Dual Graph Convolutional Network for Semantic Segmentation

Li Zhang*[1]
lz@robots.ox.ac.uk
Xiangtai Li*[2]
lxtpku@pku.edu.cn
Anurag Arnab[1]
aarnab@robots.ox.ac.uk
Kuiyuan Yang[3]
kuiyuanyang@deepmotion.ai
Yunhai Tong[2]
yhtong@pku.edu.cn
Philip H.S. Torr[1]
phst@robots.ox.ac.uk

[1] Department of Engineering Science, Torr Vision Group, University of Oxford
[2] Key Laboratory of Machine Perception, School of EECS, Peking University
[3] DeepMotion AI Research

Abstract

Exploiting long-range contextual information is key for pixel-wise prediction tasks such as semantic segmentation. In contrast to previous work that uses multi-scale feature fusion or dilated convolutions, we propose a novel graph-convolutional network (GCN) to address this problem. Our *Dual Graph Convolutional Network* (DGCNet) models the global context of the input feature by modelling two orthogonal graphs in a single framework. The first component models spatial relationships between pixels in the image, whilst the second models interdependencies along the channel dimensions of the network's feature map. This is done efficiently by projecting the feature into a new, lower-dimensional space where all pairwise interactions can be modelled, before reprojecting into the original space. Our simple method provides substantial benefits over a strong baseline and achieves state-of-the-art results on both Cityscapes (82.0% mean IoU) and Pascal Context (53.7% mean IoU) datasets. Our code is available at: https://github.com/lzrobots/DGCNet

1 Introduction

Semantic segmentation is a fundamental problem in computer vision, and aims to assign an object class label to each pixel in an image. It has numerous applications including autonomous driving, augmented- and virtual reality and medical diagnosis.

An inherent challenge in semantic segmentation is that pixels are difficult to classify when considered in isolation, as local image evidence is ambiguous and noisy. Therefore, segmentation systems must be able to effectively capture contextual information in order to reason about occlusions, small objects and model object co-occurrences in a scene.

* equal contribution.

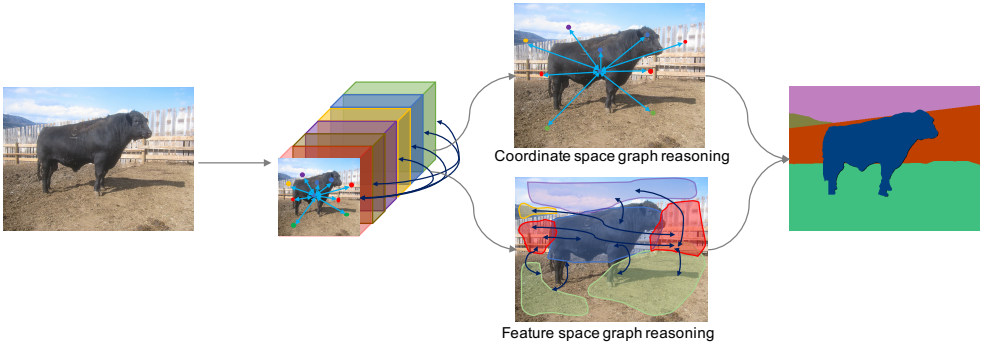

Figure 1: Our proposed *DGCNet* exploits contextual information across the whole image by proposing a graph convolutional network to efficiently propagate information along both the spatial and channel dimensions of a convolutional feature map.

Current state-of-the-art methods are all based on deep learning using fully convolutional networks (FCNs) [30]. However, the receptive field of an FCN grows slowly (only linearly) with increasing depth in the network, and its limited receptive field is not able to capture longer-range relationships between pixels in an image. Dilated convolutions [4, 43] have been proposed to remedy this. However, the resulting feature representation is dominated by large objects in the image, and consequently, performance on small objects is poor. Another direction has been to fuse multiscale features within the network [29, 49] or to use LSTMs to propagate information spatially [3, 37]. Recently, several methods based on self-attention [14, 39, 44] have also been used to learn an affinity map at each spatial position that propagates information to its neighbours. However, the memory requirements of the large affinity matrix renders these methods unsuitable for high resolution imagery (such as the Cityscapes dataset [9]).

In this paper, we use graph-convolutional networks (GCNs) [18] to effectively and efficiently model contextual information for semantic segmentation. GCNs have recently been applied to scene understanding tasks [8, 23, 24], as they are able to globally propagate information through the whole image in a manner that is conditional on the input. This provides greater representational power than methods based on Conditional Random Fields [2, 4, 51] which have historically been employed for semantic segmentation [16, 35].

As shown in Fig. 1, our proposed method consists of two primary components: the coordinate space GCN explicitly models the spatial relationships between pixels in the image, enabling our network to produce coherent predictions that consider all objects in the image, whilst the feature space GCN models interdependencies along the channel dimensions of the network's feature map. Assuming that filters in later layers of a CNN are responsive to object parts and other high-level features [45], the feature space GCN captures correlations between more abstract features in the image like object parts. After reasoning, these two complementary relation-aware feature are distributed back to the original coordinate space and added to the original feature.

Using our proposed approach, we obtain state-of-the-art results on the Cityscapes [9] and Pascal Context [31] datasets. Furthermore, to encourage reproducibility, we have publicly released our training code and models.

2 Related Work

Following the success of deep neural networks for image classification [15, 20, 38], recent works in semantic segmentation all leverage fully-convolutional networks (FCNs) [30]. A limitation of standard FCNs is their small receptive field which prevents them from taking all the contextual information in the scene into account. The DeepLab series of papers [4, 5, 6] proposed atrous or dilated convolutions and atrous spatial pyramid pooling (ASPP) to increase the effective receptive field. DenseASPP improved on [6] by densely connecting convolutional layers with different dilation rates. PSPNet [49], on the other hand, used a pyramid pooling module to fuse convolutional features from multiple scales. Similarly, encoder-decoder network structures [28, 32, 34] combine features from early- and late-stages in the network to fuse mid-level and high-level semantic features. Deeplab-v3+ [7] also followed this approach by fusing lower-level features into its decoder. Di *et al*. [25] also recursively, locally fused feature maps of every two levels in a feature pyramid into one.

Another approach has been proposed to more explicitly account for the relations between all pixels in the image. DAG-RNN [37] models a directed acyclic graph with a recurrent network that propagates information. PSANet [50] captures pixel-to-pixel relations using an attention module that takes the relative location of each pixel into account. On the other hand, EncNet [47] and DFN [42] use attention along the channel dimension of the convolutional feature map to account for global context such as the co-occurrences of different classes in the scene.

Following on from these approaches, graph neural networks [18] have also been used to model long-range context in the scene. Non-local networks [39] applied this to video understanding and object detection by learning an affinity map between all pixels in the image or video frames. This allowed the network to effectively increase its receptive field to the whole image. The non-local operator has been applied to segmentation by OCNet [44] and DANet [14] recently. However, these methods have a (sometimes prohibitively) high memory cost as the affinity matrix grows quadratically with the number of pixels in the image. To bypass this problem, several works [8, 23, 24] have modelled the dependencies between regions of the image rather than individual pixels. This is done by aggregating features from the original "co-ordinate space" to a lower-dimensional intermediate representation, performing graph convolutions in this space, and then reprojecting back onto the original co-ordinate space.

However, differently from these recent GCN methods, we propose the Dual Graph Convolutional Network (DGCNet) to model the global context of the input feature by considering *two orthogonal graphs in a single general framework*. Specifically, with different mapping strategies, we first project the feature into a new coordinate space and a non-coordinate (feature) space where global relational reasoning can be computed efficiently. After reasoning, two complementary relation-aware features are distributed back to the original coordinate space and added to the original feature. The refined feature thus contains rich global contextual information and can be further provided to the following layers to learn better task-specific representations.

3 Methodology

In this section, we first revisit the graph convolutional network in Sec. 3.1 and then introduce the formulation of our proposed DGCNet in Sec. 3.2 and 3.3. Finally, we detail the resulting

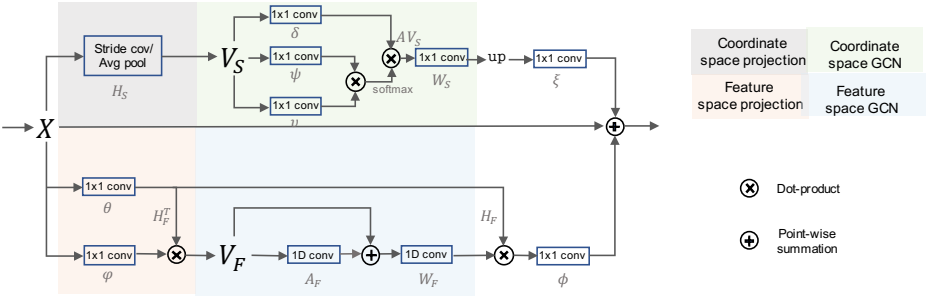

Figure 2: Illustration of our proposed DGCNet. Our method consists of two branches, which each consist of a graph convolutional network (GCN) to model contextual information in the spatial- and channel-dimensions in a convolutional feature map, X.

network architecture in Sec. 3.4.

3.1 Preliminaries

Revisiting the graph convolution. Assume an input tensor $\mathbf{X} \in \mathbb{R}^{N \times D}$, where D is the feature dimension and $N = H \times W$ is the number of locations defined on regular grid coordinates $\Omega = \{1, ..., H\} \times \{1, ..., W\}$. In standard convolution, information is only exchanged among positions in a small neighborhood defined by the filter size (*e.g.* typically 3×3). In order to create a large receptive field and capture long range dependencies, one needs to stack numerous layers after each other, as done in common architectures [15, 33]. Graph convolution [18], is a highly efficient, effective and differentiable module that generalises the neighborhood definition used in standard convolution, and allows long-range information exchange in a single layer. This is done by defining edges $\mathcal{E}$ among nodes $\mathcal{V}$ in a graph $\mathcal{G}$. Formally, following [18], graph convolution is defined as

$$\tilde{\mathbf{X}} = \sigma(\mathbf{AXW}), \tag{1}$$

where $\sigma(\cdot)$ is the non-linear activation function, $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix characterising the neighbouhood relations of the graph and $\mathbf{W} \in \mathbb{R}^{D \times \tilde{D}}$ is the weight matrix. Clearly, the graph definition and structure play a key role in determining the information propagation. Our proposed framework is motivated by building orthogonal graph spaces via different graph projection strategies to learn a better task-specific representation. As summarised in Fig. 2, we now describe how we propagate information in the coordinate space in Sec. 3.2, and in feature space in Sec. 3.3.

3.2 Graph convolution in coordinate space

Coordinate space projection. We first project the input feature into a new coordinate space $\Omega_{\mathcal{S}}$. In general, we adopt a spatial downsampling operation $\mathbf{H}_{\mathcal{S}}$ to transform the input feature $\mathbf{X}$ to a new feature $\mathbf{V}_{\mathcal{S}} \in \mathbb{R}^{\frac{H \times W}{d^2} \times D}$ in the new coordinate space $\Omega_{\mathcal{S}}$, where the d denotes the downsample rate,

$$\mathbf{V}_{\mathcal{S}} = \mathbf{H}_{\mathcal{S}}\mathbf{X}. \tag{2}$$

We consider two different operations for $\mathbf{H}_{\mathcal{S}}$: (1). Parameter-free operation. We take the downsampling operation $\mathbf{H}_{\mathcal{S}}$ to be average pooling which requires no additional learnable parameters. (2). Parameterized operation. For efficiency, a downsampling rate of d is achieved by chaining $\log_2(d)$ depth-wise convolution layers, each with a stride of 2 and kernel size of 3×3.

Coordinate graph convolution. After projecting the features into the new coordinate space $\Omega_{\mathcal{S}}$, we can build a lightweight fully-connected graph with adjacency matrix $\mathbf{A}_{\mathcal{S}} \in \mathbb{R}^{\frac{H}{d} \times \frac{W}{d}}$ for diffusing information across nodes. Note that the nodes of the graph aggregate information from a "cluster" of pixels, and the edges measure the similarity between these clusters.

The global relational reasoning is performed on the downsampled feature $\mathbf{V}_{\mathcal{S}}$ to model the interaction between the features of the corresponding nodes. In particular, we adopt three learnable linear transformations (δ, ψ, υ) on the feature $\mathbf{V}_{\mathcal{S}}$ to produce the message, $\mathbf{M}_{\mathcal{S}}$

$$\mathbf{M}_{\mathcal{S}} = f(\delta(\mathbf{V}_{\mathcal{S}}),\ \psi(\mathbf{V}_{\mathcal{S}})^{\top})\upsilon(\mathbf{V}_{\mathcal{S}})\mathbf{W}_{\mathcal{S}}, \tag{3}$$

where $f(\delta(\mathbf{V}_{\mathcal{S}}),\ \psi(\mathbf{V}_{\mathcal{S}})^{\top})$ gives the adjacency matrix in Eq. 1, f is the dot-product operation.

Reprojection. After the reasoning, we map the new features $\mathbf{M}_{\mathcal{S}}$ back into the original coordinate space ($\mathbb{R}^{H \times W \times D}$) to be compatible with a regular CNN. Opposite to the downsample operation $\mathbf{H}_{\mathcal{S}}$ used in graph projection, we simply perform *upsampling* as the reprojection operation. In particular, nearest neighbour interpolation, $interp(\cdot)$, is adopted to resize $\mathbf{M}_{\mathcal{S}}$ to the original spatial input size (N). Hence, the output map is computed as $\tilde{\mathbf{X}}_{\mathcal{S}} = \xi(interp(\mathbf{M}_{\mathcal{S}}))$, where $\xi(\cdot)$ is a 1×1 convolution that transforms $\mathbf{M}_{\mathcal{S}}$ into the channel dimension D.

Discussion. Our coordinate GCN is built in a coarser spatial grid and its size is determined by the downsampling rate d (we usually set $d = 8$, the effect of changing the downsample rate is analysed in Section 4.2.1). Compared to the Non-local operator [39] that needs to build a large fully-connected graph with adjacency matrix $\mathbf{A} \in \mathbb{R}^{HW \times HW}$, our method is significantly more efficient. Moreover, by re-ordering Eq. 3 to $\mathbf{M}_{\mathcal{S}} = \delta(\mathbf{V}_{\mathcal{S}})(\psi(\mathbf{V}_{\mathcal{S}})^{\top}\upsilon(\mathbf{V}_{\mathcal{S}}))\mathbf{W}_{\mathcal{S}}$ (following the associative rule), we can obtain large savings in terms of memory and computation (from $O((HW)^2)$ to $O(HW)$).

3.3 Graph convolution in feature space

Given that the coordinate space GCN explicitly models the spatial relationships between pixels in the image, we now consider projecting the input feature into the feature space $\mathcal{F}$. The coordinate space GCN enables our network to produce coherent predictions that consider all objects in the image, whilst the feature space GCN models interdependencies along the channel dimensions of the network's feature map. Assuming that filters in later layers of a FCN are responsive to object parts and other high-level features [45], then the feature space GCN captures correlations between more abstract features in the image like object parts.

Feature space projection. In practice, we first reduce the dimension of the input feature $\mathbf{X}$ with function $\theta(\mathbf{X}) \in \mathbb{R}^{N \times D_1}$ and formulate the projection function $\varphi(\mathbf{X}) = \mathbf{H}_{\mathcal{F}}^{\top} \in \mathbb{R}^{N \times D_2}$ as a linear combination of input $\mathbf{X}$ such that the new features can aggregate information from multiple regions.

Formally, the input feature $\mathbf{X}$ is projected to a new feature $\mathbf{V}_{\mathcal{F}}$ in the feature space $\mathcal{F}$ via the projection function $\mathbf{H}_{\mathcal{F}}^{\top}$. Thus we have

$$\mathbf{V}_{\mathcal{F}} = \mathbf{H}_{\mathcal{F}}^{\top}\theta(\mathbf{X}) = \varphi(\mathbf{X})\theta(\mathbf{X}), \tag{4}$$

where both functions of $\theta(\cdot)$ and $\varphi(\cdot)$ are implemented with 1×1 convolutional layer. This results in a new feature $\mathbf{V}_{\mathcal{F}} \in \mathbb{R}^{D_2 \times D_1}$, which consists of D_2 nodes, each of dimension D_1.

Feature graph convolution. After projection, we can build a fully-connected graph with adjacency matrix $\mathbf{A}_{\mathcal{F}} \in \mathbb{R}^{D_2 \times D_2}$ in the feature space $\mathcal{F}$, where each node contains the feature descriptor. Following Eq. 1, we have

$$\mathbf{M}_{\mathcal{F}} = (\mathbf{I} - \mathbf{A}_{\mathcal{F}})\mathbf{V}_{\mathcal{F}}\mathbf{W}_{\mathcal{F}}, \tag{5}$$

where $\mathbf{W}_{\mathcal{F}} \in \mathbb{R}^{D_1 \times D_1}$ denotes the layer-specific trainable edge weights. We consider Laplacian smoothing [8, 21] by updating the adjacency matrix to $(\mathbf{I} - \mathbf{A}_{\mathcal{F}})$ to propagate the node features over the graph. The identity matrix $\mathbf{I}$ serves as a residual *sum* connection in our implementation that alleviates optimisation difficulties. Both adjacency matrix $\mathbf{A}_{\mathcal{F}}$ and $\mathbf{W}_{\mathcal{F}}$ are randomly initialised and optimised by gradient decent during training in an end-to-end fashion.

Reprojection. As in Sec. 3.2, after the reasoning, we map the new features $\mathbf{M}_{\mathcal{F}}$ back into the original coordinate space with output $\tilde{\mathbf{X}}_{\mathcal{F}} \in \mathbb{R}^{N \times D}$ to be compatible with regular convolutional neural networks,

$$\tilde{\mathbf{X}}_{\mathcal{F}} = \phi(\mathbf{H}_{\mathcal{F}}\mathbf{M}_{\mathcal{F}}). \tag{6}$$

This is done by first reusing the projection matrix $\mathbf{H}_{\mathcal{F}}$ and then performing a linear projection (*e.g.* 1×1 convolution layer) to transform $\tilde{\mathbf{M}}_{\mathcal{F}}$ into the original coordinate space. As a result, we have the feature $\tilde{\mathbf{X}}_{\mathcal{F}}$ with feature dimension of D at each grid coordinate.

3.4 DGCNet

The final refined feature is computed as $\tilde{\mathbf{X}} = \mathbf{X} + \tilde{\mathbf{X}}_{\mathcal{S}} + \tilde{\mathbf{X}}_{\mathcal{F}}$, where "+" denotes point-wise summation. To this end, we can easily incorporate our proposed module into existing backbone CNN architectures (*e.g.*, ResNet-101). Figure 2 shows the schematic illustration of our proposed DGCNet .

Implementation of DGCNet. We insert our proposed module between two 3×3 convolution layers (both layers output $D = 512$ channels) appended at the end of a Fully Convolutional Network (FCN) for the task of semantic segmentation. Specifically, we use an ImageNet pretrained ResNet-101 as backbone network, removing the last pooling and FC layer. Our proposed module is then random initialised. Dilated convolution and multi-grid strategies [14] are adopted in last two stages of the backbone. We simply set $D_1 = \frac{D}{2}$ and $D_2 = \frac{D}{4}$ in our implementation. We add a synchronised batch normalisation (BN) layer and ReLU non-linearity after each convolution layer in our proposed module, except for the convolution layers in the coordinate space GCN (there are no BN and ReLU operations in the coordinate space GCN defined in Sec. 3.2).

4 Experiments

To evaluate our proposed method, we carry out comprehensive experiments on the Cityscapes [9] and PASCAL Context [31] datasets, where we achieve state-of-art performance. We describe our experimental setup in Sec. 4.1, before presenting experiments on the Cityscapes dataset (on which we also perform ablation studies) in Sec. 4.2 and finally Pascal Context in Sec. 4.3.

Table 1: Ablation studies on (a) the proposed components of our network and (b) additional training and inference strategies. All methods use the ResNet-101 backbone, and are evaluated using the mean IoU on the Cityscapes validation set. Refer to Sec. 4.2.1 for additional details.

	Backbone	Coord. GCN	Feature GCN	mIoU (%)
Dilated FCN	ResNet-101	✘	✘	75.2
GCN	ResNet-101	✔	✘	78.8
GCN	ResNet-101	✘	✔	79.3
DGCNet	ResNet-101	✔	✔	80.5

(a) Comparison of different graph modules.

	OHEM	Multi-grid	MS	mIoU (%)
DGCNet	✘	✘	✘	79.5
DGCNet	✔	✘	✘	79.8
DGCNet	✔	✔	✘	80.5
DGCNet	✔	✔	✔	81.8

(b) Additional training and inference strategies.

Table 2: Ablation studies on (a) computational cost (input size $[1 \times 512 \times 128 \times 128]$ and (b) graph projection strategy. All methods are evaluated using the mean IoU at a single scale, using the ResNet-101 backbone on the Cityscapes validation set.

	GFLOPs	#Params	mIoU
DA module [12]	24.87	1 496 224	79.8
DGC module (Ours)	14.15	1 240 704	80.5

(a) Comparison on computation cost.

Downsample rate	d=4	d=8	d=16
Avg. pooling	80.2	80.5	80.5
Stride conv.	80.0	80.5	80.5

(b) Ablation on graph projection strategies.

4.1 Experimental setup

Cityscapes: This dataset [9] densely annotates 19 object categories in urban scenes captured from cameras mounted on a car. It contains 5000 finely annotated images, split into 2975, 500 and 1525 for training, validation and testing respectively. The images are all captured at a high resolution of 2048×1024.

PASCAL Context: This dataset [31] provides detailed semantic labels for whole scenes (both "thing" and "stuff" classes [13]), and contains 4998 images for training and 5105 images for validation. Following previous works which we compare to, we evaluate on the most frequent 59 classes with along with one background category (60 classes) [31].

Implementation details: We implement our method using Pytorch. Following [49], we use momentum and adopt a polynomial learning rate decay schedule where the initial learning rate is multiplied by $(1 - \frac{\text{iter}}{\text{total\_iter}})^{0.9}$. The initial learning rate is set to 0.01 for Cityscapes and 0.001 for Pascal Context. Momentum and weight decay coefficients are set to 0.9 and 0.0001 respectively. For data augmentation, we apply random cropping (crop size with 769) and random left-right flipping during training for Cityscapes. For the Pascal Context dataset, our crop size is 480. We also use synchronised batch normalisation for better estimation of the batch statistics.

Metrics: Following the common procedure of [9, 12, 52], we report the mean Intersection over Union (IoU), averaged over all classes.

4.2 Experiments on Cityscapes

4.2.1 Ablation Studies

Effect of proposed modules: As shown in Table 1a, our proposed GCN modules substantially improve performance. Compared to our baseline FCN with dilated convolutions (with the ResNet-101 backbone), appending the Channel-GCN module obtains a mean IoU of 79.3%, which is an improvement of 4.1%. Similarly, the Spatial-GCN module on its own improves the baseline by 3.6%. The best results are obtained by combining the two modules together, resulting in a mean IoU of 80.5%. The effect of our modules are visualised in Fig. 3. The contextual information captured by our graph module improve the consistency of our results

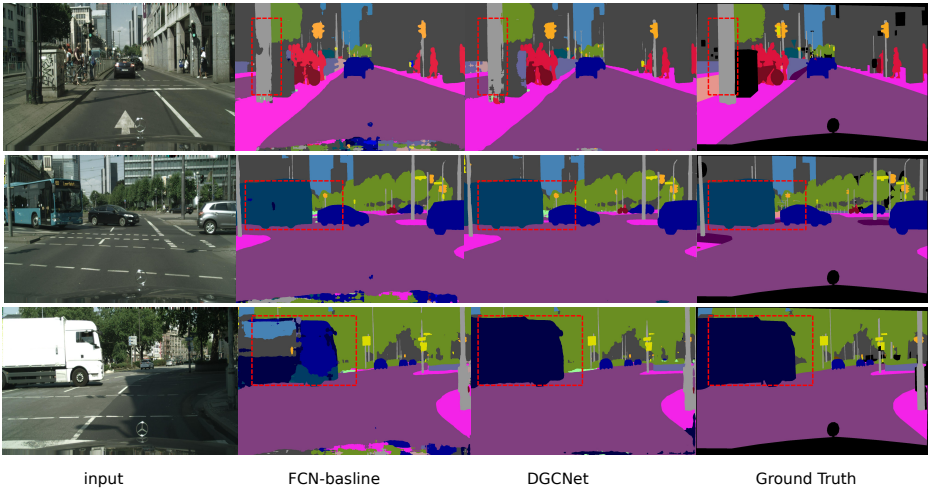

Figure 3: Cityscapes results compared with Dilated FCN ResNet101 baseline [43]. Red boxes show our method can handle inconsistent results within the same object. Best view in color. More results in supplementary material.

leading to fewer artifacts in the prediction.

Effect of additional training and inference strategies: It is common to use additional "tricks" to improve results for semantic segmentation benchmarks [5, 14, 36, 44, 49]. Table 1b shows that our proposed GCN module is complementary to them, and incorporating these strategies improves our results too.

Specifically, we consider 1) Online Hard Example Mining (OHEM) [22, 33, 36, 44] where the loss is only computed on the K pixels with the highest loss in the image. Following [44], we used $K = 10^6$ in a 769×769 cropped training image 2) Multi-Grid [5, 14] employs a series of convolutional filters in parallel using different dilation rates (4, 8 and 16) in the last ResNet block. 3) Multi-Scale (MS) ensembling is commonly used at inference time to improve results [4, 7, 10, 14, 44, 49, 50] We average the segmentation probability maps from 6 image scales{0.75, 1 1.25, 1.5, 1.75, 2} during inference.

As shown in Table. 1b each of these strategies provides consistent improvements to our overall results. Using these strategies, we compare to the state-of-art in the following subsection.

Computational cost analysis Table 2a shows that our proposed method costs significantly fewer floating point operations than the related work of DANet [14], but still achieves higher performance.

Effect of mapping strategies: As mentioned in Sec. 3.2, different mapping strategies are possible to build the coordinate space graph. We present the effect of two strategies – average pooling and strided convolution – for different downsampling ratios, d, in Tab. 2b. It is interesting to observe that the model with average pooling achieves similar performance as strided convolution, even slightly outperforming it for $d = 4$. One reason may be that the model with the parameter-free mapping strategy can overfit less than the one using the strided convolution operation. Note that both strategies are robust to the choice of d, and similar performance is obtained for $d = 4$, 8 and 16. The final model that we use for comparing to the state-of-the-art uses strided convolutions with $d = 8$.

Table 3: State-of-the-art comparison on the Cityscapes test set.

Method	Backbone	mIoU (%)
PSPNet [19]	ResNet-101	78.4
PSANet [50]	ResNet-101	78.6
OCNet [42]	ResNet-101	80.1
DGCNet (Ours) †	ResNet-101	**80.9**
SAC [43]	ResNet-101	78.1
AAF [15]	ResNet-101	79.1
BiSeNet [40]	ResNet-101	78.9
PSANet [50]	ResNet-101	80.1
DFN [41]	ResNet-101	79.3
DepthSeg [18]	ResNet-101	78.2
DenseASPP [40]	ResNet-101	80.6
GloRe [8]	ResNet-101	80.9
DANet [12]	ResNet-101	81.5
OCNet [42]	ResNet-101	81.7
DGCNet (Ours) ‡	ResNet-50	80.8
DGCNet (Ours) ‡	ResNet-101	**82.0**

†: trained only on train-fine set.
‡: trained on train-fine and val-fine sets.

Table 4: Comparison to other methods on Pascal Context [31] dataset.

Method	Backbone	mIoU (%)
FCN8-s [30]	VGG-16	37.8
HO CRF [1]	VGG-16	41.3
Piecewise [26]	VGG-16	43.3
DeepLab-v2 (COCO) [8]	ResNet-101	45.7
RefineNet [27]	ResNet-101	47.3
PSPNet [43]	ResNet-101	47.8
Ding *et al.* [11]	ResNet-101	51.6
EncNet [46] (SS)	ResNet-50	49.0
EncNet [46] (MS)	ResNet-101	51.7
SGR [24]	ResNet-101	52.5
DANet [12]	ResNet-50	50.1
DANet [12]	ResNet-101	52.6
Dilated FCN baseline	ResNet-50	44.3
DGCNet (SS)	ResNet-50	50.1
DGCNet (SS)	ResNet-101	**53.0**
DGCNet (MS)	ResNet-101	**53.7**

SS: Single scale. MS: Multi scale

4.2.2 Comparisons with state-of-the-art

Table 3 compares our approach with existing methods on the Cityscapes test set. Following common practice towards obtaining the highest performance, we use the inference strategies described in the previous section. For fair comparison, Tab. 3 shows methods that are only trained using the fine annotations from Cityscapes, and evaluated on the evaluation server. We achieve a mean IoU of 80.7% when only using the training set, thus outperforming PSANet [50] by 2.1% and OC-Net [42] by 0.8%. Training with both train-fine and val-fine sets achieves an IoU of 82.0%. In both scenarios, we obtain state-of-the-art results. Detailed per-class results are provided in the supplementary material, which shows that our method achieves the highest IoU in 16 out of the 19 classes.

4.3 Experiments on Pascal Context

Table 4 shows our results on Pascal Context. We follow prior work [14, 27, 46] to use the semantic labels of the most frequent 59 object categories plus background (therefore, there are 60 classes in total). The Dilated-FCN baseline achieves a mean IoU of 44.3% with the ResNet-50 backbone. Our proposed DGCNet significantly improves this baseline, achieving an IoU of 50.1% with the same ResNet-50 backbone under single scale evaluation (SS), which outperforms previous work using the same backbone (49.0) [47]. With the ResNet-101 backbone, DGCNet achieves an IoU of 53.0%. Moreover, our performance further improves to 53.7% when multiscale inference (MS) is adopted, surpassing the previous state-of-the-art [14] by a large margin.

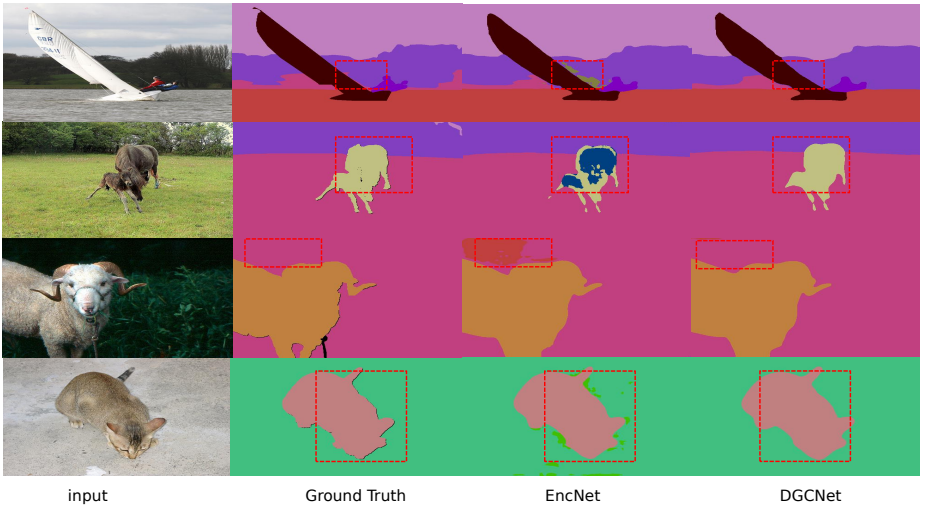

Figure 4: Comparison of our results on Pascal Context to the state-of-art EncNet [47] method. Note that our results are more consistent and have fewer artifacts. Best view in color.

5 Conclusion

We proposed a graph-convolutional module to model the contextual relationships in an image, which is critical for dense prediction tasks such as semantic segmentation. Our method consists of two branches, one to capture context along the spatial dimensions, and another along the channel dimensions, in a convolutional feature map. Our proposed approach provides significant improvements over a strong baseline, and achieves state-of-art results on the Cityscapes and Pascal Context datasets. Future work is to address other dense prediction tasks such as instance segmentation and depth estimation.

Acknowledgments

This work was supported by EPSRC Programme Grant Seebibyte EP/M013774/1, ERC grant ERC-2012-AdG 321162-HELIOS and EPSRC/MURI grant EP/N019474/1. We gratefully acknowledge the use of the University of Oxford Advanced Research Computing (ARC) facility in carrying out this work. We would also like to acknowledge the Royal Academy of Engineering, FiveAI and the support of DeepMotion AI Research for providing the computing resources in carrying out this research.

References

[1] Anurag Arnab, Sadeep Jayasumana, Shuai Zheng, and Philip HS Torr. Higher order conditional random fields in deep neural networks. In *European Conference on Computer Vision*, pages 524–540. Springer, 2016.

[2] Anurag Arnab, Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Måns

Larsson, Alexander Kirillov, Bogdan Savchynskyy, Carsten Rother, Fredrik Kahl, and Philip HS Torr. Conditional random fields meet deep neural networks for semantic segmentation: Combining probabilistic graphical models with deep learning for structured prediction. *IEEE Signal Processing Magazine*, 2018.

[3] Wonmin Byeon, Thomas M. Breuel, Federico Raue, and Marcus Liwicki. Scene labeling with lstm recurrent neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected CRFs. *International Conference on Learning Representations*, 2015.

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

[6] 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*, 2018.

[7] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *European Conference on Computer Vision*, 2018.

[8] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Shuicheng Yan, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. *arXiv preprint arXiv:1811.12814*, 2018.

[9] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[10] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1635–1643, 2015.

[11] Henghui Ding, Xudong Jiang, Bing Shuai, Ai Qun Liu, and Gang Wang. Context contrasted feature and gated multi-scale aggregation for scene segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[12] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 2010.

[13] David A Forsyth, Jitendra Malik, Margaret M Fleck, Hayit Greenspan, Thomas Leung, Serge Belongie, Chad Carson, and Chris Bregler. Finding pictures of objects in large collections of images. In *International workshop on object representation in computer vision*, pages 335–360. Springer, 1996.

[14] Jun Fu, Jing Liu, Haijie Tian, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. *arXiv preprint arXiv:1809.02983*, 2018.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[16] Xuming He, Richard S Zemel, and Miguel Á Carreira-Perpiñán. Multiscale conditional random fields for image labeling. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[17] Tsung-Wei Ke, Jyh-Jing Hwang, Ziwei Liu, and Stella X. Yu. Adaptive affinity fields for semantic segmentation. In *European Conference on Computer Vision*, 2018.

[18] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

[19] Shu Kong and Charless C. Fowlkes. Recurrent scene parsing with perspective understanding in the loop. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

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

[22] Qizhu Li, Anurag Arnab, and Philip HS Torr. Holistic, instance-level human parsing. In *British Machine Vision Conference*, 2017.

[23] Yin Li and Abhinav Gupta. Beyond grids: Learning graph representations for visual recognition. In *Advances in Neural Information Processing Systems*, 2018.

[24] Xiaodan Liang, Zhiting Hu, Hao Zhang, Liang Lin, and Eric P Xing. Symbolic graph reasoning meets convolutions. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, 2018.

[25] Di Lin, Yuanfeng Ji, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Multi-scale context intertwining for semantic segmentation. In *European Conference on Computer Vision*, 2018.

[26] Guosheng Lin, Chunhua Shen, Anton van den Hengel, and Ian D. Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[27] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian D. Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[28] Tsung-Yi Lin, Piotr DollÃąr, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

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

[30] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[31] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[32] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *IEEE International Conference on Computer Vision*, 2015.

[33] Tobias Pohlen, Alexander Hermans, Markus Mathias, and Bastian Leibe. Full-resolution residual networks for semantic segmentation in street scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4151–4160, 2017.

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

[35] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *European Conference on Computer Vision*, 2006.

[36] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[37] Bing Shuai, Zhen Zuo, Bing Wang, and Gang Wang. Scene segmentation with dag-recurrent neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[38] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*, 2015.

[39] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[40] Maoke Yang, Kun Yu, Chi Zhang, Zhiwei Li, and Kuiyuan Yang. Denseaspp for semantic segmentation in street scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[41] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *European Conference on Computer Vision*, 2018.

[42] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Learning a discriminative feature network for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[43] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *International Conference on Learning Representations*, 2016.

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

[45] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, 2014.

[46] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[47] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[48] Rui Zhang, Sheng Tang, Yongdong Zhang, Jintao Li, and Shuicheng Yan. Scale-adaptive convolutions for scene parsing. In *IEEE International Conference on Computer Vision*, 2017.

[49] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[50] 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 *European Conference on Computer Vision*, 2018.

[51] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip H. S. Torr. Conditional random fields as recurrent neural networks. In *IEEE International Conference on Computer Vision*, 2015.

[52] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ADE20K dataset. *arXiv preprint arXiv:1608.05442*, 2016.