Dual Graph-Based Context Aggregation for Scene Parsing

Mengyu Liu
mengyu.liu@manchester.ac.uk
Hujun Yin
hujun.yin@manchester.ac.uk

Department of Electrical and Electronic Engineering
The University of Manchester
Manchester, UK

Abstract

Exploiting global contextual information has been shown useful for improving performance of scene parsing and hence is widely used. In this paper, unlike previous work that captures long-range dependencies with multi-scale feature fusion or attention mechanism, we address the scene parsing tasks by aggregating rich contextual information based on graph reasoning. Specifically, we propose two graph reasoning modules, in which features are aggregated over the coordinate space and projected to the feature and probabilistic spaces, respectively. The feature graph reasoning module adaptively constructs pyramid graphs as multi-scale feature representations and then performs graph reasoning to model global context. Whilst, in the probabilistic graph reasoning module, graph reasoning is performed over a graph consisting of class-dependent representations generated by aggregating the pixels that belong to the same classes. We have conducted extensive experiments on the popular scene parsing datasets, including Cityscapes, PASCAL Context and ADE20K, and achieved state-of-the-art performances.

1 Introduction

Scene parsing, a fundamental topic in computer vision, aims to predict classes of all pixels based on their properties for a given image. It has found various applications in auto-driving, indoor scene understanding and robot navigation. In recent years, convolutional neural networks (CNNs) based on the fully convolutional network (FCN) [21] have pushed the performance of scene parsing algorithms to soaring heights [4, 19, 33, 38]. Scene parsing can be considered as a pixel-wise classification process, and capturing long-range relations between pixels and modelling global contextual information can help achieve good results. However, due to the local connectivity of CNN filters and large input images, size of receptive field is often too small to aggregate enough information even with very deep models [41].

To address the above problem, many approaches have been proposed to enlarge the size of receptive field or to aggregate and model global context. Early work [2, 3, 34] removed the last two downsampling layers and utilized dilated convolutions in CNNs to obtain larger feature maps and richer contextual information. Later, pyramid pooling based approaches [3, 4, 38] have been proposed to further enlarge the receptive field and capture global contextual information to boost the performance. Recent work [8, 13, 14, 35, 40, 43] based on the

self-attention mechanism [31] has been developed to capture global feature dependencies and update the representations for each pixel. More recently, graph reasoning based context modelling methods [5, 11, 17, 18, 32, 37] have shown excellent results on scene parsing tasks, where a graph is learned from clusters of pixels to generate feature representations and graph reasoning is performed over the graph to model global relations. In this paper, we address scene parsing by further exploring the properties of the optimal graphs constructed in the graph-based methods.

In principle, vertices in the optimal graphs should contain semantic contextual information, representing groups of pixels that share similar characteristics in the coordinate space, whilst irrelevant information should be suppressed as much as possible. Objects in images have varied sizes and locations, hence it is crucial to construct multi-scale graphs consisting of multi-level feature representations to capture multi-level context. Aggregating pixels belonging to the same classes is an effective way to obtain representative vertices in graph. To this end, we propose two types of graph reasoning modules to capture long-range dependencies and aggregate contextual information for scene parsing in the feature space and probabilistic space, respectively. Specifically, we append two parallel graph reasoning modules on the top of dilated CNNs. One is a feature graph reasoning (FGR) module, where pyramid graphs are constructed to generate multi-scale feature representations. We introduce graph reasoning to capture the dependencies between vertices in the graphs, and then graphs are fused and distributed back to coordinate space to enhance feature learning. The other is a probabilistic graph reasoning (PGR) module, which utilizes a coarse segmentation map to aggregate pixels and generate class-dependent representations in one graph. Then graph reasoning is performed to model the relations between different classes. Finally, the outputs of these two modules are fused to yield the final segmentation result.

Main contributions are summarized as follows:

(i) We explore the desired properties of graphs in the graph-based method, and demonstrate that exploiting multi-scale feature representations in pyramid graphs is an efficient way for feature learning, and utilizing prior segmentation information to construct graph can help model context and improve performance.

(ii) Two graph-based context aggregation modules are proposed to capture long-range dependencies and model contextual information in feature and probabilistic spaces respectively for scene parsing tasks.

(iii) We developed a model based on the proposed dual modules, and achieved state-of-the-art performances on the Cityscapes [6], the PASCAL Context [23] and the ADE20K [42] datasets, demonstrating the effectiveness of the proposed method.

2 Related Work

Scene Parsing and Semantic Segmentation. Recent approaches based on CNNs have achieved great successes in scene parsing and semantic segmentation tasks. FCN [21] was the first approach to replace the fully connected layers in CNNs with convolutional layers to convert the scene parsing tasks into pixel-level classification tasks. Since then, methods for scene parsing can be roughly divided into two categories. One category removes the last two downsampling operations and employs dilated convolutions to preserve receptive field and resolution [2, 3, 34, 38]. The other category adopts the encode-decoder structure to recover resolutions step by step [1, 24, 28]. Moreover, some segmentation methods focus on improving efficiency. In ICNet [39] and ContextNet [27], downsampled images were applied

to deep branches while large images were applied to shallow branches to reduce computational complexity. Raszke *et al*. [25] built a lightweight backbone to reduce computational complexity by discarding the last stage of the network. Mehta *et al*. proposed the ESPNet [22], where multiple dilated convolutions were adopted in each module to extract features.

Context Aggregation. Due to varied scales of objects in scene images and local connectivity of CNN filters, aggregating contextual information and capturing long-range dependencies can help boost the performance of scene parsing. In DeepLab series [2, 3, 4], an atrous spatial pyramid pooling (ASPP) module was proposed to encode and aggregate multi-scale contextual features by using different dilated convolutions. In ParseNet [20], global pooling was used to aggregate contextual information to provide sufficient global information. In PSPNet [38], different average pooling operations were utilized to obtain contextual information at different scales. EncNet [36] uses a context encoding module to selectively highlight class-dependent features. Another popular way for context aggregation is adopting the self-attention mechanism based on the non-local block [31] to model relations between each pair of pixels. In [8], two attention modules were proposed to capture global dependencies along spatial and channel dimensions respectively. In OCR [35], object contextual representations were learned by aggregating pixels lying in the object regions. Moreover, some methods [13, 14, 43] have been proposed to reduce the computational complexity of attention mechanism.

Graph Reasoning. Graph-based methods have been increasingly used in computer vision recently. Graph convolutions [15] were initially proposed for semi-supervised classification. Later, Chen *et al*. [5] projected pixels to an interactive space to obtain a feature graph, and then relational reasoning was performed by graph convolutions to model global context. SGR [18] uses external human knowledge to guide the graph reasoning module to enhance local feature representations. In A^2-FPN [12], multi-level features were extracted and projected to different graphs to capture dependencies for instance segmentation. GINet [32] creates two interactive graphs to encode dependencies between visual features and linguistic correlations respectively. In CDGCNet [11], a coarse prediction map is used to extract features and construct a separate graph for each class, and graph reasoning is independently performed to learn useful information. Different from this approach, our method constructs a single graph where each vertex represents a class and models the relations between different class-specific representations. In [17], the original feature maps were considered as pyramid graphs to be performed graph reasoning, while in our method, pixels in the feature maps are first aggregated and projected to pyramid graphs in different spaces, and each graph contains more essential feature representations than the original feature maps.

3 Methods

In this section, we first review the basic knowledge of graph convolution, then introduce in detail the proposed graph reasoning modules performing in the feature and probabilistic spaces respectively. Finally we describe how to combine them in the overall framework.

3.1 Graph Convolution

Traditional convolutions operate as sliding windows on input feature maps to encode feature pixels in neighbouring cells, pixels in neighbouring cells are connected in order with this operation. While in graph convolutions [15, 16], the input is an undirected graph, where

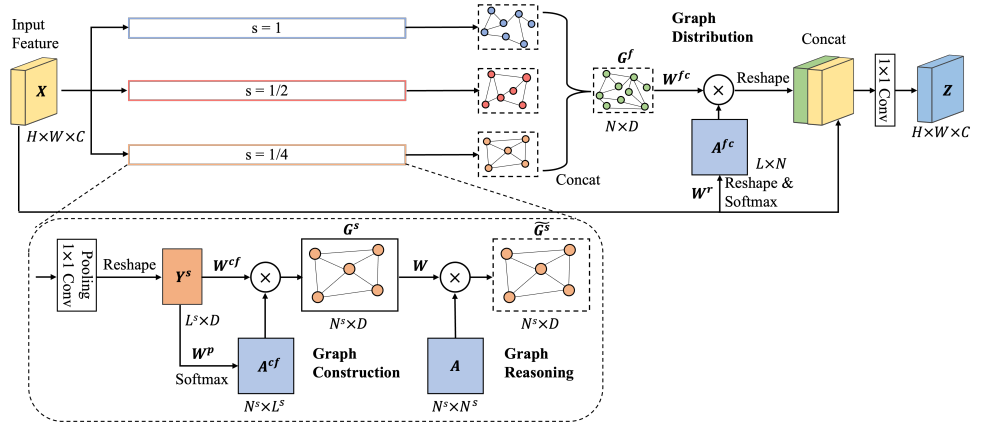


Figure 1: An overview of the proposed feature graph reasoning module. "s" denotes the scale factor of pooling operation, and $\otimes$ is matrix multiplication.

vertices are unordered and cannot be convolved by a structured convolutional filter. To propagate information over the entire graph, an adjacency matrix is used to model the relations between pairs of vertices. Explicitly, given an input graph $G \in \mathbb{R}^{N\times D}$, where N and D are the number of vertices and channel number of the graph respectively, the graph convolution is formulated as

$$\widetilde{G} = \sigma((A+I)GW), \tag{1}$$

where A is a $N \times N$ adjacency matrix for information diffusion over the graph, I is the identity matrix to add self-connections for the adjacency matrix, $W \in \mathbb{R}^{D\times D}$ is a trainable weight matrix to perform linear transformation and σ is a non-linear function. In our experiments, A is randomly initialized and learnable, W is set to a 1×1 2D convolution, and σ is set to ReLU function. After graph convolution, each vertex is incorporated with necessary contextual information from other vertices.

3.2 Dual Graph Reasoning Modules

3.2.1 Overview

In scene parsing tasks, the relationships between different objects provide vital clues. For example, *boats* are usually on the *water*, and *cars* are often by the side of the *road*. Dual to the limited receptive field and local connectivity of traditional convolutional filters, such global scene clues are hard to utilize. Capturing long-range dependencies and modelling global context can effectively address this issue.

Different from the previous pyramid pooling approaches or methods based on the self-attention mechanism, we propose two graph-based modules to capture global dependencies by modelling relations among vertices, and we also investigate the desired properties of optimal graphs. We design a feature graph reasoning (FGR) module, which constructs pyramid graphs to model multi-level feature representations, and a probabilistic graph reasoning (PGR) module to aggregate class-level contextual information and model class-dependent representations.

3.2.2 Feature Graph Reasoning (FGR) Module

Due to different scales and locations of objects in scene parsing tasks, multi-level features need be encoded to provide sufficient information. Towards this objective, we construct M feature graphs in the FGR module to encode multi-scale feature representations. Here we just take one scale s as an example, other scales are processed in a similar way. For this scale, given an input feature, $X \in \mathbb{R}^{H\times W\times C}$, where H, W and C are the height, width and channel number, respectively, we first use an average pooling operation to reduce spatial size and a 1×1 convolutional layer to squeeze channels, obtaining feature, $Y^s \in \mathbb{R}^{H^s\times W^s\times D}$, where H^s and W^s are the height and width at scale s respectively, and D is the reduced channel number.

As illustrated in Figure 1, for each scale, feature Y^s is first reshaped to $\mathbb{R}^{L^s\times D}$, where $L^s = H^s \times W^s$ is the number of pixels. Then a projection matrix, $A^{cf} \in \mathbb{R}^{N^s\times L^s}$, is calculated to transform Y^s from the coordinate space to a graph in the feature space, and N^s is the number of vertices. The projection matrix A^{cf} is generated by a linear transformation and softmax normalization as

$$A^{cf}_{ij} = \frac{\exp(\mathbf{y}^s_j \cdot \mathbf{w}^{p\top}_i)}{\sum_{t=1}^{L_s}\exp(\mathbf{y}^s_t \cdot \mathbf{w}^{p\top}_i)}, \tag{2}$$

where A^{cf}_{ij} is the normalized projection weight of assigning pixel $\mathbf{y}^s_j \in \mathbb{R}^{1\times D}$ to vertex i, and $W^p = [\mathbf{w}^p_1, \cdots, \mathbf{w}^p_{N^s}] \in \mathbb{R}^{N^s\times D}$ is a trainable weight to calculate the projection matrix. Then, following the previous work [5, 18], we use the projection matrix A^{cf} to construct the feature graph, $G^s \in \mathbb{R}^{N^s\times D}$, in the feature space as

$$G^s = A^{cf} Y^s W^{cf}, \tag{3}$$

where $W^{cf} \in \mathbb{R}^{D\times D}$ denotes a trainable linear transformation. In this way, pixels in the original feature Y^s are adaptively aggregated to feature representations in the feature graph. Note that the number of vertices N^s is fewer than the number of pixels L^s in the original feature maps to suppress irrelevant information, and N^s decreases with the decrease of L^s to obtain multi-level feature representations. Then graph reasoning is performed over the generated graph using Eqn. 1 to generate $\widetilde{G^s}$, where each vertex is updated with global information. In the experiments, we set weights W^{cf} and W^p to two 1×1 2D convolutions.

After generating graph at each scale, we concatenate the obtained M pyramid graphs, $\left\{\widetilde{G^s}\right\}$, along vertex dimension into the final feature graph, $G^f = \left[\widetilde{G^1}, \ldots, \widetilde{G^M}\right] \in \mathbb{R}^{N\times D}$, where $N = N^1 + \cdots + N^M$ is the total number of vertices. Next, the fused multi-scale feature representations in G^f can be distributed back to coordinate space to enhance the local information of each pixel in original feature. Exactly, we adopt a reverse projection to map the obtained feature graph, G^f, from the feature space to the coordinate space and reshape the result to $H \in \mathbb{R}^{H\times W\times C}$ and then fuse with input feature X as

$$\begin{aligned} H &= A^{fc} G^f W^{fc}, \\ Z &= \mathrm{conv}(\mathrm{concat}(X, H)), \end{aligned} \tag{4}$$

where $A^{fc} \in \mathbb{R}^{L\times N}$ is a reverse projection matrix used to adaptively transform feature representations from the feature space to the coordinate space, $W^{fc} \in \mathbb{R}^{D\times C}$ is a trainable weight to recover channel number from D to C, and a 1×1 convolutional layer is adopted to fuse the concatenated feature. The reverse projection matrix A^{fc} is computed by a linear combination

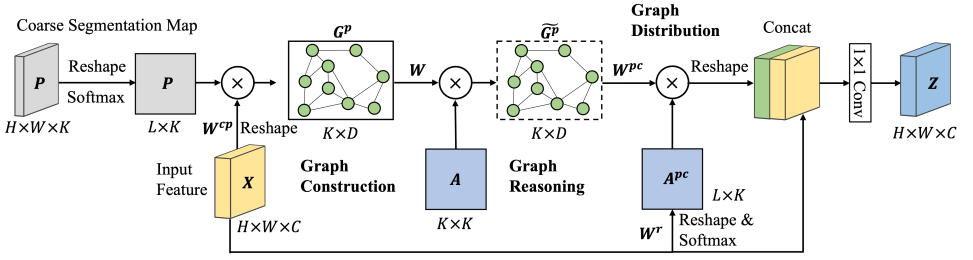


Figure 2: Architecture of the proposed probabilistic graph reasoning module.

and softmax normalization of X as

$$A_{ij}^{fc} = \frac{\exp(\mathbf{x}_i \cdot \mathbf{w}_j^{r\top})}{\sum_{l=1}^{N} \exp(\mathbf{x}_j \cdot \mathbf{w}_l^{r\top})}, \tag{5}$$

where A_{ij}^{fc} is the normalized reverse projection weight of assigning vertex j to feature $\mathbf{h}_i \in \mathbb{R}^{1\times C}$, and $W^r = [\mathbf{w}_1^r, \cdots, \mathbf{w}_N^r] \in \mathbb{R}^{N\times C}$ is a trainable weight to calculate the reverse projection matrix. In this FGR module, multi-level contextual information is aggregated and modelled in the pyramid feature graphs to help feature learning and boost performance.

3.2.3 Probabilistic Graph Reasoning (PGR) Module

In scene parsing tasks, input images usually contain multiple kinds of objects to be segmented, exploiting and modelling the relationships between class-dependent representations in graph can enhance global scene understanding.

In the PGR module, a coarse segmentation prediction map, $P \in \mathbb{R}^{H\times W\times K}$, where K is the number of classes, is used to map the input feature, $X \in \mathbb{R}^{H\times W\times C}$, from the coordinate space to the probabilistic space. As illustrated in Figure 2, we first normalize the prediction map along spatial dimension to make each value reflects the confidence of belonging to each class, and reshape it to $\mathbb{R}^{L\times K}$. Then we construct the probabilistic graph, $G^p \in \mathbb{R}^{K\times D}$, by

$$G^p = P^\top X W^{cp}, \tag{6}$$

where $W^{cp} \in \mathbb{R}^{C\times D}$ is a trainable weight to reduce channel number from C to D. In this way, pixels in original feature maps are aggregated to K class-dependent representations. Next we perform graph reasoning over the obtained graph using Eqn. 1 and generate $\widetilde{G^p}$, where the dependencies between classes are modelled. After getting the reasoned graph, we adopt a reverse projection, similar to the reverse projection in FGR module, to transform the graph back to coordinate space. We first calculate a reverse projection matrix $A^{pc} \in \mathbb{R}^{L\times K}$ following the same procedure of Eqn. 5, then we project $\widetilde{G^p}$ to H in the coordinate space and reshape H to $\mathbb{R}^{H\times W\times C}$. Finally, we obtain the final output feature, $Z \in \mathbb{R}^{H\times W\times C}$, as

$$\begin{aligned} H &= A^{pc}\widetilde{G^p}W^{pc}, \\ Z &= \mathrm{conv}(\mathrm{concat}(X, H)), \end{aligned} \tag{7}$$

where $W^{pc} \in \mathbb{R}^{D\times C}$ is a trainable weight to recover channel number from D to C. In this PGR module, class-dependent context is aggregated and modelled to emphasize object contextual information in feature maps with the guidance of prior segmentation map.

3.3 Overall Framework

We adopt the ResNets [10] pretrained on the ImageNet as our backbone. Following the previous work [2, 4, 38], we remove the last two downsampling operations in the backbone and employ dilated convolutions. The proposed dual graph reasoning modules are injected in parallel after the backbone. The output feature of res-5 stage is fed to these two modules as input. In the FGR module, three parallel branches are used to construct multi-scale feature graphs. Spatial sizes of input feature are reduced to 1, 1/2 and 1/4 of the original scale respectively, and the number of vertices in three graphs are set to 128, 64 and 32 respectively. While in the PGR module, the output feature of res-4 stage in the backbone is applied with a classifier to obtain the coarse segmentation map. Finally we aggregate the output features of the two modules to obtain the final results. We use the cross entropy loss to supervise the coarse and final segmentation, and the weight for auxiliary loss is set to 0.4 following the previous methods [36, 38].

4 Experiments

Benchmark datasets employed are described first, along with implementation details. Then the ablation studies performed are reported to show the effectiveness of the proposed method. Finally we report the evaluation results on the Cityscapes [6], PASCAL Context [23] and ADE20K [42] datasets.

4.1 Datasets

Cityscapes. The dataset, collected for urban scene understanding, contains 5000 images with 19 classes being annotated for scene parsing. All images are of size 2048×1024, and in our experiments only the fine annotated images were used for training and evaluation. The training, validation and test sets consist of 2975, 500 and 1525 images, respectively.

PASCAL Context. The dataset provides detailed semantic labels for the PASCAL VOC 2010 images, with training set and test set containing 4998 and 5105 images, respectively. There are 59 foreground categories and one background class. Following the previous work [29, 35], we evaluated our method on the 59 annotated classes.

ADE20K. The dataset is a very challenging scene parsing dataset, involving 150 dense labels and containing 20K and 2K images for training and validation respectively.

4.2 Implementation Details

We conducted all the experiments based on PyTorch [26]. The "poly" learning rate policy [20] was used (the learning rate is multiplied by $(1 - \frac{iter}{max\_iter})^{power}$ with $power = 0.9$), and the initial learning rate was set to 0.005 for Cityscapes and 0.001 for other datasets. Stochastic gradient descent (SGD) optimizer with momentum 0.9 and weight decay 0.0001 was applied to all the networks. For data augmentation, we adopted random horizontal flip, random brightness, random scaling in the range of $[0.5, 1.75]$ and random crop. Crop size is set to 769×769 for Cityscapes and 520×520 for others. The networks were trained for 120 epochs on PASCAL Context and ADE20K with batch size 16, and 240 epochs on Cityscapes with batch size 8. For evaluation, the mean Intersection-over-Union (mIoU) metric was used as the evaluation metric. Given prediction set A and target B for class c, the IoU of class

Method	PSF	mIoU (%)
Baseline	—	47.85
+ FGR	$\{1\}$	49.79
+ FGR	$\{1, \frac{1}{2}\}$	49.66
+ FGR	$\{1, \frac{1}{2}, \frac{1}{4}\}$	50.52
+ FGR	$\{1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}\}$	50.58

Table 1: Comparisons of FGR module with different pyramid scale factors. "PSF" means pyramid scale factors.

MAX	AVG	CR	mIoU (%)
			49.85
√			50.10
	√		50.24
	√	√	50.52

Table 2: Results of FGR module with different settings. "MAX" and "AVG" denote max and average pooling, "CR" means applying channel reduction after pooling.

Method	mIoU (%)	Params (M)	Inf. time (ms)
Dilated ResNet-50	47.85	33.1	11.5
ResNet-50 + GloRe [5] (Our impl.)	49.92	38.1	12.9
ResNet-50 + ASPP [4] (Our impl.)	50.47	41.4	16.8
ResNet-50 + OCR [35] (Our impl.)	50.38	39.0	13.1
ResNet-50 + FGR	50.52	48.1	14.9
ResNet-50 + PGR	50.11	38.7	12.9
ResNet-50 + FGR + PGR (serial)	50.74	53.2	15.7
ResNet-50 + FGR + PGR (parallel)	51.09	63.2	17.7

Table 3: Ablation studies on the FGR and PGR modules, "serial" and "parallel" mean adopting two modules in serial and parallel order respectively.

c is calculated by $(A \cap B)/(A \cup B)$, where $\cap$ and $\cup$ are intersection and union operations respectvely, and the mIoU is calculated by averaging the IoU of each class.

4.3 Ablation Studies

A series of ablation experiments were conducted on PASCAL Context with single scale testing for the proposed method. We adopted the dilated ResNet-50 based FCN as the baseline.

Pyramid Scale Factors. We built FGR modules with different settings of pyramid scale factors to make comparisons, and results are summarized in Table 1. Comparing with the baseline result in the first row, all the schemes of FGR module can improve the performance to some extent. We can see that the performance increases with the increase of number of branches, indicating that constructing multi-scale feature graphs can help model multi-scale context and capture information of objects with varied sizes. Although the scale factors of $\{1, 1/2, 1/4, 1/8\}$ achieved the best result (50.58%), the improvement is minor compared to factors of $\{1, 1/2, 1/4\}$ (50.52%). Therefore, to make a trade-off between accuracy and computational complexity, we adopted scale factors of $\{1, 1/2, 1/4\}$ in the final architecture.

Pooling Operation. We investigated the effect of different pooling operations in the FGR module. As shown in Table 2, constructing pyramid graphs without pooling operation achieved the worst performance, while average pooling worked better than max pooling.

Channel Reduction. We explored the performance of the FGR module with different positions for applying channel reduction operation. As shown in Table 2, applying the operation after pooling improved the result by 0.28% (50.24% $\rightarrow$ 50.52%).

Effects of FGR and PGR. We conducted experiments to evaluate the effects of FGR and PGR modules and compare with other context aggregation methods, i.e., GloRe [5], ASPP

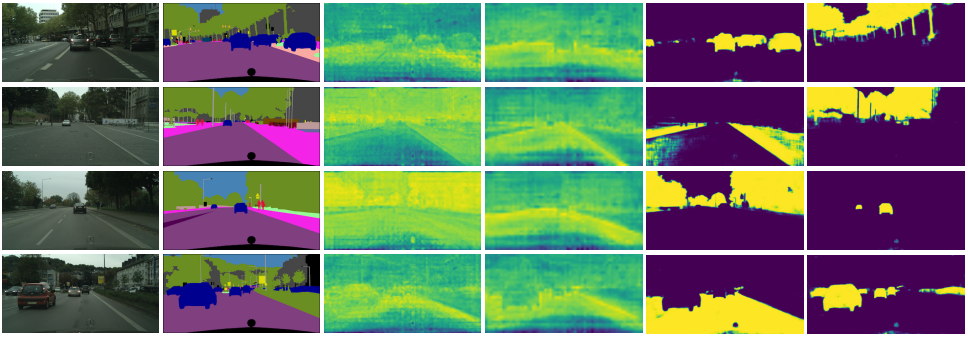
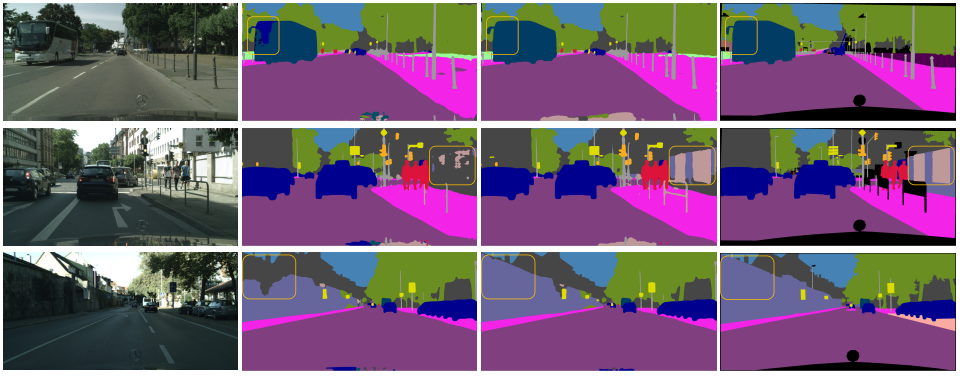


Figure 3: Visualizations of projection matrices. First two columns are input images and ground truths. Columns 3 and 4 are projection weights of two vertices in the feature graphs, and last two columns are the projection matrices of two vertices in the probabilistic graphs.



(a) Image (b) FCN (c) Ours (d) Ground truth

Figure 4: Qualitative comparison results on the Cityscapes validation set.

[4] and OCR [35] modules. To ensure fairness, we reproduced these methods under the same experimental settings. As shown in Table 3, when adding FGR module, performance was improved by 2.67% (47.85% $\rightarrow$ 50.52%), and 2.26% (47.85% $\rightarrow$ 50.11%) by adopting PGR module. Moreover, combining the two modules in parallel achieved better result of 51.09% than in serial, which also outperformed other approaches, demonstrating the effectiveness of the proposed method. Further more, we measured the model parameters and inference time on a NVIDIA Titan V100 GPU with input size 520×520.

4.4 Visualizations and Analysis

We provide visualizations of projection weights on the Cityscapes dataset in Figure 3. Specifically, the first and second columns are input images and the ground truth scene parsing masks, respectively. In the third and fourth columns, we select two vertices of the feature graphs and show their corresponding projection weights, *i.e.*, A_i^{cf}, as heatmaps. It can be seen that different vertices correspond to different patterns (the brighter pixels, the higher response), such as the sky and ground, and long-range contextual information is captured. Besides, relevant features are also aggregated by the same vertex, such as the cars and road.

Method	Backbone	Cityscapes	ADE20K	PASCAL Context
PSPNet [38]	ResNet-101	78.4	43.29	47.8
GloRe [6]	ResNet-101	80.9	–	–
EncNet [36]	ResNet-101	–	44.65	51.7
DUpsampling [30]	Xception-71	–	–	52.5
SGR [18]	ResNet-101	–	44.32	52.5
SVCNet [7]	ResNet-101	81.0	–	53.2
ISA [13]	ResNet-101	81.4	45.04	54.1
ANNet [43]	ResNet-101	81.3	45.24	52.8
CCNet [14]	ResNet-101	81.4	45.22	–
DANet [8]	ResNet-101	81.5	45.32	52.6
DMNet [9]	ResNet-101	–	45.50	54.4
OCR [35]	ResNet-101	**81.8**	45.28	54.8
SpyGR [17]	ResNet-101	81.6	–	52.8
GINet[32]	ResNet-101	–	45.54	54.9
Ours	ResNet-101	**81.8**	**45.67**	**55.1**

Table 4: Comparisons with the state-of-the-art methods on the test set of Cityscapes, validation sets of ADE20K and PASCAL Context, results are reported in terms of mIoU (%).

While in the last two columns, we visualize two projection matrices, *i.e.*, P, of probabilistic graph vertices as heatmaps. Comparing with the vertices in feature graphs, we can see that the vertices in probabilistic graph correspond to specific classes (*e.g.*, car, tree and road).

Qualitative comparisons with the baseline are shown in Figure 4, and yellow squares are used to mark the challenging objects. In the FCN model, large objects (*e.g.*, bus in the first row, fences in the second row and building in the third row) are hard to be covered by valid receptive fields and this would lead to inconsistent results. Due to the long-range dependencies modelling performed by our method, this issue can be efficiently addressed.

4.5 Comparisons with State-of-the-Art Methods

Based on the ablation studies, we designed the dual graph reasoning modules and adopted them in parallel at the top of dilated ResNet-101 backbone. We then evaluated its performances on three benchmark datasets: PASCAL Context, Cityscapes and ADE20K using multi-scale testing strategy, results and comparisons with other methods are shown in Table 4. The proposed method outperformed the state-of-the-art methods on these benchmarks.

5 Conclusions

For scene parsing, we propose two graph-based context aggregation modules to adaptively aggregate contextual information and model long-rang dependencies using graph reasoning. A feature graph reasoning module is introduced to construct pyramid feature graphs containing multi-level feature representations to help feature learning, and a probabilistic graph reasoning module is presented to construct a probabilistic graph consisting of class-dependent representations to emphasize object contextual information. The ablation studies show that the method can significantly improve segmentation performance, and its efficacy has been demonstrated by achieving state-of-the-art results on various benchmark datasets.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017.

[2] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018.

[4] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 801–818, 2018.

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

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016.

[7] Henghui Ding, Xudong Jiang, Bing Shuai, Ai Qun Liu, and Gang Wang. Semantic correlation promoted shape-variant context for segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8885–8894, 2019.

[8] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019.

[9] Junjun He, Zhongying Deng, and Yu Qiao. Dynamic multi-scale filters for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3562–3572, 2019.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[11] Hanzhe Hu, Deyi Ji, Weihao Gan, Shuai Bai, Wei Wu, and Junjie Yan. Class-wise dynamic graph convolution for semantic segmentation. *arXiv preprint arXiv:2007.09690*, 2020.

[12] Miao Hu, Yali Li, Lu Fang, and Shengjin Wang. Aˆ 2-fpn: Attention aggregation based feature pyramid network for instance segmentation. *arXiv preprint arXiv:2105.03186*, 2021.

[13] Lang Huang, Yuhui Yuan, Jianyuan Guo, Chao Zhang, Xilin Chen, and Jingdong Wang. Interlaced sparse self-attention for semantic segmentation. *arXiv preprint arXiv:1907.12273*, 2019.

[14] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 603–612, 2019.

[15] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

[16] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[17] Xia Li, Yibo Yang, Qijie Zhao, Tiancheng Shen, Zhouchen Lin, and Hong Liu. Spatial pyramid based graph reasoning for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8950–8959, 2020.

[18] Xiaodan Liang, Zhiting Hu, Hao Zhang, Liang Lin, and Eric P Xing. Symbolic graph reasoning meets convolutions. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 1858–1868, 2018.

[19] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1925–1934, 2017.

[20] Wei Liu, Andrew Rabinovich, and Alexander C Berg. Parsenet: Looking wider to see better. *arXiv preprint arXiv:1506.04579*, 2015.

[21] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[22] Sachin Mehta, Mohammad Rastegari, Anat Caspi, Linda Shapiro, and Hannaneh Hajishirzi. Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 552–568, 2018.

[23] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 891–898, 2014.

[24] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1520–1528, 2015.

[25] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016.

[26] 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. In *Advances in Neural Information Processing Systems Workshops*, 2017.

[27] Rudra PK Poudel, Ujwal Bonde, Stephan Liwicki, and Christopher Zach. Contextnet: Exploring context and detail for semantic segmentation in real-time. In *British Machine Vision Conference*, 2018.

[28] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 234–241, 2015.

[29] Ke Sun, Yang Zhao, Borui Jiang, Tianheng Cheng, Bin Xiao, Dong Liu, Yadong Mu, Xinggang Wang, Wenyu Liu, and Jingdong Wang. High-resolution representations for labeling pixels and regions. *arXiv preprint arXiv:1904.04514*, 2019.

[30] Zhi Tian, Tong He, Chunhua Shen, and Youliang Yan. Decoders matter for semantic segmentation: Data-dependent decoding enables flexible feature aggregation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3126–3135, 2019.

[31] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[32] Tianyi Wu, Yu Lu, Yu Zhu, Chuang Zhang, Ming Wu, Zhanyu Ma, and Guodong Guo. Ginet: Graph interaction network for scene parsing. In *European Conference on Computer Vision*, pages 34–51, 2020.

[33] Maoke Yang, Kun Yu, Chi Zhang, Zhiwei Li, and Kuiyuan Yang. Denseaspp for semantic segmentation in street scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3684–3692, 2018.

[34] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[35] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. *arXiv preprint arXiv:1909.11065*, 2019.

[36] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7151–7160, 2018.

[37] Songyang Zhang, Xuming He, and Shipeng Yan. Latentgnn: Learning efficient non-local relations for visual recognition. In *International Conference on Machine Learning*, pages 7374–7383, 2019.

[38] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2881–2890, 2017.

[39] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images. In *Proceedings of the European Conference on Computer Vision*, pages 405–420, 2018.

[40] Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin, and Jiaya Jia. Psanet: Point-wise spatial attention network for scene parsing. In *Proceedings of the European Conference on Computer Vision*, pages 267–283, 2018.

[41] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*, 2014.

[42] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019.

[43] Zhen Zhu, Mengde Xu, Song Bai, Tengteng Huang, and Xiang Bai. Asymmetric non-local neural networks for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 593–602, 2019.