Class-wise Dynamic Graph Convolution for Semantic Segmentation

Hanzhe Hu[1][0000−0003−2799−2655]*, Deyi Ji[2][0000−0001−7561−9789], Weihao Gan[2], Shuai Bai[3], Wei Wu[2], and Junjie Yan[2]

[1] Peking University, Beijing, China
[2] SenseTime Group Limited, Beijing, China
[3] Beijing University of Posts and Telecommunications, Beijing, China
huhz@pku.edu.cn, {jideyi,ganweihao,wuwei, yanjunjie}@sensetime.com, baishuai@bupt.edu.cn

Abstract. Recent works have made great progress in semantic segmentation by exploiting contextual information in a local or global manner with dilated convolutions, pyramid pooling or self-attention mechanism. In order to avoid potential misleading contextual information aggregation in previous works, we propose a class-wise dynamic graph convolution(CDGC) module to adaptively propagate information. The graph reasoning is performed among pixels in the same class. Based on the proposed CDGC module, we further introduce the Class-wise Dynamic Graph Convolution Network(CDGCNet), which consists of two main parts including the CDGC module and a basic segmentation network, forming a coarse-to-fine paradigm. Specifically, the CDGC module takes the coarse segmentation result as class mask to extract node features for graph construction and performs dynamic graph convolutions on the constructed graph to learn the feature aggregation and weight allocation. Then the refined feature and the original feature are fused to get the final prediction. We conduct extensive experiments on three popular semantic segmentation benchmarks including Cityscapes, PASCAL VOC 2012 and COCO Stuff, and achieve state-of-the-art performance on all three benchmarks.

Keywords: Semantic Segmentation, Graph Convolution, Coarse-to-fine Framework

1 Introduction

Semantic Segmentation is a fundamental and challenging problem in computer vision, which aims to assign a category label to each pixel in an image. It has been widely applied to many scenarios, such as autonomous driving, scene understanding and image editing.

Recent state-of-the-art semantic segmentation methods based on the fully convolutional network(FCN) [23] have made great progress. To capture the long-range contextual information, the atrous spatial pyramid pooling(ASPP) module

* This work is done when Hanzhe Hu is an intern at SenseTime Group Limited.

in DeepLabv3 [6] aggregates spatial regularly sampled pixels at different dilated rates and the pyramid pooling module in PSPNet [42] partitions the feature maps into multiple regions before pooling. More comprehensively, PSANet [43] was proposed to generate dense and pixel-wise contextual information, which learns to aggregate information via a predicted attention map. Non-local Network [31] adopts self-attention mechanism, which enables every pixel to receive information from every other pixels in the image, resulting in a much complete pixel-wise representation.

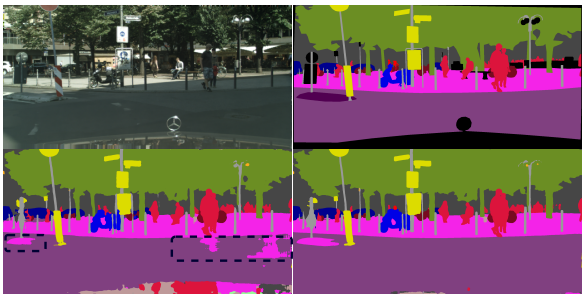

Fig. 1. Viusal example from left to right, top to bottom is : original image, groundtruth, deeplabv3 result, the proposed CDGCNet result. From the two indicated regions, our method preserves more contextual details and accurate prediction along boundaries.

However, the ways of utilizing the contextual information in existing approaches are still problematic. From one point of view, larger receptive field in deeper network is necessary for semantic prediction. Also, dilated based or the pooling based methods take even larger contextual information into consideration. These two operations are neither adaptive nor friendly to pixel-wised segmentation prediction problem. Another view of self-attention based methods (PSANet [43], Non-local Network [31], and etc [12,15,38,18]) is that, pixels from long-range non-local regions have different feature representations, which results in major issues on two aspects when optimizing the convolution neural network. First, contextual information is learned from previous network layers by considering the local and non-local cues. Considering the large variations and uncorrelations in contextual representations, weighted convoluting all the regions together results in difficulties of learning discriminative pixel-level features. For example, feature of a sky location with neighborhood tree region should be different from the one of a sky location with building region, which should not be learned together. Second, contextual information is also class-specific. That means, feature of a tree region is not proper to contribute to the learning of a sky region. The target is to directly distinguish whether the region is a sky region or not.

Aiming to address the above issues, we propose the Class-wise Dynamic Graph Convolution Network (CDGCNet), which can efficiently utilize the long-

range contextual dependencies and aggregate the useful information for better pixel label prediction. Since graph convolution is remarkable at leveraging relations between nodes and can serve as a suitable reasoning method. It is worth noting that self-attention method is actually to build a fully-connected graph, so we further improve the structure of plain GCN for better performance. First, we adopt the class-wise strategy to construct the graph (node and edge) for each class, so that the useful information for each class can be independently learned. Second, for the graph of each class, not all the context regions are included during graph reasoning. Specifically, the hard positive and negative regions are dynamically identified into the graph transform. With these two designs in graph, the most important contextual information can be exploited for pixel level semantic prediction.

The overall framework of the proposed CDGCNet method is shown in Fig. 2, which follows the coarse-to-fine paradigm. The first part is a simple but complete semantic segmentation network, called basic network, which can generate coarse prediction map and it can be any of state-of-the-art semantic segmentation architectures. The second part is the CDGC module. Firstly, the CDGC module takes coarse prediction map and feature map from the basic network as inputs, and transforms the prediction map into class mask to extract node features from different classes for graph construction. After that, for each class, dynamic graph convolution is performed on the constructed graph to learn the feature aggregation and weight allocation. Finally, the refined feature and the original feature are fused to get the final prediction.

The main contributions of this paper are summarized as follows:

- The proposed CDGCNet utilizes a class-wise learning strategy so that semantically related features are considered for contextual learning.
- During the graph construction on each class, hard positive and hard negative information are dynamically sampled from the coarse segmentation result, which avoids heavy graph connections and benefits the feature learning.
- We conduct extensive experiments on several public datasets, and obtain state-of-the-art performances on the Cityscapes [9], PASCAL VOC 2012 [11] and COCO Stuff [2] datasets.

2 Related Work

Semantic Segmentation. Benefiting from the success of deep neural networks [17,29,14], semantic segmentation has achieved great progress. FCN [23] is the first approach to adopt fully convolutional network for semantic segmentation. Later, many FCN-baed works are proposed, such as UNet [26], SegNet [1], RefineNet [22], PSPNet [42], DeepLab series [4,5,6,7]. Chen *et al.* [5] and Yu *et al.* [37] removed the last two downsample layers to obtain a dense prediction and utilized dilated convolutions to enlarge the receptive field. In our model, we also adopt the above paradigm to get a better feature map and hence, improve the performance of the model.

Context. Context plays a critical role in various vision tasks including semantic segmentation. Many works are proposed to generate better feature representations by exploiting better contextual information. From the spatial perspective, DeepLab v3 [6] employs multiple atrous convolutions with different dilation rates to capture contextual information, while PSPNet [42] employs pyramid pooling over sub-regions of four pyramid scales to harvest information. These methods, however, are all focusing on enlarging receptive fields in a local perspective and hence lose global context information. While from the attention perspective, Wang *et al.* [31] extend the idea of self-attention from transformer [30] into the vision field and proposed the non-local module to generate the attention map by calculating the correlation matrix between each spatial point in the feature map, and then the attention map guides the dense contextual information aggregation. Later, DANet [12] applied both spatial and channel attention to gather information around the feature maps . Unlike works mentioned above, our proposed module separately allocates attention to pixels belonging to the same category, effectively avoiding wrong contextual information aggregation.

Graph Reasoning. Graph-based methods have been very popular these days and shown to be an efficient way of relation reasoning. CRF [3] is proposed based on the graph model for image segmentation and works as an effective postprocessing method in DeepLab [5]. Recently, Graph Convolution Networks(GCN) [16] are proposed for semi-supervised classification, and Wang *et al.* [32] use GCN to capture relations between objects in video recognition tasks. Later, a few works based on GCN have been proposed onto the semantic segmentation problem, including [8,20,19], which all similarly model the relations between regions of the image rather than individual pixels. Concretely, clusters of pixels are defined as the vertices of the graph, hence graph reasoning is performed in the intermediate space projected from the original feature space to reduce computation cost. Different from these recent GCN-based methods, we perform graph convolution in a class-wise manner, where GCNs are employed only to the nodes in the same category, leading to a better feature learning. The refined features thus can provide a better prediction result in semantic segmentation task.

3 Approach

In this section, we will describe the proposed class-wise dynamic graph convolution (CDGC) module in detail. Firstly, we will revisit the basic knowledge of graph convolutional network. Then we will present a general framework of our network and introduce class-wise dynamic graph convolution module which separately performs graph reasoning on the pixels within the same category, hence producing a refined prediction map for semantic segmentation. Finally, we will bring out the supervision manner of the proposed model.

3.1 Preliminaries

Graph Convolution. Given an input feature $X \in \mathbb{R}^{N \times D}$, where N is the number of nodes in the feature map and D is the feature dimension, we can

build a feature graph G from this input feature. Specifically, the graph G can be formulated as $G = \{V, \varepsilon, A\}$ with V as its nodes, ε as its edges and A as its adjacency matrix. Normally, the adjacency matrix A is a binary matrix, in practice, we try many ways to construct the graph, including top-k binary matrix or dynamic learnable matrix, and further design a novel dynamic sampling method to construct the graph and perform extensive experiments to verify its validity. Intuitively, unlike standard convolutions which operates on a local regular grid, the graph enables us to compute the response of a node based on its neighbors defined in the adjacency matrix, hence receiving a much wider receptive field than regular convolutions. Formally, the graph convolution is defined as,

$$Z = \sigma(AXW), \tag{1}$$

where $\sigma(\cdot)$ denotes the non-linear activation function, $A \in \mathbb{R}^{N\times N}$ is the adjacency matrix measuring the relations of nodes in the graph and $W \in \mathbb{R}^{D\times D}$ is the weight matrix. In our experiments, we use ReLU as activation function and perform experiments with different graph construction methods.

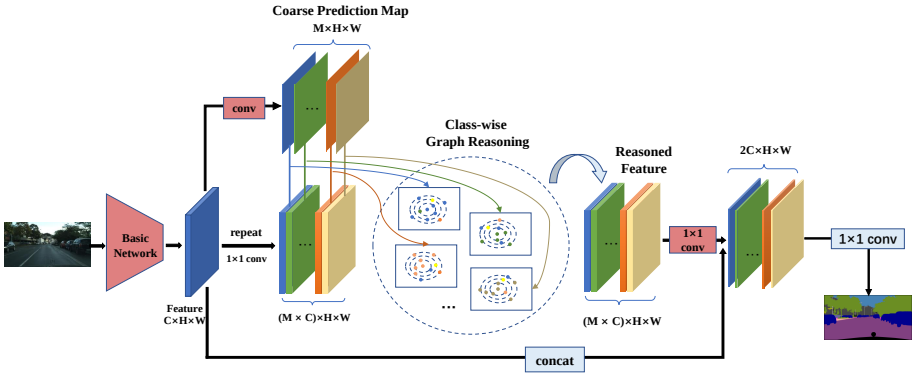

Fig. 2. An Overview of the Class-wise Dynamic Graph Convolution Network. Given an input image, we first feed it into the basic segmentation network to get the high-level feature map and the corresponding coarse segmentation result. Then the CDGC module is applied to preform graph reasoning along nodes of the feature map, producing a refined feature which is subsequently fused with the original feature to get the final refined segmentation result. Specially, in the class-wise graph reasoning part, different colors of lines and dots denote different classes of pixels, under the guidance of coarse prediction map, most positive pixels are sampled while also harvesting few hard pixels in different colors from the target color.

3.2 Overall Framework

As illustrated in Fig. 2, we present the Class-wise Dynamic Graph Convolution Network to adaptively capture long-range contextual information. We use the

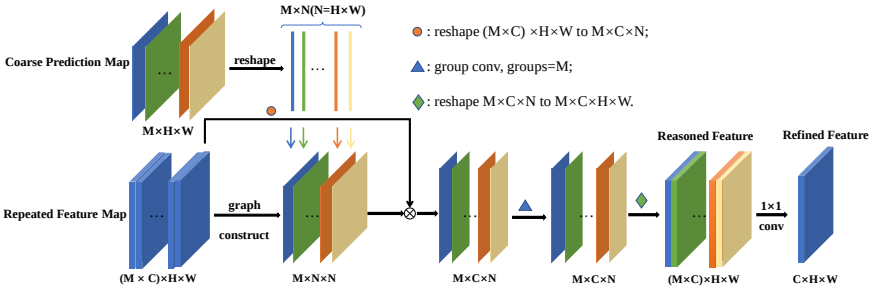

Fig. 3. The details of Class-wise Dynamic Graph Convolution Module.

ResNet-101 pretrained on the ImageNet dataset as the backbone, replace the last two down-sampling operations and employ dilation convolutions in the subsequent convolutional layers, hence enlarging the resolution and receptive field of the feature map, so the output stride becomes 8 instead of 16.

Our model consists of two parts: basic network and CDGC module. Specifically, we adopt ResNet-101 together with atrous spatial pyramid pooling(ASPP) as the basic complete segmentation network. An input image is passed through the backbone and ASPP module, then produces a feature map $X \in \mathbb{R}^{C\times H\times W}$, where C,H,W represent channel number, height and width respectively. Then we apply a convolution layer to realize the dimension reduction and the feature X will participate in two different branches. The first branch is the classification step which produces the coarse segmentation prediction map. After that, the prediction map is transformed into masks for different classes, the masks and the feature X are subsequently fed into the CDGC module to perform class-wise graph reasoning. And the output feature of our CDGC module is concatenated with the input feature, and refined through a 1×1 conv to get the final refined segmentation result.

3.3 Class-wise Dynamic Graph Convolution Module

The detailed structure of CDGC module is shown in Fig. 3. It consists of two subsequent processes, including graph construction and reasoning. The proposed module is based on a coarse-to-fine framework, where the input is the feature map X, coarse prediction map and the output is the refined feature map.

Class-wise Learning Strategy. Different from previous works [8,20,19] where graph construction is performed on all the nodes of different classes in the feature map, we adopt a class-wise learning strategy. There are several advantages. First, contextual information from different classes is considered separately so that the irrelevant region can be excluded to avoid the difficulty of learning. Second, it is easy to hard-mine the important information for a binary task (determine whether it is the target class or not) compared to the multi-class task learning.

Specifically, in the training process, a coarse-to-fine framework is adopted. The coarse prediction can be generated from a basic network. Each coarse predicted category is utilized to filter out the corresponding category and perform a graph construction based on the filtering operation. Hence, graph reasoning and information transmission only occur inside the chosen category, protecting the process of context aggregation from the interference of features in other categories.

Graph Construction. (1) Similarity Graph. Intuitively, we can build the graph (which is adjacency matrix in our formulation) based on the similarity between different nodes, for two node features x_i, x_j, the pairwise similarity between two nodes is defined as,

$$F(\boldsymbol{x_i}, \boldsymbol{x_j}) = \phi(\boldsymbol{x_i})^T \phi'(\boldsymbol{x_j}), \tag{2}$$

where ϕ, ϕ' denote two different transformations of the original features. In practice, we adopt linear transformations, hence $\phi(\boldsymbol{x}) = \boldsymbol{w}\boldsymbol{x}$ and $\phi'(\boldsymbol{x}) = \boldsymbol{w'}\boldsymbol{x}$. The parameters $\boldsymbol{w}$ and $\boldsymbol{w'}$ are both $D \times D$ dimensions weights which can be learned via back propagation, forming a dynamically learned graph construction method. After computing the similarity matrix, we perform normalization on each row of the matrix so that the sum of all the edge values connected to one node i will be 1. In practice, we choose softmax as normalization function, so the output adjacency matrix will be,

$$A_{ij} = \frac{exp(F(\boldsymbol{x_i}, \boldsymbol{x_j}))}{\sum_{j=1}^{N} exp(F(\boldsymbol{x_i}, \boldsymbol{x_j}))} \tag{3}$$

(2) Dynamic Sampling. The original sampling method adopts a fully-connected fashion for pixels in the same category. However, since the prediction mask is obtained from a coarse segmentation result, it is possible that the sampled pixels are not actually belong to the same category, which makes the sampled set include 'easy positive' part and 'hard negative' part. In order to allocate enough attention to these hard-to-classify pixels, we develop a dynamic sampling method which focuses on selecting out these hard pixels. As shown in Fig. 4, in the training process, we take coarse segmentation mask and groundtruth mask as input, and compute the intersection set between them, which is pure 'easy positive' part. Formally, we denote the coarse segmentation mask, groundtruth mask set as C and G respectively, hence the intersection set can be denoted as $C \cap G$. Then with coarse segmentation mask subtracting the intersection set, the rest part is pure 'hard negative' denoted as $C - C \cap G$. Similarly, with groundtruth mask getting rid of the intersection set, the rest part is pure 'hard positive', denoted as $G - C \cap G$. Besides, some ratio of 'easy positive' samples are needed to guide the learning of these hard pixels, so we randomly choose some ratio of pixels from the intersection set which consists of pure 'easy positive' samples, so we finally get the sampled set denoted as,

$$\begin{aligned} Sampled &= C - C \cap G + G - C \cap G + ratio \cdot C \cap G \\ &= C \cup G - (1 - ratio) \cdot C \cap G \end{aligned} \tag{4}$$

Therefore, with this dynamic sampling method, our graph construction process can pay enough attention to these hard pixels.

Specifically, dynamic sampling is only used at the training stage but not the inference stage. At the training stage, we use both coarse prediction mask and groundtruth mask to mine hard positive and negative samples in a class-wise manner. Besides, some easy positive samples are also selected to guide the hard samples learning. All these samples compose the graph nodes for the training stage. At the inference stage, pixels in the same category according to the coarse prediction mask are sampled to construct the graph.

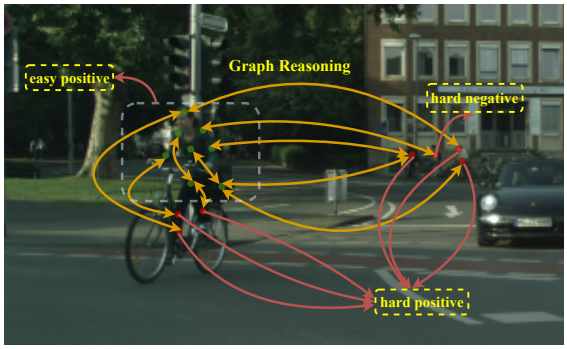

Fig. 4. Illustration of dynamic sampling method. For one category 'rider' in this image, green and red points denote easy and hard samples, respectively. Hard positive samples consist of distant objects and boundaries. And hard negative samples denote the illegible object (person) in this image, which is likely to be recognized as rider.

Graph Reasoning. Discriminative pixel-level feature representations are essential for semantic segmentation, which could be obtained by the proposed graph convolution based module in a class-wise manner. By exploiting the relations between pixels sampled by category, the intra-class consistency can be preserved and moreover, inter-class discrepancy can also be enhanced with our dynamic sampling method.

The detailed structure of CDGC module is shown in Fig. 3. The module takes the repeated feature map $\boldsymbol{X} \in \mathbb{R}^{(M\times C)\times H\times W}$ and coarse prediction map $P \in \mathbb{R}^{M\times H\times W}$ as input, where M, C, H, W denote the number of classes in the dataset, dimension of the feature map, height and width, respectively. Inspired by point cloud segmentation [25,33], we treat nodes in the feature map as the vertexes in the graph. Therefore, we transform the feature map to the graph representation: $X \in \mathbb{R}^{M\times C\times N}$, where $N = H \times W$ denotes the number of nodes in the feature map. Similarly, we transform the coarse prediction map into $P \in \mathbb{R}^{M\times N}$. Applying the graph construction methods discussed above, we can obtain the adjacency matrix of the feature map for each category, treating each graph feature $x \in \mathbb{R}^{C\times N}$ separately (M in total), thus producing M adjacency matrices integrated as $A \in \mathbb{R}^{M\times N\times N}$.

Following the paradigm of graph convolution, we multiply the adjacency matrix and the transposed feature map to get the sampled feature map $X \in \mathbb{R}^{M\times C\times N}$. Subsequently, group graph convolution is performed, resulting in a feature $X \in \mathbb{R}^{M\times C\times N}$ which will be reshaped back to the original grid form: $X \in \mathbb{R}^{M\times C\times H\times W}$. Then a 1×1 conv is performed to learn the weights of adaptively aggregrating feature maps for M classes, producing a refined feature $X \in \mathbb{R}^{C\times H\times W}$. Once obtaining the refined feature map, we combine this feature map with the input feature map to get the final output. Specifically, the combine method is concatenation or summation. Finally ,the output feature is passed through the conventional 1×1 convolution layer to get the final segmentation prediction map.

3.4 Loss Function

Both coarse and refined output are supervised with the semantic labels. Moreover, following normal practice in previous state-of-the-art works [42,44,39], we add the auxiliary supervision for improving the performance, as well as making the network easier to optimize. Specifically, the output of the third stage of our backbone ResNet-101 is further fed into a auxiliary layer to produce a auxiliary prediction, which is supervised with the auxiliary loss. As for the main path, coarse segmentation result and refined segmentation result are produced and hence require proper supervision. We apply standard cross entropy loss to supervise the auxiliary output and the coarse prediction map, and employ OHEM loss [27] for the refined prediction map. In a word, the loss can be formulated as follows,

$$L = \alpha \cdot l_c + \beta \cdot l_f + \gamma \cdot l_a \tag{5}$$

where α, β, γ are used to balance the coarse prediction loss l_c, refined prediction loss l_f and auxiliary loss l_a.

4 Experiments

To evaluate the performance of our proposed CDGC module, we carry out extensive experiments on benchmark datasets including Cityscapes [9], PASCAL VOC 2012 [11] and COCO Stuff [2]. Experimental results demonstrate that the proposed method can effectively boost the performance of the state-of-the-art methods. In the following section, we will introduce the datasets and implementation details, and then perform ablation study on Cityscapes dataset. Finally, we report the results on PASCAL VOC 2012 dataset and COCO Stuff dataset.

4.1 Datasets and Evaluation Metrics

Cityscapes. The Cityscapes dataset [9] is tasked for urban scene understanding, which contains 30 classes and only 19 classes of them are used for scene parsing evaluation. The dataset contains 5000 finely annotated images and 20000

coarsely annotated images. The finely annotated 5000 images are divided into 2975/500/1525 images for training, validation and testing.

PASCAL VOC 2012. The PASCAL VOC 2012 dataset [11] is one of the most competitive semantic segmentation dataset which contains 20 foreground object classes and 1 background class. The dataset is split into 1464/1449/1556 images for training, validation and testing. [13] has augmented this dataset with annotations ,resulting in 10582 train-aug images.

COCO Stuff. The COCO Stuff dataset [2] is a challenging scene parsing dataset containing 59 semantic classes and 1 background class. The training and test set consist of 9K and 1K images respectively.

In our experiments, the mean of class-wise Intersection over Union (mIoU) is used as the evaluation metric.

4.2 Implementation Details

We choose the ImageNet pretrained ResNet-101 as our backbone and remove the last two down-sampling operations, and employ dilated convolutions in the subsequent convolution layers, making the output stride equal to 8. For training, we use the stochastic gradient descent(SGD) optimizer with initial learning rate 0.01, weight decay 0.0005 and momentum 0.9 for Cityscapes dataset. Moreover, we adopt the 'poly' learning rate policy, where the initial learning rate is multiplied by $(1 - \frac{iter}{max\_iter})^{power}$ with power=0.9. For Cityscapes dataset, we adopt the crop size as 769×769, batch size as 8 and training iterations as 30K. For PASCAL VOC 2012 dataset, we set the initial learning rate as 0.001, weight decay as 0.0001, crop size as 513×513, batch size as 16 and training iterations as 30K. For COCO Stuff dataset, we set initial learning rate as 0.001, weight decay as 0.0001, crop size as 520×520, batch size as 16, and training iterations as 60K. Moreover, the loss weights α, β, γ are set to be 0.6, 0.7 and 0.4 respectively.

4.3 Ablation Study

In this subsection, we conduct extensive ablation experiments on the validation set of Cityscapes with different settings for our proposed CDGCNet.

The impact of class-wise learning strategy. We use the dilated ResNet-101 as the baseline network, and final segmentation result is obtained by directly upsampling the output. To evaluate the effectiveness of the proposed class-wise learning strategy, we carry out the experiments where plain GCN and class-wise GCN are adopted separately. Concretely, plain GCN is realized by simply performing graph construction operation on the feature map obtained from the backbone, while class-wise GCN is realized in a class-wise manner. Their graph construction methods are similar. As shown in Table 1, the proposed class-wise GCN reasoning performs better than the plain GCN. Since plain-CGN adopts fully connected fashion onto the input feature map, it serves similarly as self-attention based method, which is likely to mislead the contextual information aggregation with features from pixels of other categories, while our method, on the other hand, is capable of avoiding this kind of problem.

Table 1. Performance comparisons of our proposed class-wise GCN and plain-GCN on Cityscapes validation set.

Method	mIOU (%)
ResNet-101 Baseline	76.3
ResNet-101 + plain-GCN	78.2
ResNet-101 + class-GCN	79.4

Table 2. Detailed performance comparisons of our proposed Class-wise Dynamic Graph Convolution module on Cityscapes validation set.

Method	mIOU (%)
ResNet-101 Baseline	76.3
ResNet-101 + ASPP	78.4
ResNet-101 + CDGC(concat)	79.4
ResNet-101 + CDGC(sum)	79.2
ResNet-101 + ASPP + CDGC(sum)	79.9
ResNet-101 + ASPP + CDGC(concat)	80.0

The impact of CDGC module. Based on the dilated ResNet-101 backbone, we subsequently add ASPP module and the proposed module to evaluate the performance, as shown in Table 2. The graph is constructed based on dynamic similarity. The result of solely adding ASPP module is 78.4%, which is about 1% lower than solely adding CDGC module. Furthermore, we perform experiments on the feature aggregation manners which include concatenation and summation. As results shown in Table 2, the CDGC module can significantly improve the performance over the baseline network by 3% in mIOU and concatenation method is slightly better than the summation one, so we will use concatenation aggregation method in later comparisons. Finally, we choose ResNet-101 plus ASPP module as our basic segmentation network and use CDGC module to get the final refined prediction map, achieving 1.6% gain in mIOU, which demonstrates that CDGC module can be easily plugged into any state-of-the-art segmentation network to further boost the performance. The effect of CDGC module can be visualized in Fig. 5. Some details and boundaries are refined compared to the coarse map predicted by the basic network. These results prove that our proposed CDGC module can significantly capture long-range contextual information together with local cue and also preserve intra-class consistency, which can effectively boost the performance of segmentation.

Comparisons of different graph construction methods. In this subsection, we evaluate the performance of our module using two different graph construction methods mentioned before. Specifically, we use ResNet-101+ASPP as basic segmentation network and the original feature is concatenated with refined feature to get the final prediction map. Table 3 indicates the performance on Cityscapes validation set by adopting different graph construction method, where 'sim' denotes the similarity graph method and 'ds' denotes the dynamic sampling method and the easy positive sampling ratio is set as [0.2, 0.4, 0.6, 0.8]. As can be seen in Table 3, as the easy positive sampling ratio grows, the performance becomes better since the easy positive samples serve as the guiding criterion for learning the reasonable weights for hard samples. From the result shown in Table 3, when sampling ratio is above 0.4, the dynamic sampling

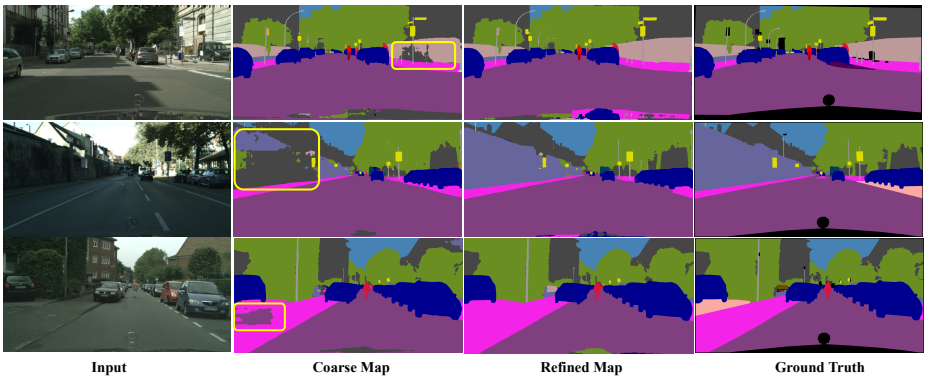

Fig. 5. Visualization results on Cityscapes validation set.

Table 3. Performance comparisons of graph construction method on Cityscapes validation set.

Method	mIOU (%)
ResNet-101 + ASPP	78.4
ResNet-101 + ASPP + CDGC(sim)	80.0
ResNet-101 + ASPP + CDGC(ds 0.2)	79.8
ResNet-101 + ASPP + CDGC(ds 0.4)	80.3
ResNet-101 + ASPP + CDGC(ds 0.6)	80.8
ResNet-101 + ASPP + CDGC(ds 0.8)	80.9
ResNet-101 + ASPP + CDGC(ds 1.0)	81.1

Table 4. Performance influences with different evaluation strategies on Cityscapes validation set.

Method	MS	Flip	mIOU (%)
CDGCNet			81.1
CDGCNet	✓		81.6
CDGCNet		✓	81.4
CDGCNet	✓	✓	**81.9**

method can outperform the similarity graph method since it gives more attention to hard samples including hard positive ones and hard negative ones while similarity graph adaptively learn the parameters of the construction weights, which may not be efficiently learned in similarity graph method.

The impact of hard samples. We further perform experiments to evaluate the impact of hard samples utilized in dynamic sampling method. At the training stage, we construct the graph with dynamic sampling method while keeping the ratio of easy positive samples as 1.0. From the result shown in Table 5, utilizing hard samples can improve the performance since extra attention can be paid to hard pixels, hence performing a better feature learning process.

The impact of evaluation strategies. Based on details discussed above, we propose Class-wise Dynamic Graph Convolution Network (CDGCNet) with ResNet-101+ASPP as basic network and dynamic sampling method to construct the graph. Like previous work [42,34,12,15,38], we also adopt the left-right flipping and multi-scale [0.75, 1.0, 1.25, 1.5, 1.75, 2.0] evaluation strategies. From Table 4, MS/Flip improves the performance by 0.8% on validation set.

Table 5. Performance comparisons of different samples used in dynamic sampling method on Cityscapes validation set.

Sample	mIOU(%)
Easy Positive	79.9
Easy Positive + Hard Positive	80.5
Easy Positive + Hard Negative	80.0
Easy Positive + Hard Positive + Hard Negative	81.1

Visualizations of class-wise features. Qualitative results are provided in Figure 6 to compare the difference of class-wise features before and after CDGC module. We use white squares to mark the challenging regions which compose of hard samples. As shown in the figure, after class-wise dynamic graph convoluton, hard pixels can be effectively resolved. In particular, in the first and third lines, hard pixels are specified to hard negative pixels and can be successfully distinguished. While in the second line, hard pixels are specified to hard positive pixels, as shown in the visualization, ambiguity is well taken care of. Moreover, with dynamic sampling method mining hard samples, boundary information is preserved and enhanced, hence producing better results.

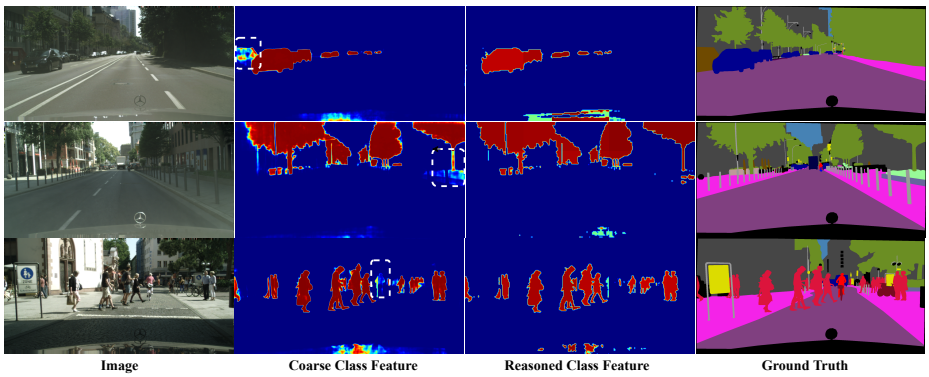

Fig. 6. Visualizations of class-wise features before and after graph convolution on Cityscapes validation set. From left to right: input image, class-wise feature before CDGC module, class-wise feature after CDGC module, ground truth. From top to bottom, the visualized category is car, vegetation and person.

4.4 Comparisons with state-of-the-arts

Furthermore, we evaluate our method on the test set of three benchmark datasets: Cityscapes, PASCAL VOC 2012 and COCO Stuff datasets. Specifically, we use

Table 6. Comparisons with State-of-the-art methods on three benchmark datatsets.

Methods	Backbone	Cityscapes mIOU(%)	PASCAL VOC 2012 mIOU(%)	COCO Stuff mIOU(%)
FCN [23]	VGG-16	-	62.2	22.7
DeepLab-CRF [5]	VGG-16	-	71.6	-
DAG-RNN [28]	VGG-16	-	-	31.2
RefineNet [22]	ResNet-101	73.6	-	33.6
GCN [24]	ResNet-101	76.9	-	-
SAC [41]	ResNet-101	78.1	-	-
CCL [10]	ResNet-101	-	-	35.7
PSPNet [42]	ResNet-101	78.4	82.6	-
BiSeNet [35]	ResNet-101	78.9	-	-
DFN [36]	ResNet-101	79.3	82.7	-
DSSPN [21]	ResNet-101	-	-	37.3
SGR [20]	ResNet-101	-	-	39.1
PSANet [43]	ResNet-101	80.1	-	-
DenseASPP [34]	DenseNet-161	80.6	-	-
GloRe [8]	ResNet-101	80.9	-	-
EncNet [40]	ResNet-101	-	82.9	-
DANet [12]	ResNet-101	81.5	82.6	39.7
CDGCNet(Ours)	ResNet-101	**82.0**	**83.9**	**40.7**

ResNet-101 as backbone, dynamic sampling method with ratio 1.0 as graph construction method. Moreover, we train the proposed CDGCNet with both training and validation set and use the multi-scale and flip strategies while testing. From Table 6, it can be observed that our CDGCNet achieves state-of-the-art performance on all three benchmark datasets.

5 Conclusions

In this paper, we have presented the Class-wise Dynamic Graph Convolution Network (CDGCNet) which can adaptively capture long-range contextual information, hence performing a reliable graph reasoning along nodes for better feature aggregation and weight allocation. Specifically, we utilize a class-wise learning strategy to enhance contextual learning. Moreover, we develop a dynamic sampling method for graph construction, which gives extra attention to hard samples, thus benefiting the feature learning. The ablation experiments demonstrate the effectiveness of CDGC module. Our CDGCNet achieves outstanding performance on three benchmark datasets, *i.e.* Cityscapes, PASCAL VOC 2012 and COCO Stuff.

References

1. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence **39**(12), 2481–2495 (2017)
2. Caesar, H., Uijlings, J., Ferrari, V.: Coco-stuff: Thing and stuff classes in context. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1209–1218 (2018)
3. 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. pp. 5103–5112 (2017)
4. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected crfs. arXiv preprint arXiv:1412.7062 (2014)
5. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: 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 (2017)
6. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017)
7. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European conference on computer vision (ECCV). pp. 801–818 (2018)
8. 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. pp. 433–442 (2019)
9. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3213–3223 (2016)
10. Ding, H., Jiang, X., Shuai, B., Qun Liu, A., Wang, G.: Context contrasted feature and gated multi-scale aggregation for scene segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2393–2402 (2018)
11. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International journal of computer vision **88**(2), 303–338 (2010)
12. 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. pp. 3146–3154 (2019)
13. Hariharan, B., Arbeláez, P., Bourdev, L., Maji, S., Malik, J.: Semantic contours from inverse detectors. In: 2011 International Conference on Computer Vision. pp. 991–998. IEEE (2011)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
15. Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., Liu, W.: Ccnet: Criss-cross attention for semantic segmentation. arXiv preprint arXiv:1811.11721 (2018)
16. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)

17. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
18. Li, X., Zhong, Z., Wu, J., Yang, Y., Lin, Z., Liu, H.: Expectation-maximization attention networks for semantic segmentation. In: The IEEE International Conference on Computer Vision (ICCV) (October 2019)
19. Li, Y., Gupta, A.: Beyond grids: Learning graph representations for visual recognition. In: Advances in Neural Information Processing Systems. pp. 9225–9235 (2018)
20. Liang, X., Hu, Z., Zhang, H., Lin, L., Xing, E.P.: Symbolic graph reasoning meets convolutions. In: Advances in Neural Information Processing Systems. pp. 1853–1863 (2018)
21. Liang, X., Zhou, H., Xing, E.: Dynamic-structured semantic propagation network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 752–761 (2018)
22. Lin, G., Milan, A., Shen, C., Reid, I.: Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1925–1934 (2017)
23. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015)
24. Peng, C., Zhang, X., Yu, G., Luo, G., Sun, J.: Large kernel matters–improve semantic segmentation by global convolutional network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4353–4361 (2017)
25. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 652–660 (2017)
26. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
27. Shrivastava, A., Gupta, A., Girshick, R.: Training region-based object detectors with online hard example mining. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 761–769 (2016)
28. Shuai, B., Zuo, Z., Wang, B., Wang, G.: Scene segmentation with dag-recurrent neural networks. IEEE transactions on pattern analysis and machine intelligence **40**(6), 1480–1493 (2017)
29. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
30. 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. pp. 5998–6008 (2017)
31. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7794–7803 (2018)
32. Wang, X., Gupta, A.: Videos as space-time region graphs. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 399–417 (2018)
33. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. ACM Transactions on Graphics (TOG) **38**(5), 146 (2019)

34. Yang, M., Yu, K., Zhang, C., Li, Z., Yang, K.: Denseaspp for semantic segmentation in street scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3684–3692 (2018)
35. Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., Sang, N.: Bisenet: Bilateral segmentation network for real-time semantic segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 325–341 (2018)
36. Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., Sang, N.: Learning a discriminative feature network for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1857–1866 (2018)
37. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122 (2015)
38. Yuan, Y., Wang, J.: Ocnet: Object context network for scene parsing. arXiv preprint arXiv:1809.00916 (2018)
39. Zhang, F., Chen, Y., Li, Z., Hong, Z., Liu, J., Ma, F., Han, J., Ding, E.: Acfnet: Attentional class feature network for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 6798–6807 (2019)
40. Zhang, H., Dana, K., Shi, J., Zhang, Z., Wang, X., Tyagi, A., Agrawal, A.: Context encoding for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7151–7160 (2018)
41. Zhang, R., Tang, S., Zhang, Y., Li, J., Yan, S.: Scale-adaptive convolutions for scene parsing. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2031–2039 (2017)
42. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2881–2890 (2017)
43. 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). pp. 267–283 (2018)
44. 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. pp. 593–602 (2019)