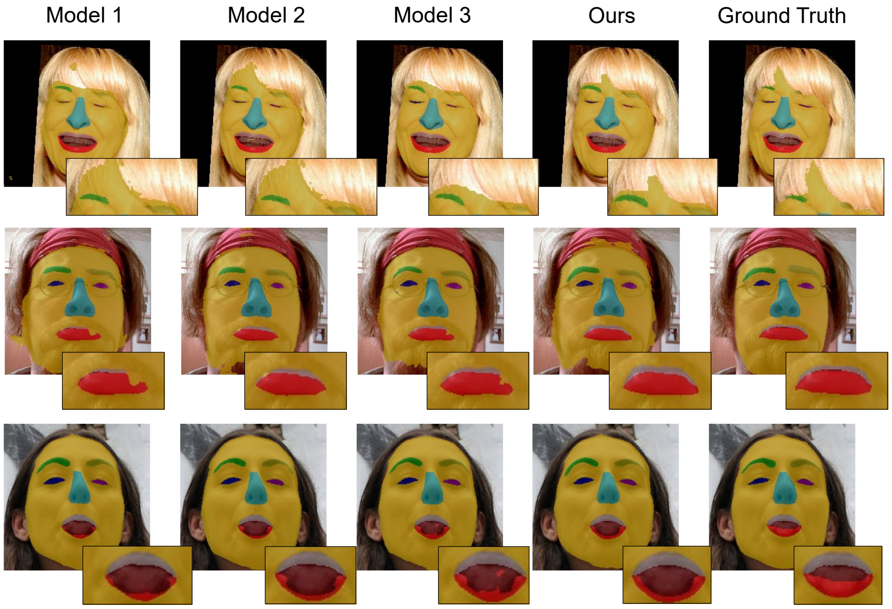

Fig. 3. Parsing results of different models on the Helen dataset. (Best viewed in color)

Table 2. Performance comparison with different deployment of the EAGR module and pooling size.

	Deployment				Pooling Size		
Model	0-module	1-module	2-modules	3-modules	4 × 4	6 × 6	8 × 8
mIoU	77.6	77.6	78.2	77.4	77.0	78.2	78.0
F1-score	92.0	92.5	92.8	92.3	92.1	92.8	92.6
Accuracy	85.5	86.0	87.3	85.4	87.4	87.3	87.0

edges in particular, such as the edge between the hair and skin in the first row, the upper lips in the second row, and edges around the mouse in the third row. In contrast, our complete model produce the best results with accurate edges between face constitutes, which is almost the same as the ground truth. This validates the effectiveness of the proposed edge aware graph reasoning.

On the deployment of the EAGR module We also conduct experiments on the deployment of the EAGR module with respect to the feature maps as well as pooling sizes. We take the output of Conv2 in the ResNet as the low-level feature map, and that of the pyramid pooling module as the high-level feature map. We compare four deployment schemes: 1) 0-module, where no EAGR module is applied; 2) 1-module, where the low-level and high-level feature maps are concatenated, and then fed into an EAGR module; 3) 2-modules, where the low-

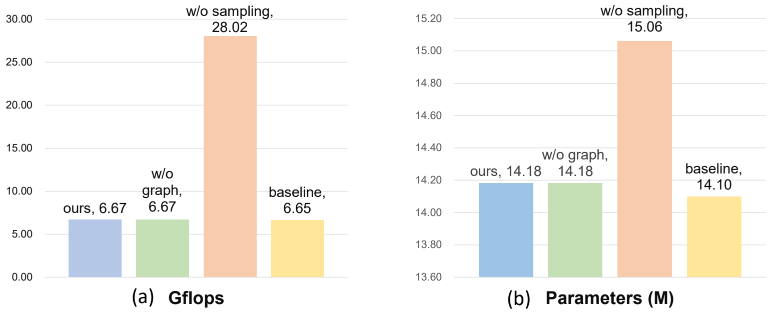

Fig. 4. Complexity comparison on the Helen dataset. We reset the start value of y-axis for better appearance.

level and high-level feature maps are fed into one EAGR module respectively; 4) 3-moduels, which combines 2) and 3). As listed in Table 2, the scheme of 2-modules leads to the best performance, which is the one we finally adopt.

We also test the influence of the pooling size, where the number of vertices changes along with the pooling size. As presented in Table 2, the size of 6 × 6 leads to the best performance, while enlarging the pooling size further does not bring performance improvement. This is because more detailed anchors lead to the loss of integrity, which breaks the holistic semantic representation.

On the complexity in time and space Further, we study the complexity of different models in time and space in Fig. 4. We compare with three schemes: 1) a simplified version without the EAGR module, which we refer to as the *Baseline*; 2) a non-local module [19] employed without edge aware sampling (*i.e.*, pooling) as *Without sampling*; and 3) a version without graph convolution for reasoning as *Without graph*. As presented in Fig. 4, compared with the typical non-local module, our proposed method reduces the computation time by more than 4× in terms of flops. We also see that the computation and space complexity of our method is comparable to those of the *Baseline*, which indicates that most complexity comes from the backbone network. Using Nvidia P40, the time cost of our model for a single image is 89ms in the inference stage. This demonstrates that the proposed EAGR module achieves significant performance improvement with trivial computational overhead.

4.4 Comparison with the state-of-the-art

We conduct experiments on the broadly acknowledged Helen dataset to demonstrate the superiority of the proposed model. To keep consistent with the previous works[6,4,38,5,33], we employ the overall F1 score to measure the performance, which is computed by combining the merged eyes, brows, nose and mouth categories. As Table 3 shows, Our model surpasses state-of-the-art methods and achieves 93.2% on this dataset.

Table 3. Comparison with state-of-the-art methods on the Helen dataset (in F1 score).

Methods	Skin	Nose	U-lip	I-mouth	L-lip	Eyes	Brows	Mouth	Overall
Liu *et al.* [6]	92.1	93.0	74.3	79.2	81.7	86.8	77.0	89.1	88.6
Lin *et al.* [4]	94.5	95.6	79.6	86.7	89.8	89.6	83.1	95.0	92.4
Wei *et al.* [38]	**95.6**	95.2	80.0	86.7	86.4	89.0	82.6	93.6	91.7
Yin *et al.* [5]	-	**96.3**	82.4	85.6	86.6	89.5	84.8	92.8	91.0
Liu *et al.* [33]	94.9	95.8	**83.7**	89.1	**91.4**	89.8	83.5	**96.1**	93.1
Ours	94.6	96.1	83.6	**89.8**	91.0	**90.2**	**84.9**	95.5	**93.2**

Table 4. Experimental comparison on the CelebAMask-HQ dataset (in F1 score).

Methods	Face	Nose	Glasses	L-Eye	R-Eye	L-Brow	R-Brow	L-Ear	R-Ear	Mean
	I-Mouth	U-Lip	L-Lip	Hair	Hat	Earring	Necklace	Neck	Cloth	
Zhao *et al.* [35]	94.8	90.3	75.8	79.9	80.1	77.3	78	75.6	73.1	76.2
	89.8	87.1	88.8	90.4	58.2	65.7	19.4	82.7	64.2	
Lee *et al.* [1]	95.5	85.6	**92.9**	84.3	85.2	81.4	81.2	84.9	83.1	80.3
	63.4	88.9	90.1	86.6	**91.3**	63.2	26.1	**92.8**	68.3	
Ours	**96.2**	**94**	92.3	**88.6**	**88.7**	**85.7**	**85.2**	**88**	**85.7**	**85.1**
	95	**88.9**	**91.2**	**94.9**	87.6	**68.3**	**27.6**	89.4	**85.3**	

We also evaluate our model on the newly proposed CelebAMask-HQ [1] and LaPa [33] datasets, whose scales are about 10 times larger than the Helen dataset. Different from the Helen dataset, CelebAMask-HQ and LaPa have accurate annotation for hair. Therefore, mean F1-score (over all foreground categories) is employed for better evaluation. Table 4 and Table 5 give the comparison results of the related works and our method on these two datasets, respectively.

4.5 Visualization of Graph Projection

Further, we visualize the graph projection for intuitive interpretation. As in Fig. 5, given each input image (first row), we visualize the weight of each pixel that contributes to a vertex marked in a blue rectangle in the other rows, which we refer to as the response map. Darker color indicates higher response. We observe that the response areas are consistent with the vertex, which validates that our graph projection maps pixels in the same semantic component to the same vertex.

5 Conclusion

We propose a novel graph representation learning paradigm of edge aware graph reasoning for face parsing, which captures region-wise relations to model long-range contextual information. Edge cues are exploited in order to project significant pixels onto graph vertices on a higher semantic level. We then learn the

Table 5. Experimental comparison on the LaPa dataset (in F1 score).

Methods	Skin	Hair	L-Eye	R-Eye	U-lip	I-mouth	L-lip	Nose	L-Brow	R-Brow	Mean
Zhao *et al.* [35]	93.5	94.1	86.3	86.0	83.6	86.9	84.7	94.8	86.8	86.9	88.4
Liu *et al.* [33]	97.2	**96.3**	88.1	88.0	84.4	87.6	85.7	95.5	**87.7**	**87.6**	89.8
Ours	**97.3**	96.2	**89.5**	**90.0**	**88.1**	**90.0**	**89.0**	**97.1**	86.5	87.0	**91.1**

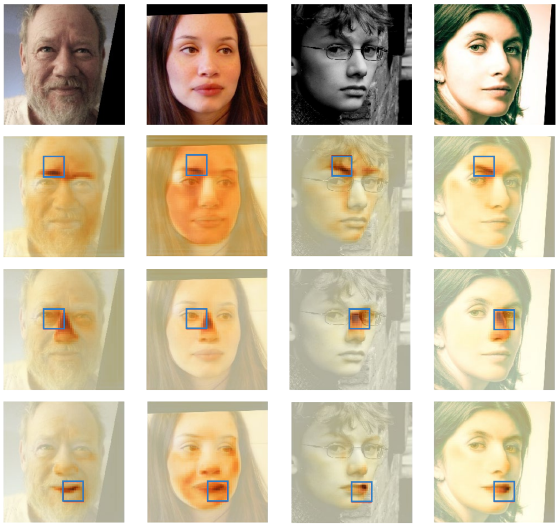

Fig. 5. Visualization of graph projection via response maps. The first row shows the input image, and the rest visualize response maps with respect to the vertex marked in a blue rectangle. Darker color indicates higher response.

relation between vertices (regions) and reason over all vertices to characterize the semantic information. Experimental results demonstrate that the proposed method sets the new state-of-the-art with low computation complexity, which efficiently reconstructs boundary details in particular. In future, we will apply the paradigm of edge aware graph reasoning to more segmentation applications, such as scene parsing.

Acknowledgement

This work was supported by National Natural Science Foundation of China [61972009], Beijing Natural Science Foundation [4194080] and Beijing Academy of Artificial Intelligence (BAAI).

References

1. Lee, C.H., Liu, Z., Wu, L., Luo, P.: Maskgan: Towards diverse and interactive facial image manipulation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 5549–5558
2. Zhang, H., Riggan, B.S., Hu, S., Short, N.J., Patel, V.M.: Synthesis of high-quality visible faces from polarimetric thermal faces using generative adversarial networks. International Journal of Computer Vision (2018) 1–18
3. Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters **23**(10) (2016) 1499–1503
4. Lin, J., Yang, H., Chen, D., Zeng, M., Wen, F., Yuan, L.: Face parsing with roi tanh-warping. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 5654–5663
5. Yin, Z., Yiu, V., Hu, X., Tang, L.: End-to-end face parsing via interlinked convolutional neural networks. arXiv preprint arXiv:2002.04831 (2020)
6. Liu, S., Shi, J., Liang, J., Yang, M.H.: Face parsing via recurrent propagation. In: 28th British Machine Vision Conference, BMVC 2017. (2017) 1–10
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE international conference on computer vision. (2016) 770–778
8. Henaff, M., Bruna, J., LeCun, Y.: Deep convolutional networks on graph-structured data. arXiv preprint arXiv:1506.05163 (2015)
9. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in Neural Information Processing Systems. (2016) 3844–3852
10. Smith, B.M., Zhang, L., Brandt, J., Lin, Z., Yang, J.: Exemplar-based face parsing. In: Proceedings of the IEEE international conference on computer vision. (2013) 3484–3491
11. Warrell, J., Prince, S.J.: Labelfaces: Parsing facial features by multiclass labeling with an epitome prior. In: IEEE international conference on image processing (ICIP). (2009) 2481–2484
12. Kae, A., Sohn, K., Lee, H., Learned-Miller, E.: Augmenting crfs with boltzmann machine shape priors for image labeling. In: Proceedings of the IEEE international conference on computer vision. (2013) 2019–2026
13. Liu, S., Yang, J., Huang, C., Yang, M.H.: Multi-objective convolutional learning for face labeling. In: Proceedings of the IEEE international conference on computer vision. (2015) 3451–3459
14. Luo, P., Wang, X., Tang, X.: Hierarchical face parsing via deep learning. In: Proceedings of the IEEE international conference on computer vision. (2012) 2480–2487
15. Zhou, E., Fan, H., Cao, Z., Jiang, Y., Yin, Q.: Extensive facial landmark localization with coarse-to-fine convolutional network cascade. In: Proceedings of the IEEE international conference on computer vision workshops. (2013) 386–391
16. Zhou, Y., Hu, X., Zhang, B.: Interlinked convolutional neural networks for face parsing. In: International symposium on neural networks, Springer (2015) 222–231
17. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
18. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. (2017) 5998–6008

19. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 7794–7803
20. Chen, Y., Kalantidis, Y., Li, J., Yan, S., Feng, J.: Aˆ 2-nets: Double attention networks. In: Advances in Neural Information Processing Systems. (2018) 352–361
21. Zhao, H., Zhang, Y., Liu, S., Shi, J., Change Loy, C., Lin, D., Jia, J.: Psanet: Point-wise spatial attention network for scene parsing. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 267–283
22. Zhu, Z., Xu, M., Bai, S., Huang, T., Bai, X.: Asymmetric non-local neural networks for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 593–602
23. Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H.: Dual attention network for scene segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 3146–3154
24. Chen, Y., Rohrbach, M., Yan, Z., Shuicheng, Y., Feng, J., Kalantidis, Y.: Graph-based global reasoning networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 433–442
25. Li, X., Zhong, Z., Wu, J., Yang, Y., Lin, Z., Liu, H.: Expectation-maximization attention networks for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 9167–9176
26. Chandra, S., Usunier, N., Kokkinos, I.: Dense and low-rank gaussian crfs using deep embeddings. In: Proceedings of the IEEE International Conference on Computer Vision. (2017) 5103–5112
27. Li, Y., Gupta, A.: Beyond grids: Learning graph representations for visual recognition. In: Advances in Neural Information Processing Systems. (2018) 9225–9235
28. Lu, Y., Chen, Y., Zhao, D., Chen, J.: Graph-fcn for image semantic segmentation. In: International Symposium on Neural Networks, Springer (2019) 97–105
29. Pourian, N., Karthikeyan, S., Manjunath, B.S.: Weakly supervised graph based semantic segmentation by learning communities of image-parts. In: Proceedings of the IEEE international conference on computer vision. (2015) 1359–1367
30. Te, G., Hu, W., Guo, Z.: Exploring hypergraph representation on face anti-spoofing beyond 2D attacks. In: 2020 IEEE International Conference on Multimedia and Expo (ICME), IEEE (2020) 1–6
31. Zhang, L., Li, X., Arnab, A., Yang, K., Tong, Y., Torr, P.H.: Dual graph convolutional network for semantic segmentation. arXiv preprint arXiv:1909.06121 (2019)
32. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, OpenReview.net (2017)
33. Liu, Y., Shi, H., Shen, H., Si, Y., Wang, X., Mei, T.: A new dataset and boundary-attention semantic segmentation for face parsing. In: AAAI. (2020) 11637–11644
34. Le, V., Brandt, J., Lin, Z., Bourdev, L., Huang, T.S.: Interactive facial feature localization. In: European conference on computer vision. (2012) 679–692
35. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of the IEEE international conference on computer vision. (2017) 2881–2890
36. Ruan, T., Liu, T., Huang, Z., Wei, Y., Wei, S., Zhao, Y.: Devil in the details: Towards accurate single and multiple human parsing. In: Proceedings of the AAAI Conference on Artificial Intelligence. Volume 33. (2019) 4814–4821

37. Rota Bulò, S., Porzi, L., Kontschieder, P.: In-place activated batchnorm for memory-optimized training of dnns. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018)
38. Wei, Z., Liu, S., Sun, Y., Ling, H.: Accurate facial image parsing at real-time speed. IEEE Transactions on Image Processing **28**(9) (2019) 4659–4670