Bidirectional Graph Reasoning Network for Panoptic Segmentation

Yangxin Wu[1†], Gengwei Zhang[1†], Yiming Gao[1], Xiajun Deng[1], Ke Gong[2], Xiaodan Liang[1,2⋆], and Liang Lin[1,2]

[1]Sun Yat-sen University, [2]DarkMatter AI Research

{*wuyx29, zhanggw8, gaoym9, dengxj9*}*@mail2.sysu.edu.cn, kegong936@gmail.com, xdliang328@gmail.com, linliang@ieee.org*

Abstract

Recent researches on panoptic segmentation resort to a single end-to-end network to combine the tasks of instance segmentation and semantic segmentation. However, prior models only unified the two related tasks at the architectural level via a multi-branch scheme or revealed the underlying correlation between them by unidirectional feature fusion, which disregards the explicit semantic and co-occurrence relations among objects and background. Inspired by the fact that context information is critical to recognize and localize the objects, and inclusive object details are significant to parse the background scene, we thus investigate on explicitly modeling the correlations between object and background to achieve a holistic understanding of an image in the panoptic segmentation task. We introduce a Bidirectional Graph Reasoning Network (BGRNet), which incorporates graph structure into the conventional panoptic segmentation network to mine the intra-modular and inter-modular relations within and between foreground things and background stuff classes. In particular, BGRNet first constructs image-specific graphs in both instance and semantic segmentation branches that enable flexible reasoning at the proposal level and class level, respectively. To establish the correlations between separate branches and fully leverage the complementary relations between things and stuff, we propose a Bidirectional Graph Connection Module to diffuse information across branches in a learnable fashion. Experimental results demonstrate the superiority of our BGRNet that achieves the new state-of-the-art performance on challenging COCO and ADE20K panoptic segmentation benchmarks.

1. Introduction

Thanks to the visual reasoning based on human commonsense, humans are capable of accomplishing recognition and segmentation of the objects and background of an image at a single glance. Recent researches have been devoted to developing numerous specific models for instance segmentation [6, 23] and semantic segmentation [27]. Generally, instance segmentation detects and segments each foreground object (named *things*) while semantic segmentation parses amorphous regions and background (named *stuff*). Tackling the two correlated tasks in separate models, these methods have sacrificed the holistic understanding of an image. Recently, a new proposed panoptic segmentation task has attracted researches [19, 20, 22, 26] to develop end-to-end networks to segment all foreground objects and background contents at the same time. As shown in Figure 1(a, b), some of the previous works [19, 20] unified instance segmentation and semantic segmentation at the architectural level via a multi-branch scheme. The others moved forward to reveal the underlying connection between the two related tasks by unidirectional feature fusion [22]. Although successfully tackling two tasks in one network, these approaches overlooked the explicit semantic and co-occurrence relations between objects and background in a complicated environment, which leads to limited performance gain.

To address these realistic challenges, we reconsider the characteristics of object segmentation as well as scene parsing and investigate on robustly modeling the various relations between them to better tackle the panoptic segmentation task. Intuitively, visual context is essential for instance segmentation when predicting fine-grained objects categories and contours [9], while foreground object details can benefit the segmentation of global scene and stuff [22]. It is obvious and remarkable that *things* and *stuff* can benefit each other by information propagation in one unified network to boost the overall performance of panoptic segmentation. Inspired by this, we introduce a new Bidirectional Graph Reasoning Network (named BGRNet) that incorporates graph structure into the conventional panoptic segmen-

† Equal contribution. ⋆ Corresponding Author.

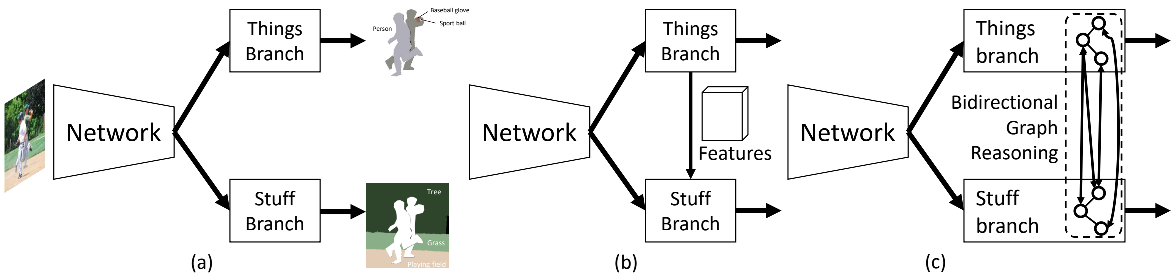

Figure 1. Different architectures for panoptic segmentation. (a) Simple multi-branch structure [19, 26] where two branches have no connection. (b) Unidirectional feature connection structure [22] that propagates information from *things* branch to *stuff* branch. (c) Our Bidirectional Graph Reasoning Network that enables mutual interaction and promotion for *things* and *stuff* based on graph convolution.

tation network to encode the semantic and co-occurrence relations as well as diffuse information between *things* and *stuff*, as shown in Figure 1(c).

Specifically, taking advantage of graph convolutional networks [18], our BGRNet extracts image-specific graphs from a panoptic segmentation pipeline and learns the diverse relations of *things* and *stuff* utilizing a multi-head attention mechanism. We propose a Bidirectional Graph Connection Module to bridge *things* graph and *stuff* graph in different branches, which enables graph reasoning and information propagation in a bidirectional way. Then we refine the feature representations in both branches by projecting the diffused graph node features. In this way, BGRNet is aware of the reciprocal relations between *things* and *stuff* and exhibits superior performance in panoptic segmentation.

Furthermore, our BGRNet can be easily instantiated to various network backbones and optimized in an end-to-end fashion. We perform extensive experiments on two challenging panoptic segmentation benchmarks, i.e., COCO [25] and ADE20K [38]. Our approach shows the superior flexibility and effectiveness in modeling and utilizing the relations between *things* and *stuff*, which achieves state-of-the-art performance in terms of PQ on two benchmarks.

2. Related Work

Instance Segmentation. Instance segmentation mainly focuses on locating and segmenting each foreground object. Early methods [7, 12] followed a bottom-up scheme [1] or top-down scheme based on segment proposals [13], until Mask R-CNN [14] extended Fast R-CNN to deal with instance segmentation by predicting instance masks and class labels in parallel, which became a common backbone for instance segmentation. Mask Scoring R-CNN [16] corrected Mask R-CNN by aligning mask quality with mask score.

Semantic Segmentation. Semantic segmentation parses scene images into per-pixel semantic classes. Began with FCNs [27] and DeepLab family [2], methods like fully convolutional network and atrous convolution made semantic segmentation thriving by boosting the overall segmentation quality. Besides, the scene parsing method with global context information was also studied in [36, 37].

Panoptic Segmentation. Panoptic Segmentation, a novel task introduced by [20], has lately received extensive attention by researchers. The task, which unifies instance segmentation and semantic segmentation, requires an algorithm that can segment foreground instances and background semantic classes simultaneously. In [20], Kirillov *et al.* simply combined the results from PSPNet and Mask R-CNN heuristically to produce panoptic segmentation outputs. Not long after, [19] proposed an end-to-end network for the panoptic task with a shared backbone and two branches: thing branch for instance segmentation and stuff branch for semantic segmentation, respectively. Instead of learning two tasks separately, [22] tried to utilize the features of the instance segmentation branch to boost the performance of the semantic segmentation branch through an attention mechanism. [26] proposed a spatial ranking module, to address the occlusion problem which hinders the performance of panoptic segmentation. Moreover, UPSNet [33] made use of deformable convolutions together with a parameter-free panoptic head in pursuit of more performance gain. A mini-deeplab module was also used to capture more contextual information in [29].

Graph Reasoning. There have been a surge of interest in graph-based methods [18, 30, 34, 35, 5] and graph reasoning has shown to have substantial practical merit for many tasks through modeling the domain knowledge in a single graph [4, 17, 32, 11] or directly fusing the graph reasoning results [10]. However, the mainstream approaches of panoptic segmentation are lack of the investigation on mining mutual relations from different domains (e.g. position and channel reasoning in network, *things* and *stuff* subsets) since different graph subsets need more explicit connections for mutual interaction and promotion. In this paper, we propose Bidirectional Graph Reasoning that propagates information from different graphs to support more flexible and

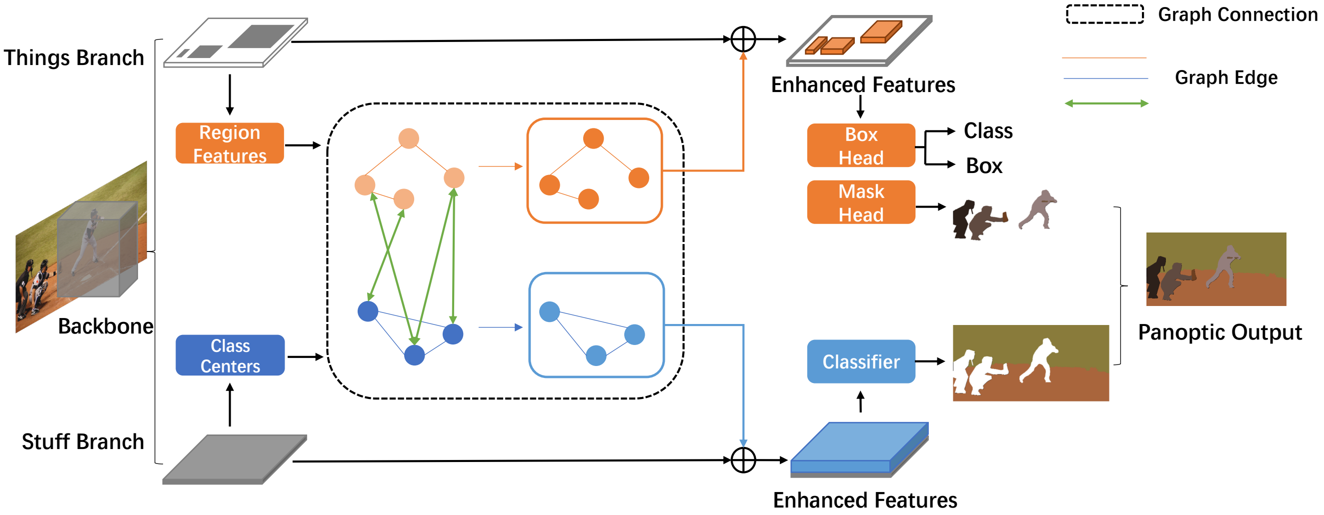

Figure 2. An overview of our BGRNet that can be stacked on any existing two branches panoptic segmentation network. The image features extracted by deep convolutional networks are fed into *things* branch and *stuff* branch. We construct Thing-Graph based on the region features after pooling. And we obtain Stuff-Graph node representations by extracting class centers from local feature. Then Bidirectional Graph Connection Module is used to propagate the high-level semantic graph representations within separate branches and across branches. Finally, we re-project the graph features to enhance the discriminability of visual features and improve the performance of both *things* and *stuff* branch.

complex reasoning tasks in general cases. Moreover, different from [4, 17, 32] that use a single graph for reasoning, our method aims to build a Graph Connection Module, whose nodes have strong semantics (rather than ambiguous nodes in [4]) and are hence more explainable and capable of encoding various relations.

3. Bidirectional Graph Reasoning Network

3.1. Overview

The panoptic segmentation task is to assign each pixel in an image a semantic label and an instance id. Current methods typically address this issue with a unified model using two branches for foreground *things* and background *stuff* separately [8, 19, 21, 22]. In detail, for an input image, the final panoptic segmentation result was generated by combining results from two branches using fusion strategy following [20]. Extending the simple but effective baseline in [19], we aim at further mining the intra-branch and inter-branch relations within and between foreground *things* or background *stuff*. Firstly, as shown in Figure 3, we build image-specific graphs in two separate branches in the network to enable flexible reasoning at the proposal level and class level. In the instance segmentation branch, a region graph is established to capture the pair-wise relationships among proposals. In the semantic segmentation branch, we build a graph based on the extracted class center that allows efficient global reasoning in a coarse-to-fine paradigm. Secondly, we propose a Bidirectional Graph Connection Module to deduce the implicit semantic relations between *things* and *stuff* in a learnable fashion. After diffusing information across various nodes, intra-modular reasoning is performed to refine the visual features of two branches. In this way, we explicitly model the correlations between *things* and *stuff* class and leverage their complementary relations in a global view, which facilitates panoptic segmentation and has substantial practical merit in our experiments. An overview of our Bidirectional Graph Reasoning Network is shown in Figure 2.

3.2. Graph Representation

Formally, we define a graph as $G = (V, A, X)$ where V is the set of nodes, A denotes the adjacency matrix and X is the feature matrix where each row corresponds to a node in V.

Building Thing-Graph. In the classic object detection paradigm, extracted regions are analyzed separately without considering the underlying dependencies between objects, which leads to inconsistent detection results and limited performance in more challenging tasks like panoptic segmentation. To remedy this issue, we introduce a Thing-Graph to reason directly beyond local regions, which can refine visual features of certain regions that suffer from occlusions, class ambiguities and tiny-size objects. Specifically, we build a Thing-Graph $G_{th} = (V_{th}, A_{th}, X_{th})$ on each input image, where $|V_{th}|$ equals to the number of detected regions in the image, $X_{th} \in \mathbb{R}^{|V_{th}| \times N}$ are extracted features from backbone of all regions and N is the dimension of the region feature. Considering the diverse relations among regions, we render the edges in G_{th} learnable to al-

这边的poposal都是固定的

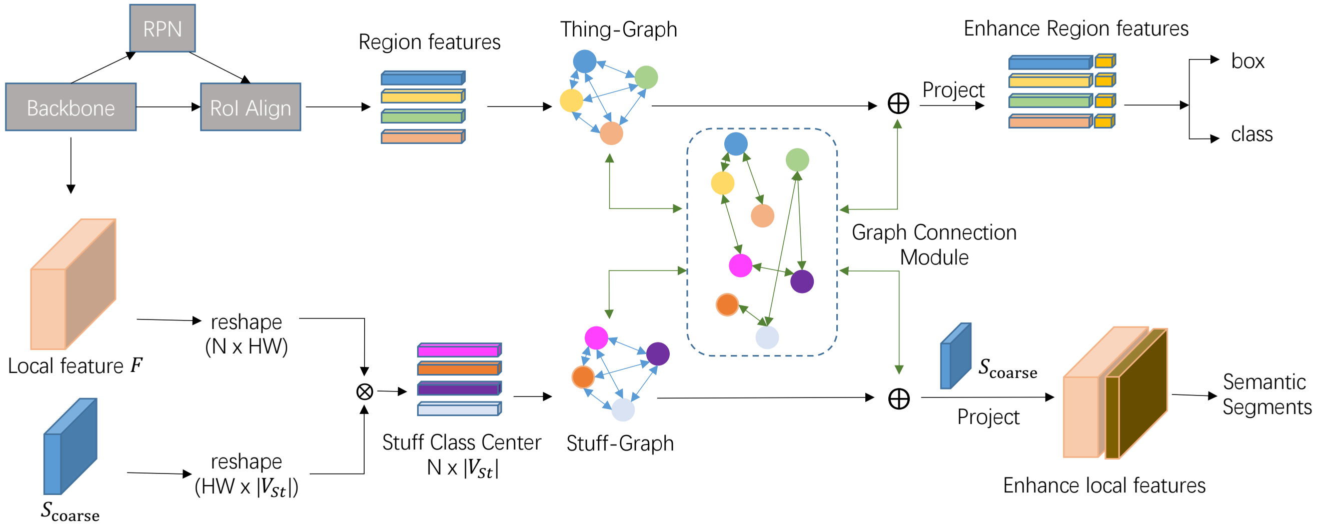

Figure 3. Diagram of our intra-modular graph, i.e., Thing-Graph and Stuff-Graph, and our inter-modular Graph Connection module. For Thing-Graph, we utilize the pooled region features as region graph nodes. For Stuff-Graph, we extract class center from local feature via coarse score map. Then Graph Connection Module diffuses information across various graph nodes and intra-modular graph reasoning is performed to project graph nodes features to visual features at the proposal and pixel level, respectively, in order to refine the results of instance segmentation and semantic segmentation, which are then heuristically combined in an NMS-like procedure following [20].

low flexible reasoning among multiple proposals. We also demonstrate the effectiveness of this learnable scheme by comparing the results of using different kinds of knowledge graphs in Section 4.3.

Building Stuff-Graph. As for semantic segmentation, a naive idea of building a Stuff-Graph can be considering each pixel as a graph node similar to the non-local network [31]. However, this approach exhibits clear limitations in dense predictions of semantic segmentation since it requires a large amount of computation and vast GPU memory occupation. Thus, to reduce the computation overhead as well as capture the long-range dependencies, we project the entire feature map to the vertices of Stuff-Graph so that every vertex represents a specific *stuff* class. Regarding Stuff-Graph $G_{st} = (V_{st}, A_{st}, X_{st})$, given the coarse score map $S_{coarse} \in \mathbb{R}^{|V_{st}| \times H \times W}$ produced by the original segmentation head in the baseline network, and segmentation feature map $F \in \mathbb{R}^{N \times H \times W}$, where N is the number of feature channels, we first reshape S_{coarse} to $\mathbb{R}^{HW \times |V_{st}|}$ and F to $\mathbb{R}^{N \times HW}$. After performing softmax along the HW channel on score map, we can obtain class nodes feature $X_{st} \in \mathbb{R}^{|V_{st}| \times N}$ by matrix multiplication and transposition:

$$X_{st} = (\bar{F}\bar{S}_{coarse})^T, \tag{1}$$

where $\bar{F}$ and $\bar{S}_{coarse}$ represent F and S_{coarse} after reshaping. The intuition behind Equation 1 is that local features, i.e., the features of pixels, are gathered to obtain class nodes feature based on pixel affinity via soft-mapping. By assigning global class nodes features to X_{st}, we significantly reduce computation overhead in building a Stuff-Graph since $HW \gg |V_{st}|$. Besides, the extracted *stuff* nodes are more representative and can provide global clues to further benefit the final classification process after remapping them to local features. We further demonstrate the representative characteristics of the extracted class centers in Stuff-Graph in Section 4.3. The processes of building Thing-Graph and Stuff-Graph are visualized in Figure 3.

3.3. Bidirectional Graph Connection Module

Given the Thing-Graph and Stuff-Graph, we aim to model the mutual relations between *things* and *stuff* and propagate the features across all nodes in both G_{th} and G_{st}. The rationale behind the design of graph nodes feature fusion module across branches is quite straightforward and comprehensible since there exists a consistent pattern of the co-occurrence of foreground *things* and background *stuff* in real-world scenarios. For example, when there exist objects like *persons*, *sports balls*, *baseball bats* and *baseball gloves* in an image, it is more reasonable to predict the stuff of *sand* and *playing field*, and vice versa. Therefore, we distill this insight into Graph Connection Module to bridge all semantic information across branches (between foreground *things* and background *stuff*). In this way, the information, relations or visual correlations of different categories from separate branches can be exploited.

The Graph Connection from Thing-Graph to Stuff-Graph can be formulated as:

$$X_{t-s} = A_{t-s} X_{th} W_{st}, \tag{2}$$

where $A_{t-s} \in \mathbb{R}^{|V_{st}| \times |V_{th}|}$ is a transfer matrix for propagating the information from Thing-Graph to Stuff-Graph, $W_{st} \in \mathbb{R}^{N \times D_0}$ is a trainable projection matrix. X_{t-s} is the mapped node features from Thing-Graph to Stuff-Graph. Similarly, the Graph Connection from Stuff-Graph to Thing-Graph can be obtained utilizing X_{st} and transfer matrix A_{s-t} with a trainable matrix W_{th}. Therefore, we seek for appropriate transfer matrix $A_{t-s} = \{a_{ij}^{t-s}\}$ and $A_{s-t} = \{a_{ij}^{s-t}\} \in \mathbb{R}^{|V_{th}| \times |V_{st}|}$, where a_{ij}^{s-t} denotes the connection weight from the j^{th} node of Stuff-Graph to the i^{th} node of Thing-Graph.

Based on the graph representation and Graph Connection, our graph structure can be naturally decomposed into blocks, given by

$$\hat{\mathbf{A}} = \begin{bmatrix} A_{th} & A_{s-t} \\ A_{t-s} & A_{st} \end{bmatrix}, \hat{\mathbf{X}} = \begin{bmatrix} X_{th} \\ X_{st} \end{bmatrix}, \tag{3}$$

where $A_{th}, A_{st}, A_{t-s}, A_{s-t}$ are normalized adjacency matrices for thing-to-thing pairs, stuff-to-stuff pairs, thing-to-stuff pairs, and stuff-to-thing pairs respectively. To model the distribution of different node features and adaptively handle their pairwise relations, we resort to attention mechanism [30] to obtain sufficient expressive power in our model. Formally, for any two nodes x_i, x_j in $\hat{\mathbf{X}}$, the edge weight α_{ij} is computed by:

$$\alpha_{ij} = \frac{\exp\left(\delta\left(W\left[x_i \| x_j\right]\right)\right)}{\sum_{k \in \mathcal{N}_i} \exp\left(\delta\left(W\left[x_i \| x_k\right]\right)\right)}, \tag{4}$$

where $||$ is the concatenation operation, $\mathcal{N}_i$ is the neighborhood of node i, δ is LeakyReLU nonlinear activation function, and W is weight matrix. For simplicity, we build a fully connected graph for $\hat{\mathbf{X}}$, i.e., $\mathcal{N}_i$ contains all nodes in $\hat{\mathbf{X}}$.

Updating node features. Formally, with normalized graph adjacency matrix $\hat{\mathbf{A}}$ and node features $\hat{\mathbf{X}}$, a single graph reasoning layer is given by

$$\widetilde{\mathbf{X}} = \begin{bmatrix} \widetilde{X}_{th} \\ \widetilde{X}_{st} \end{bmatrix} = \hat{\mathbf{X}} \oplus \sigma(\hat{\mathbf{A}}\hat{\mathbf{X}} \otimes \hat{\mathbf{W}}), \tag{5}$$

where

$$\hat{\mathbf{W}} = \begin{bmatrix} W_{th} \\ W_{st} \end{bmatrix}, \hat{\mathbf{X}} \otimes \hat{\mathbf{W}} = \begin{bmatrix} X_{th}W_{th} \\ X_{st}W_{st} \end{bmatrix}, \tag{6}$$

$W_{th}, W_{st} \in \mathbb{R}^{D_0 \times D_0}$ are trainable weight matrices, $\widetilde{X}_{th}$, $\widetilde{X}_{st}$ are node features of new Thing-Graph and Stuff-Graph respectively, $\oplus$ denotes concatenation, and σ is ReLU nonlinear function. Using T Graph Reasoning layers, the model will propagate and update the information among classes to build more discriminating representations.

3.4. Project Nodes Features to Visual Features

To refine the results of instance and semantic segmentation, we project graph nodes features to visual features at the proposal and pixel level, respectively. We illustrate this process in Figure 3.

Intra-modular reasoning for detection. When enhancing the features of *things* branch, we only care about the features in proposals. Hence we concatenate the updated Thing-Graph features to each proposal after adjusting their dimension:

$$f_{th} = A_{th}\tilde{X}_{th}W_{th}^{intra}, \tag{7}$$

where $W_{th}^{intra} \in \mathbb{R}^{(N+D_0) \times D_1}$ is the weight matrix for intra-modular reasoning in *things* branch. Then we concatenate enhanced features f_{th} to the visual features of proposals and feed them into the final fully connected layer to obtain the detection results.

Intra-modular reasoning for segmentation. To facilitate the dense prediction in the *stuff* branch, we need to enhance the local feature of each pixel under the guidance of extracted class centers. This can be regarded as the inverse operation of Equation 1. We reshape S_{coarse} to $\mathbb{R}^{HW \times |V_{st}|}$, the enhanced feature of *stuff* branch can be calculated as:

$$f_{st} = S_{coarse}\tilde{X}_{st}W_{st}^{intra}, \tag{8}$$

where $W_{st}^{intra} \in \mathbb{R}^{(N+D_0) \times D_2}$ is the weight matrix for intra-modular reasoning in *stuff* branch. Then f_{st} is concatenated with local feature F, which is then fed into the final convolution layer to obtain semantic segmentation results.

4. Experiments

4.1. Experimental Settings

Implementation Details. The architecture of BGRNet is built on Mask R-CNN [14] with a simple semantic segmentation branch similar to [33]. To be exact, the multi-level features from ResNet50-FPN [15, 24] first undergo deformable subnets with 3 convolution layers per level and are then bilinearly upsampled to 1/4 of the original scale of the input image. Finally, features from different levels are added together and 1 × 1 convolution with softmax is applied to predict all *stuff* classes. We follow all hyper-parameters settings and data augmentation strategies in Panoptic-FPN [19]. We implement our model using Pytorch [28] and train all models with 8 GPUs with a batch size of 16. The initial learning rate is 0.02 and is divided by 10 two times during fine-tuning. For COCO, we train for 12 epochs, i.e., 1x schedule, following [19]. For ADE20K, we train for 24 epochs and keep the learning rate schedule in proportion to COCO. We adopt an SGD optimizer with a momentum of 0.9 and a weight decay of 5e-4. We find it beneficial to extend the attention mechanism to multi-head

Table 1. Performance comparisons with the state-of-the-art on the COCO val set. † indicates our implementation. Panoptic-FPN-D is the deformable counterpart of Panoptic-FPN [19]. All methods use ResNet50-FPN as the backbone network.

Method	DF Conv.	PQ	PQTh	PQSt
Panoptic-FPN [19]		39.0	45.9	28.7
Panoptic-FPN-D†	✓	39.9	46.9	29.3
AUNet [22]		39.6	49.1	25.2
OANet [26]		39.0	48.3	26.6
UPSNet-C [33]	✓	41.5	47.5	32.6
UPSNet-CP [33]	✓	41.5	47.3	32.8
UPSNet [33]	✓	42.5	48.5	33.4
SpatialFlow [3]		40.9	46.8	31.9
Our BGRNet	✓	**43.2**	**49.8**	**33.4**

attention [30] and we applied 3 independent output attention heads. We use two Graph Reasoning layers (*i.e.* $T = 2$) and dimension $N = D_0 = D_1 = D_2 = 128$.

Datasets and Evaluation Metrics. We evaluate our method on COCO [25] and ADE20K [38]. COCO is one of the most challenging datasets for panoptic segmentation consisting of 115k images for training, 5k images for validation, and 20k images for *test-dev* with 80 *things* and 53 *stuff* classes. ADE20K is a densely annotated dataset for panoptic segmentation containing 20k images for training, 2k images for validation and 3k images for test, with 100 *things* and 50 *stuff* classes. Following [20], we adopt *panoptic quality (PQ)*, *semantic quality (SQ)*, and *recognition quality (RQ)* for evaluation.

Table 2. Performance comparisons on ADE20K val set. Panoptic-FPN-D is the deformable counterpart of Panoptic-FPN [19]. † indicates our implementation.

Methods	PQ	PQTh	PQSt
Panoptic-FPN† [19]	29.3	32.5	22.9
Panoptic-FPN-D† [19]	30.1	33.1	24.0
Our BGRNet	**31.8**	**34.1**	**27.3**

4.2. Comparisons with state-of-the-art

Comparisons with recent state-of-the-art methods on COCO and ADE20K dataset are listed in Table 1, 2. Some previous methods achieve high performance with over 42.5% PQ, thanks to the specially designed panoptic head [26], multi-scale information [19, 26], and two sources of attention [22]. Unlike previous methods [33, 26, 22], our BGRNet does not rely on complicated feature fusion process, i.e., RoI-Upsample [22], spatial ranking module [26], mask pruning process [33]. Instead, we utilize powerful graph models to capture intra-modular and inter-modular dependencies across separate branches. Thus, we achieve consistent accuracy gain over existed methods and set the new state-of-the-art results in terms of PQ, PQ^{Th}, PQ^{St}. The advanced results demonstrate the superiority of our BGRNet that incorporates the reciprocal information and deduces underlying relations between *things* and *stuff* appeared in the image.

The qualitative results on the ADE20K dataset are shown in Figure 5. As can be observed, our approach outputs more semantically meaningful and precise predictions than baseline methods despite the existence of complex object appearances and challenging background contents. For example, the baseline mistakes *field* for *grass* while our BGRNet predicts correctly thanks to the propagated information from the *things* in the image. More visual results on COCO and ADE20K can be found in Supplementary Materials.

Table 3. Ablation studies on ADE20K val set.

Methods	PQ	PQTh	PQSt
Baseline	30.1	33.3	23.7
w Thing-Graph	30.6	33.7	24.9
w Stuff-Graph	30.7	33.0	26.2
w Thing-Graph/Stuff-Graph	31.1	33.5	26.5
Our BGRNet	31.8	34.1	27.3

4.3. Ablation Study

Combinations of intra-modular and inter-modular graphs. Table 3 shows the performance of different components of our BGRNet on ADE20K val set. “w Thing(Stuff)-Graph” only has a single graph for foreground or background branch, while “w Thing-Graph/Stuff-Graph” contains graphs in both two branches with no inter-branch interaction, and the graph nodes are re-projected to visual features similar to Section 3.4.

We first analyze the effect of a single graph in either *things* branch or *stuff* branch. For single Thing-Graph, both PQTh and PQSt get improved thanks to the region-wise reasoning that considers the correlations among proposals. For single Stuff-Graph, PQSt got a 2.5% relative improvement, which showcases the great effect of extracting class centers to refine local features in a coarse-to-fine paradigm. Incorporating these two graphs with no connection across branches, the overall PQ is already 1% higher than the baseline, which is a considerable improvement on challenging ADE20K dataset. Furthermore, we introduce graph connection module, which greatly improves the segmentation quality of *things* and *stuff*, due to the ability to mine the underlying relations between foreground and background. As can be seen from the last row in Table 3, our BGRNet improves PQTh and PQSt by 0.8% and 3.6% respectively, resulting in 31.8% overall PQ, which outperforms Panoptic-FPN [19] by a large margin.

Thing/Stuff-Graph Construction. To validate the efficiency of the proposed Thing-Graph and Stuff-Graph, we

Table 4. Comparisons of different graphs and architectural designs on ADE20K val set.

#	Basic network [14]	Thing-Graph Construction		Stuff-Graph Construction		Graph Connection		Reasoning direction		PQ	PQ^{Th}	PQ^{St}
		Knowledge Graph [17]	Attention	Non-local [32]	Class-center	Semantic similarity	Attention	Thing-Stuff	Stuff-Thing			
1	✓									30.1	33.3	23.7
2	✓	✓								30.4	33.5	24.2
3	✓		✓							30.6	33.7	24.9
4	✓			✓						30.6	32.8	26.3
5	✓				✓					30.7	33.0	26.2
6	✓		✓		✓	✓		✓	✓	31.5	33.7	27.1
7	✓		✓		✓		✓	✓		31.4	33.6	27.0
8	✓		✓		✓		✓		✓	31.6	**34.3**	26.2
9	✓		✓		✓		✓	✓	✓	**31.8**	34.1	**27.3**

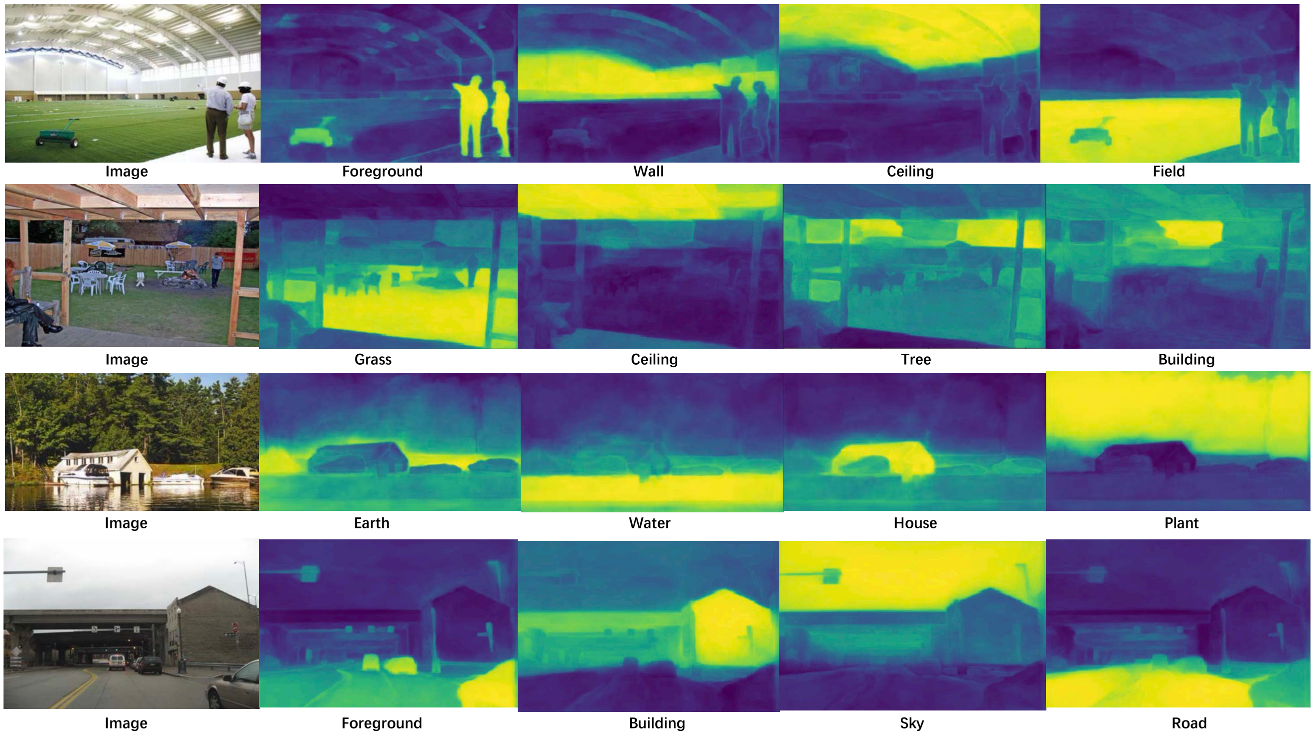

Figure 4. Visualization of similarities between extracted class centers and pixels generated by our method. Class Centers are listed below the images. The deeper the color is, the stronger the similarity between the class center and pixels. Benefited from the Class-center Stuff-Graph Construction scheme, our BGRNet can refine the local features under the guidance of the class center from a global view. Best viewed in color.

consider different construction methods and compare their performance in Table 4(#2,#3). Regarding Thing-Graph, we consider establishing the region-wise relations via a fixed knowledge graph. As for the knowledge graph for foreground objects, we follow [17] to construct a fixed relation knowledge Thing-Graph and extract an adjacency matrix of regions according to their class predictions. This scheme achieves 30.4% PQ, which is inferior to the adopted multi-head attention mechanism in BGRNet. The weakness may lie in the wrong region graphs due to the misclassification of some proposals, which indicates that the edge weights between some proposals are not reasonable anymore. As for the non-local graph for background, though slightly higher PQ^{St} (26.3% vs 26.2%) is achieved, it incurs much larger computation since every pixel is regarded as a graph node. Furthermore, with a non-local graph, the subsequent graph connection will be prohibitively expensive when the region-based Thing-Graph is considered. As can be seen, constructions of attention-based Thing-Graph and class-center Stuff-Graph lead to higher performance and moderate computation.

Different Graph Connection matrices. We also investigate the performance of our model using a different graph connection method, i.e., semantic similarity. To be exact, the $\tilde{\mathbf{A}}$ in Equation 3 is built on the semantic similarity other than a multi-head mechanism under this setting. The word embeddings of predicted classes of regions and stuff names of class centers are used to calculate the cosine similarity

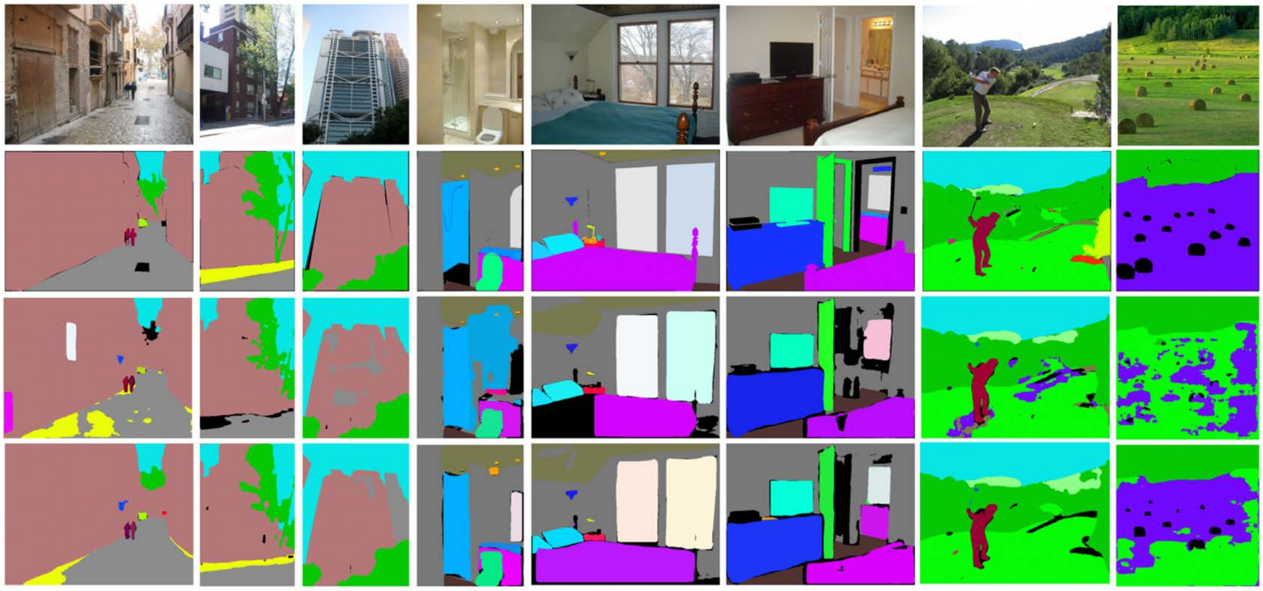

Figure 5. Visualized comparisons of panoptic segmentation outputs on ADE20K dataset. Raw images, Ground-Truth Segmentation, Panoptic-FPN outputs and BGRNet outputs are presented from top to bottom.

to form an adjacency matrix. As can be seen in Table 4, the semantic similarity-based connection is also helpful in bridging the chasm between *things* and *stuff* and achieves 31.5% PQ, which is still lower than that of attention-based mechanism (31.8% PQ). This indicates that our Graph Connection Module is supposed to obtain more sufficient expressive power and discover the diverse relations between *things* nodes and *stuff* nodes in a complicated scene than merely depends on a fixed linguistic graph.

Unidirectional enhancement. We investigate the direction of Graph Connection by exploring unidirectional enhancement in Table 4. Previous method [22] uses two sources of attention to perform unidirectional enhancement from the foreground branch to background branch. To fully leverage the reciprocal relations between foreground and background, we thus investigate and compare the performance with different enhance directions. ‘Thing-Stuff’ stands for only enhancing the feature of semantic segmentation branch after Graph Connection. ‘Stuff-Thing’ represents only enhancing the feature of detection branch after Graph Connection. It can be found that although unidirectional enhancement can lead to considerable performance gain, merely performing Graph Connection in one direction is not able to fully enhance the feature, and a two-way graph connection further boosts the overall PQ to 31.8%.

Visualize the correlations. To demonstrate the representative characteristics of the extracted class centers described in Section 3.2, we visualize the similarity between particular *stuff* class centers and local features of pixels in Figure 4. As can be seen, the extracted *stuff* class center correlates well with corresponding area and the responses in other area are inhibited, despite the existence of multiple *stuff* classes, class ambiguity and fuzzy edges between different *stuff* classes. For example, in the third row, the extracted class centers correlate well with the confusing *stuff* class including *plant*, *water* and *earth*. Under the guidance of the class center features from a global view, local features can be refined. This greatly improves the performance of our model in terms of PQ^{St}.

5. Conclusion

This paper introduces a Bidirectional Graph Reasoning Network (BGRNet) for panoptic segmentation that simultaneously segments foreground objects at the instance level and parses background contents at the class level. We propose a Bidirectional Graph Connection Module to propagate the information encoded from the semantic and co-occurrence relations between *things* and *stuff*, guided by the appearances of the objects and the extracted class centers in an image. Extensive experiments demonstrate the superiority of our BGRNet, which achieves the new state-of-the-art performance on two large-scale benchmarks.

6. Acknowledgement

This work was supported in part by National Key R&D Program of China under Grant No. 2018AAA0100300, National Natural Science Foundation of China (NSFC) under Grant No.U19A2073 and No.61976233, Nature Science Foundation of Shenzhen Under Grant No. 2019191361.

References

[1] Pablo Arbeláez, Jordi Pont-Tuset, Jonathan T Barron, Ferran Marques, and Jitendra Malik. Multiscale combinatorial grouping. In *CVPR*, pages 328–335, 2014. 2

[2] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, pages 801–818, 2018. 2

[3] Qiang Chen, Anda Cheng, Xiangyu He, Peisong Wang, and Jian Cheng. Spatialflow: Bridging all tasks for panoptic segmentation. *arXiv preprint arXiv:1910.08787*, 2019. 6

[4] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Yan Shuicheng, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 433–442, 2019. 2, 3

[5] Ziliang Chen and Zhanfu Yang. Graph neural reasoning may fail in proving boolean unsatisfiability. *arXiv preprint arXiv:1909.11588*, 2019. 2

[6] Jifeng Dai, Kaiming He, Yi Li, Shaoqing Ren, and Jian Sun. Instance-sensitive fully convolutional networks. In *ECCV*, pages 534–549. Springer, 2016. 1

[7] Jifeng Dai, Kaiming He, and Jian Sun. Convolutional feature masking for joint object and stuff segmentation. In *CVPR*, pages 3992–4000, 2015. 2

[8] Daan de Geus, Panagiotis Meletis, and Gijs Dubbelman. Panoptic segmentation with a joint semantic and instance segmentation network. *arXiv preprint arXiv:1809.02110*, 2018. 3

[9] Nikita Dvornik, Julien Mairal, and Cordelia Schmid. Modeling visual context is key to augmenting object detection datasets. In *ECCV*, pages 364–380, 2018. 1

[10] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, pages 3146–3154, 2019. 2

[11] Ke Gong, Yiming Gao, Xiaodan Liang, Xiaohui Shen, Meng Wang, and Liang Lin. Graphonomy: Universal human parsing via graph transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7450–7459, 2019. 2

[12] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *ECCV*, pages 297–312. Springer, 2014. 2

[13] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, pages 447–456, 2015. 2

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 2, 5, 7

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 5

[16] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring r-cnn. In *CVPR*, pages 6409–6418, 2019. 2

[17] Chenhan Jiang, Hang Xu, Xiaodan Liang, and Liang Lin. Hybrid knowledge routed modules for large-scale object detection. In *NeurIPS*, pages 1552–1563, 2018. 2, 3, 7

[18] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 2

[19] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. *arXiv preprint arXiv:1901.02446*, 2019. 1, 2, 3, 5, 6

[20] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. *arXiv preprint arXiv:1801.00868*, 2018. 1, 2, 3, 4, 6

[21] Jie Li, Allan Raventos, Arjun Bhargava, Takaaki Tagawa, and Adrien Gaidon. Learning to fuse things and stuff. *arXiv preprint arXiv:1812.01192*, 2018. 3

[22] Yanwei Li, Xinze Chen, Zheng Zhu, Lingxi Xie, Guan Huang, Dalong Du, and Xingang Wang. Attention-guided unified network for panoptic segmentation. In *CVPR*, June 2019. 1, 2, 3, 6, 8

[23] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In *CVPR*, pages 2359–2367, 2017. 1

[24] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017. 5

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 2, 6

[26] Huanyu Liu, Chao Peng, Changqian Yu, Jingbo Wang, Xu Liu, Gang Yu, and Wei Jiang. An end-to-end network for panoptic segmentation. In *CVPR*, June 2019. 1, 2, 6

[27] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 1, 2

[28] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 5

[29] Lorenzo Porzi, Samuel Rota Bulo, Aleksander Colovic, and Peter Kontschieder. Seamless scene segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8277–8286, 2019. 2

[30] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. 2, 5

[31] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. 4

[32] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, pages 7794–7803, 2018. 2, 3, 7

[33] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: A unified panoptic segmentation network. *arXiv preprint arXiv:1901.03784*, 2019. 2, 5, 6

[34] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018. 2

[35] Zhanfu Yang, Fei Wang, Ziliang Chen, Guannan Wei, and Tiark Rompf. Graph neural reasoning for 2-quantified boolean formula solvers. *arXiv preprint arXiv:1904.12084*, 2019. 2

[36] Yuhui Yuan and Jingdong Wang. Ocnet: Object context network for scene parsing. *arXiv preprint arXiv:1809.00916*, 2018. 2

[37] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, pages 2881–2890, 2017. 2

[38] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017. 2, 6