Pyramid Graph Networks with Connection Attentions for Region-Based One-Shot Semantic Segmentation

Chi Zhang[1], Guosheng Lin[1]*, Fayao Liu[2], Jiushuang Guo[3], Qingyao Wu [4], Rui Yao[5]
[1]Nanyang Technological University, [2]Institute for Infocomm Research A*STAR, [3]Stanford University,
[4]South China University of Technology, [5]China University of Mining and Technology
E-mail: chi007@e.ntu.edu.sg , gslin@ntu.edu.sg

Abstract

One-shot image segmentation aims to undertake the segmentation task of a novel class with only one training image available. The difficulty lies in that image segmentation has structured data representations, which yields a many-to-many message passing problem. Previous methods often simplify it to a one-to-many problem by squeezing support data to a global descriptor. However, a mixed global representation drops the data structure and information of individual elements. In this paper, We propose to model structured segmentation data with graphs and apply attentive graph reasoning to propagate label information from support data to query data. The graph attention mechanism could establish the element-to-element correspondence across structured data by learning attention weights between connected graph nodes. To capture correspondence at different semantic levels, we further propose a pyramid-like structure that models different sizes of image regions as graph nodes and undertakes graph reasoning at different levels. Experiments on PASCAL VOC 2012 dataset demonstrate that our proposed network significantly outperforms the baseline method and leads to new state-of-the-art performance on 1-shot and 5-shot segmentation benchmarks.

1. Introduction

Fully supervised learning has shown great success in many computer vision tasks, thanks to large-scale datasets [2, 13] and deep neural networks [11, 8]. However, standard supervised learning tasks, such as image classification and semantic segmentation, have their intrinsic limitations that both training and testing have to be applied on a specific task, *i.e.*, the categories to be classified. More recently, meta-learning along with the power of deep neural networks has received growing interests in the machine learning community. Meta-learning, also called learning to learn, aims to train a meta learner that is able to produce good generalization performance on unseen but similar tasks with scarce training data. The training of a meta-learner is not performed on a specific task, and instead, on a distribution of similar tasks.

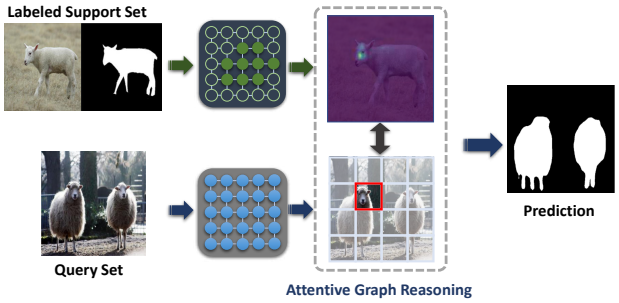

Figure 1: Illustration of the proposed Pyramid Graph Networks for solving one-shot image segmentation. Given only one training image, our model could perform segmentation on new test images. The proposed Graph Attention mechanism could find the correspondence between image regions.

A well-studied area of meta-learning is one-shot image classification which aims to classify a test image given only one training image in each class. One-shot segmentation further extends this task to pixel levels. With only one pixel-wise labeled training image, the goal is to predict a binary mask in the testing images. Unlike one-shot image classification tasks where each data point has a label, in segmentation, the data is represented with a group of structured pixels. This yields a many-to-many message passing problem that we do not know which pixels in two images are related.

To solve the problems mentioned above, previous methods transform it to a one-to-many problem, using global pooling operations to squeeze structured support data to a global descriptor and exploit it to guide dense predictions [18, 3, 15, 26, 27]. In this case, the prediction of each

*Corresponding author: G. Lin.

pixel location references the same guidance vector. For example, in [3, 27, 26], the network structure encourages the foreground pixels in the query image to have a close distance to the global descriptor, via cosine distance metric or a learnable distance metric. However, two grouped pixels may only show connections in a small fraction of the whole representations. It is common that a part of the objects shown in the support set may not appear in the query set. In this case, the global average descriptor may introduce noise, as we only want the shared components to provide guidance information.

In this work, we argue that the query set and the support set can be modeled with graphs and the process of propagating label information from a structured representation to another one can be viewed from the graph reasoning standpoint. We combine the query graph and support graph into a big bipartite graph that we assume each node in the query graph is fully connected with the nodes in the support graph. To reason the underlying correlations between the unlabeled query nodes and labeled support nodes, we apply graph attention mechanism [23] to weight the connections of each neighbouring node. With the attention weights, each query node could selectively aggregate label information from the support graph and thus help the downstream node classification task. This could also provide some interpretability by inspecting the attention distribution. An illustration of our attention mechanism is shown in Fig. 1.

The next question is how we should model the images with graph representations. Given two locations in the query image and the support image, they may show connections at different levels. For example, the connection of two human images can be an elementary feature, *e.g.*, eyes, which corresponds to a small region, but it can also be a more abstract feature, *e.g.*, faces, constituted by a group of elementary features. With the intuition above, we propose a pyramid-like branched graph reasoning structure to establish such multi-level connections. The graph nodes in different branches capture features of sub-regions at different scales. At the bottom level, each pixel in the feature map is modeled as the node in the graph. The rest branches model different sizes of sub-regions in the image as the graph nodes via adaptive pooling. Adaptive pooling could downsample arbitrary sized feature maps to a set of fixed-sized grids. Each of the branches performs graph reasoning independently, and their results are projected back to the original size. Finally, we fuse all branches for the final prediction. The pyramid structure enables graph reasoning at different scales to help the node classification.

In our experiment, We also explore various model variants that adopt other techniques for capturing multi-level relations, *e.g.*, dilated convolutions. We empirically demonstrate that our graph-based method achieves better results than the baseline methods based solely on global guidance.

Our contributions are summarized as follows:

- We propose Pyramid Graph Networks (PGNet) for one-shot image segmentation which establishes correspondence between object parts by attention mechanism. Compared with previous work based on global information, the proposed graphical model allows each pixel prediction to reference more related area in the support set.
- We propose a pyramid architecture that constructs multi-level region-based connections between two images to effectively propagate label information from the support set to the query set.
- The proposed attention mechanism also provides interpretability of the relations by visualizing the node connection weights.
- Experiments on PASCAL VOC 2012 dataset show that our method significantly outperforms the baseline model and achieves new state-of-the-art 1-shot and 5-shot segmentation results.

2. Related Work

Meta Learning. Meta-learning aims to learn a model which can be quickly adapted to new tasks with a small amount of training data, which is different from standard supervised learning that both training and testing are on a specific task. A representative study is to apply meta-learning on few-shot image classification. Previously, a line of few-shot classification literature adopts metric learning to compute the pair-wise distance between samples which can be used for classification [20, 25, 10]. In this case, the meta-learner owns the ability to make comparison between samples. Other formulations, such as learning good initial weights for fast adaptation [5, 21], updating parameters by LSTM [16], adopting temporal convolutions [14], *etc*. also yield promising results.

Few-shot Segmentation. Semantic segmentation is a fundamental computer vision task, which aims to classify each pixel in the images to a set of classes. Few-shot segmentation extends few-shot classification to pixel levels. Previous works follow a two-branch network design [18, 15, 27, 26, 3, 9]. The support branch extracts information from labeled support data to guide segmentation in the query branch. In [18], the query branch is an FCN and the support branch directly predicts a vector as the parameters of a layer in the query branch to guide segmentation. In [15, 26], the generated global vector from the support branch is upsampled and concatenated with the query feature for dense predictions. Zhang *et al*. [27] re-weight the query feature map by calculating a similarity score between the global vector and each query position to segment the target category.

The intrinsic spirits behind the above methods are to simplify the many-to-many correspondence problem to a one-to-many problem. To generate the global descriptor, early works [18, 15, 3] hide the background in the support RGB image and perform global average pooling at the end of the support branch to generate a global descriptor. Works in [26, 27] improve it by performing average pooling over the foreground mask regions in intermediate features to extract the global vector. Our method, on the other hand, establishes element-to-element correspondence that each location in the query image could selectively extract useful information from the support set.

Attention. Incorporating attention mechanism to effectively handle graph-like data has become popular in recent years [17, 12, 23, 22, 4]. Attention could allow the task to focus on the most relevant parts to help make decisions. Among them, our work is most related to Graph Attention Networks [23], which adopts attention mechanism to perform node classification in graph data. In the Graph Attention Networks, each node representation attends over all neighbouring nodes and the weights between nodes are implicitly specified with the attention mechanism. Then, the node is updated with a weighted sum of neighbouring nodes. Graph attention mechanisms are also used in other computer vision tasks, such as few-shot image classification [6] and social relationship understanding [24]. The key difference in our work is that we implicitly model image regions as the graph nodes and establish cross-image relationships.

3. Problem Set-up

Before describing our network in detail, we first introduce the notations and the formulation of a typical meta-learning task. The task of meta-learning aims to train a model $\mathcal{R}$ to undertake an unseen task $\mathcal{T}_i$, with only a few labeled examples. In the one-shot segmentation setting, a task $\mathcal{T}_i$ denotes binary segmentation of a specific category. The model is trained by sampling tasks from $\mathcal{P}_{train}(\mathcal{T})$ and is evaluated on new tasks $\mathcal{P}_{test}(\mathcal{T})$. To avoid confusion, at both training and testing time, the labeled example images is called the support set and the images for prediction are called the query set.

The model is trained by episodically sampling tasks from $\mathcal{P}_{train}(\mathcal{T})$, which is aligned with the evaluation process. The construction process of a 1-shot learning episode is provided in Algorithm 1. At training time, the model parameters are updated by optimizing the loss $\mathcal{L}_{train}$.

4. Method

In this section, we present our PGNet with the motivation of establishing the element-to-element correspondence between structured data. The network is constructed with a few primary building blocks, *i.e*. the Graph Attention Unit. In what follows, we begin with the illustration of our Graph Attention Unit. Then we introduce our pyramid graph reasoning module that models different sizes of sub-regions as the graph nodes. The overview of our network is shown in Fig. 2.

Algorithm 1 Construction of a 1-shot learning episode. $x \in \mathbb{R}^{H\times W\times 3}$ denotes an RGB image and $y \in \mathbb{R}^{H\times W}$ denotes a binary mask. $phase \in \{train, test\}$

Input: Model $\mathcal{R}$, task distribution $\mathcal{P}_{phase}(\mathcal{T})$, loss function $\mathcal{L}(\cdot)$

while not done **do**
  $\mathcal{T} \leftarrow$ Sample a task $\mathcal{T} \sim \mathcal{P}_{phase}(\mathcal{T})$
  $\mathcal{S} \leftarrow$ Sample a labeled image $(x_s, y_s(\mathcal{T}))$ as the support set
  $\{x_q, y_q(\mathcal{T})\} \leftarrow$ Sample the query set.
  $\hat{y}_q(\mathcal{T}) \leftarrow \mathcal{R}(x_q, \mathcal{S})$, make predictions
  $\mathcal{L}_{phase} \leftarrow \mathcal{L}(y_q(\mathcal{T}), \hat{y}_q(\mathcal{T}))$, compute loss
  done
end while

4.1. Graph Attention Unit

Given a query image x_q and a labeled support image x_s, we first employ a shared convolutional neural network to convert them into feature maps where each pixel location is represented by a vector. A feature map can be modeled with a graph that each node corresponds to a feature vector in the original feature maps. Our goal is to establish relations between two graph domains to propagate label information from the support images to unlabeled query images for classification.

Our Graph Attention Unit (GAU) takes inspiration from Graph Attention Network [23], where the model learns a weight coefficient for each neighbour and reconstructs each node as the weighted sum of neighbouring nodes for the downstream graph mining tasks. Our Graph Attention Unit shares the same spirit and combines the query graph and the support graph into a big bipartite graph that each query node is fully connected to all support nodes. After that, we reconstruct each query node by fusing all neighbouring support nodes with attention mechanism. Then fused nodes, along with their original node representations, are used for node classification. The architecture of the Graph Attention Unit is shown in Fig. 3. We first elaborate the node updating process with one example query node and such operations can be applied to all query nodes in parallel.

Assume we have a query node representation $\vec{h}_q$ and all support node representations $\{\vec{h}_s^1, \vec{h}_s^2, ..., \vec{h}_s^N\}$, where $\vec{h} \in \mathbb{R}^C$, N is number of all nodes in the support graph and C is the feature dimension, the Graph Attention Unit updates the query node by selectively accumulating information from

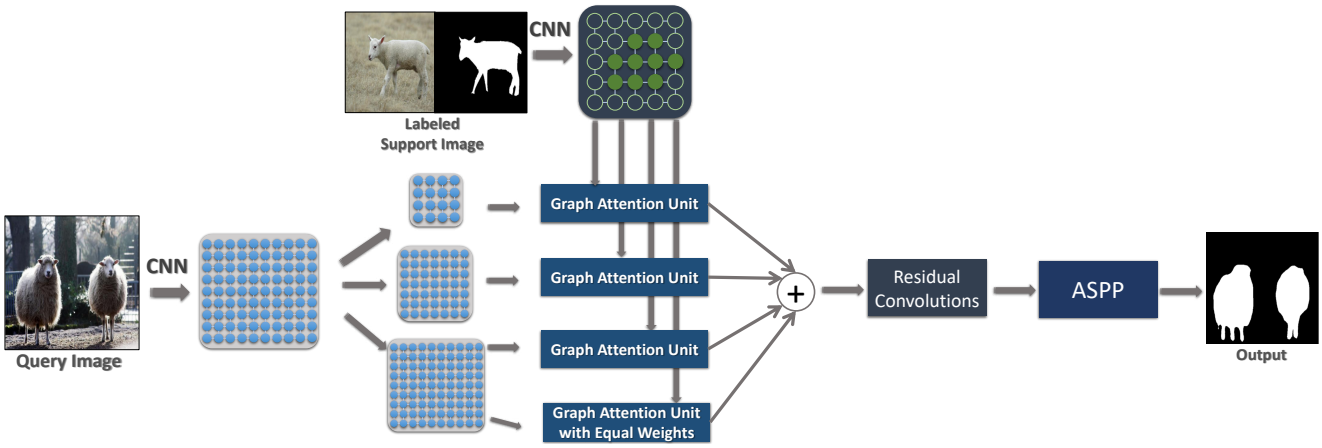

Figure 2: Illustration of our network on 1-shot image segmentation task. Given a sampled image pair, we first use a shared CNN to extract their features, then we model the query feature maps and the foreground region in support feature maps with graphs. After that, a set of paralleled adaptive pooling layers are applied to the query features to acquire different sub-region representations, which are sent to different Graph Attention Units for graph reasoning. Finally, the outputs of different branches are fused by addition and further processed by the rest convolutions. The GAU branch with equal attention weights has the same effect with global pooling.

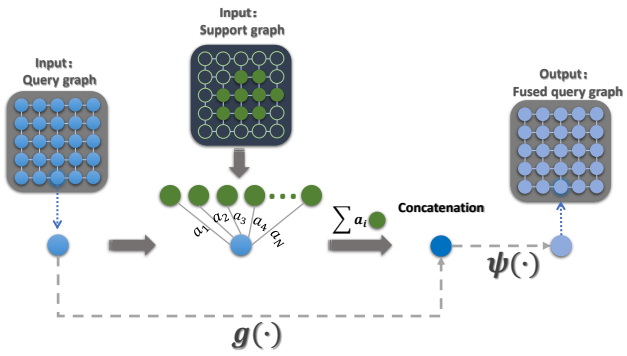

Figure 3: Illustration of our proposed Graph Attention Unit with one example node in the query graph.

all support nodes. To that end, we need to find a pairwise function $f(\cdot)$ that generates a scalar e_j as the correlation score between the query node $\vec{h}_q$ and a support node $\vec{h}_s^j$

$$e_j = f(\theta(\vec{h}_q), \phi(\vec{h}_s^j)) \tag{1}$$

where ϕ and θ are linear transformation functions, which project the node features to a new space. Here, we experiment two choices of function $f(\cdot)$ that are commonly used for vector comparison.

Linear Transformation. The original formulation in Graph Attention Network [23] for computing the correlation score is to concatenate two vectors and apply linear transformation with the weight vector $\vec{w}_f$:

$$f(\phi(\vec{h}_q), \theta(\vec{h}_s^j)) = \vec{w}_f^T(\phi(\vec{h}_q)||\theta(\vec{h}_s^j)) \tag{2}$$

where $||$ denotes the concatenation operation.

Inner product. We also attempt using the dot products of two vectors to cumpute the scalar:

$$f(\phi(\vec{h}_q), \theta(\vec{h}_s^j)) = \phi(\vec{h}_q)^T\theta(\vec{h}_s^j) \tag{3}$$

Once we obtain the correlation factor e_j of all neighbouring support nodes, we normalize them with the softmax function and generate the weights a_j. Based on that, we fuse all support node representations with the weighted sum. Finally, the fused node is concatenated with the original input query node vector, and they are fused with another linear transformation function:

$$a_j = \frac{\exp(e_j)}{\sum_{k=1}^{N}\exp(e_k)}, \tag{4}$$

$$\vec{v}_q = \sum_{j=1}^{N} a_j g(\vec{h}_s^j)), \tag{5}$$

$$\vec{h'}_q = \psi(\vec{v}_q||g(\vec{h}_q)) \tag{6}$$

where $g(\cdot)$ and $\psi(\cdot)$ are linear projection functions, followed by ReLU.

Parallel Computation. At implement time, all linear transformation operations, *i.e.* $g(\cdot)$, $\psi(\cdot)$, $\theta(\cdot)$ and $\phi(\cdot)$ can be operated on all nodes concurrently with 1×1 convolutions due to its grid arrangement property. For the pair-wise operation $f(\cdot)$, as we only model the foreground region in the support image as the support graph, we could apply $f(\cdot)$ to all positions in the support set in the first place and then mask the attention values that correspond to the background with $-\infty$ before softmax normalization. When dot product

is used in $f(\cdot)$, we could use matrix multiplication to compute attention value in parallel, and when linear transformation is used, we can apply tensor broadcasting and 1×1 convolution in the channel dimension for efficient computation.

4.2. Pyramid Graph Reasoning

We have presented the GAU by modeling each pixel as the node vector to perform graph reasoning. However, such connections built on pixel-wise elementary features may not be sufficient to discover the ideal relations. For example, an eye-like feature may falsely establish relations between humans' eyes and dogs' eyes. We may want an abstract feature, *e.g.*, faces being modeled into the graph for better reasoning.

We find that a complex object can be decomposed to a set of basic elements. For instance, faces can roughly be composed of eyes, noses, and mouths. To acquire an unseen high-level representation, we could simply mix the elementary representations. With this intuition, we propose a multi-level graph reasoning scheme that models different sizes of sub-regions as the graph nodes to undertake graph reasoning at different levels. We have demonstrated the bottom level, where each pixel is modeled as the node in the graph. The other two branches in the pyramid structure have similar structures but adopt different sizes of adaptive pooling to the query feature maps before sending them to the GAU. Then, the fused graphs of each Graph Attention Unit are upsampled back to the original feature size with bilinear interpolation, as shown in Fig. 2. Adaptive pooling can extract representations of assigned size given an arbitrary sized feature map. All pixel locations inside a sub-region together constitute a feature representation which is then modeled as a graph node.

We additionally add one GAU that sets equal attention values e_j. In this case, each query node is fused with the average support nodes, which aims to incorporate the global statistics from the support set. The output maps of different branches are fused by addition and processed by three residual convolution blocks [8]. Finally, we add the Atrous Spatial Pyramid Pooling Module (ASPP) [1] at the end to generate the final results.

5. Experiment

5.1. Implementation details.

Our network structure is modified from DeepLab V3 with ResNet-50 as the backbone. Specifically, we decompose the original DeepLab V3 network to a fully convolutional ResNet part and an Atrous Spatial Pyramid Pooling(ASPP) module as the post-processing part. The Resnet backbone, pre-trained on ImageNet, is used as the feature extractor in our network and the ASPP is added at the end. We remove layers in $block$-4 and concatenate features of $block$-3 and $block$-2 as the extracted features, which are sent to different branches. All the convolutional operations inside the GAU, residual blocks and ASPP generate features of 256 channels. The linear projection functions $g(\cdot)$ in different branches share the same parameter. Finally, the model outputs two-channel masks as the predicted foreground and background scores of each location.

At training time, we optimize the network parameter by minimizing the two-class Cross-Entropy loss over all pixel locations with momentum SGD. The network is trained for 600,000 iterations with the learning rate of 0.0025. We adopt random crop, random scale and random horizontal flip on the support images during training for data augmentation.

5.2. Dataset and Evaluation Metric

We evaluate the performance of our algorithm on the PASCAL VOC 2012 dataset with extended annotations in [7]. We follow the dataset division in [18] that 20 object classes in their official resealed order are evenly divided into 4 folds and report cross-validation results. Namely, 15 object categories are used as training tasks with the rest as the testing tasks. At test time, we random sample 1,000 tasks in each test fold. For more details about the dataset, please refer to [18].

We align the evaluation metric with previous works. Given predicted masks from 1000 test episodes, we first calculate a standard foreground Intersection over Union (IoU) score for each object class, then we average the class-wise IoU of all 5 classes as the mean IoU for this fold. When compared with the state-of-the-art results, we report the mean IoU in each test fold and the mean results over 4 folds.

5.3. Ablation Analysis

We conduct extensive ablation experiments to validate our network design. Each analysis experiment is performed twice with fold 0 and fold 1 as the test fold, respectively. We report the average performance of the two test folds for all the experiments in this section.

Backbone network. We first evaluate the importance of the backbone model in our architecture. We experiment with two backbone models that are used in previous work: VGG-16 and ResNet-50. We fix the pair-wise function $f(\cdot)$ with dot product. The results of our network with the two backbone models are presented in Table.1. As is shown, our model with ResNet-50 yields slightly better performance than the VGG version. We adopt ResNet-50 as the backbone model in all following experiments.

The Pair-wise Function. Table. 2 compares our model variants with dot-product and linear transformation as the pair-wise function $f(\cdot)$, which are denoted with **PGNet-Dot** and **PGNet-Linear**, respectively. As is shown, our model

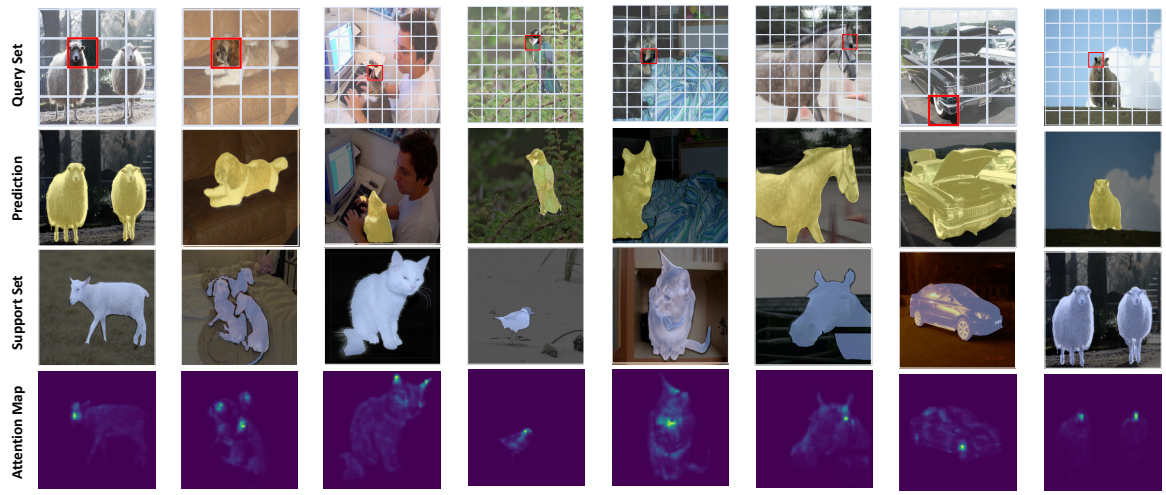

Figure 4: Qualitative results of our network. The first row is the query images. The second row is our network predictions of the query images. The third row is support images with ground-truth annotations. The fourth row is the corresponding attention maps of selected regions (marked with red rectangles in the query images). The grids in the query image indicate which branch we extract the attention maps from. The attention values are normalized to [0,1] to highlight the salient region in the support images.

Model	Backbone	Mean IoU (%)
PGNet-Dot	VGG-16	57.2
PGNet-Dot	ResNet-50	59.4

Table 1: Our model with different backbone networks. PGNet-Dot denotes our model with dot-product as the pair-wise function $f(\cdot)$. Our network with ResNet-50 backbone achieves better results.

Model	Mean IoU (%)
PGNet-Linear	58.1
PGNet-Dot	59.4

Table 2: Our network with dot-product and lienar transformation as the pair-wise function $f(\cdot)$. The dot-product version achieves higher mean IoU socre.

with dot-product as the function $f(\cdot)$ yields better results.

Adaptive Pooling *vs.* Dilated Convolution. Dilated convolution [1] is another common operation to extract information from larger ranges without introducing extra parameters. The receptive field of a filter can be explicitly controlled by varying the dilation rates of convolutional kernels. We experiment a series of model variant by replacing the adaptive pooling operations with dilated convolutions. To build a dilated-convolution version of our network, we make the following modifications to the original structure: 1) We remove the adaptive pooling operations to the input query feature maps such that all GAUs have the same query input. 2) The query node encoding function $\theta(\cdot)$ in the Graph Attention Unit is replaced with 3×3 dilated convolutions to incorporate information from different ranges. With the aim to capture information from different sized regions, the dilation rates in different Graph Attention Unit are set with 2, 4 and 8 separately, while remaining other network components still.

We also experiment adopting these operations on the support feature maps. Specifically, if adaptive pooling is employed in our network, the query feature maps to different GAUs are kept same and apply adaptive pooling operations on the support feature maps, and if dilated convolutions are used, we simply move the changes mentioned above to the support node encoding function $\phi(\cdot)$. The comparison of adaptive max pooling and adaptive average pooling is also investigated in this part. Our baseline method in this experiment is a branch-ensemble model that neither dilated convolution nor adaptive pooling is used, such that all the branches share the same structure and are all applied on the original query and support feature maps. The comparison of the model variants is shown in Table.3. We could find from the result that both the dilated convolution and the adaptive pooling could introduce multi-range information and boost the performance over the baseline result. The optimal result is achieved when adaptive average pooling is employed on query feature maps.

Compared with Global Guidance. As discussed earlier, previous works transform the many-to-many problem to a one-to-many problem that a global description vector from the support set guides the pixel-wise predictions

Method	Operated On	Mean IoU (%)
Baseline-Ensemble	-	57.5
Dilated Conv	Support	58.1
Dilated Conv	Query	57.7
Adaptive Max Pool	Support	58.6
Adaptive Max Pool	Query	57.6
Adaptive Avg Pool	Support	57.6
Adaptive Avg Pool	Query	**59.4**

Table 3: Comparison of using dilated convolution or adaptive pooling to undertake multi-range graph reasoning. Both two methods could boost performance over the baseline model. Adopting adaptive average pooling on the query feature map yields the best result.

in the query image. Here, we implement several baseline models that adopt the designs in previous works. Our first baseline model, **PGNet-Mask-RGB**, adopts the solutions in [18, 15, 3] that we mask the background region in the support images with zero and perform global average pooling over the support features to generate a global vector. Then, this vector is upsampled to the same spatial size of the query feature maps and they are fused by concatenation. The second baseline model, shares the same spirit with minor differences. In [27, 26] they maintain the original RGB support image as the input and extract the global vector by averaging the support features over the foreground mask region. We denote this baseline method with **PGNet-Mask-Feature**. We replace our pyramid graph reasoning module with the two solutions above while maintaining other network components still to validate our design. The above methods can be seen as a special case of our GAU when the attention values are set equal, such that each query node is fused with an averaging support node.

Moreover, we also experiment branch ensembles of the baseline methods to investigate whether our network performance is driven by introducing more parameters. Similarly, We also construct 4 parallel branches, all of which have identical structures with different parameters, and their results are fused by addition, as done in our pyramid structure. The results are shown in Table.4. As is shown, although branch ensembles of baseline methods could slightly boost the baseline performance, our network still outperforms all the baseline methods with a large margin. The attentive graph reasoning method turns out to be more effective in extracting guidance information from the support set than methods solely based on the global vector.

Multi-scale Input Test. As is commonly done in the segmentation literature, we test our network performance with multi-scale query and support image inputs and average their predictions. Specifically, the images are rescaled with the ratio of [0.7, 1, 1.3] and their corresponding predictions are rescaled back to the original size with bilinear interpolation. The effect of multi-scale input test is shown in Table. 5.

Model	Ensemble	Mean IoU (%)
PGNet-Mask-RGB		54.7
PGNet-Mask-RGB	✓	54.8
PGNet-Mask-Feature		56.7
PGNet-Mask-Feature	✓	57.1
PGNet-Dot		**59.4**

Table 4: Comparison of our graph-based network with model variants based on global vector guidance. Our proposed method achieves better results than the baseline methods and their ensemble versions.

Model	Query	Support	Mean IoU (%)
PGNet-Dot			59.4
PGNet-Dot	✓		61.2
PGNet-Dot		✓	59.5
PGNet-Dot	✓	✓	**61.5**

Table 5: Influence of multi-scale input test. All predictions are rescaled to the original image size and fused by average.

5.4. Qualitative Results

Fig.4 shows some qualitative results of our models. We can see that our network could accurately make predictions of the query images with only one labeled training image. Given sub-regions of different sizes in the query images, we visualize the attention weights with respect to all support locations. Our graph attention mechanism could find the most related areas in the support image to help prediction.

5.5. Comparison with the State-of-the-art Results

We compare our final model with the state-of-the-art methods on PASCAL VOC 2012 dataset. We report our results under two different experimental set-ups where the difference is the evaluation metric adopted: The first evaluation metric is the one we have explained at the beginning of this section. The second evaluation metric is the one proposed in [15]. They ignore the object classes and report the mean of foreground IoU and background IoU over all test images in the fold. We denote them with Mean IoU and IoU, respectively, to differentiate. The 1-shot results under two evaluation metrics are shown in Table. 6 and Table. 7, respectively. We can see from the tables that under both experiment set-ups, our network outperforms previous methods and achieves a new state-of-the-art performance.

5-shot Experiments. As the proposed Graph Attention Unit dynamically generates the weights between the query nodes and all support nodes, we can extend our model to

Model	fold-0	fold-1	fold-2	fold-3	mean
Reduced-DFCN8s [19]	39.2	48.0	39.3	34.2	40.2
OSLSM [18]	33.6	55.3	40.9	33.5	40.8
co-FCN [15]	36.7	50.6	44.9	32.4	41.1
SG-One [27]	40.2	58.4	48.4	38.4	46.3
CANet [26]	52.5	65.9	**51.3**	**51.9**	55.4
Ours	**56.0**	**66.9**	50.6	50.4	**56.0**

Table 6: Comparison with the state-of-the-art 1-shot segmentation performance on PASCAL VOC 2012 dataset.

Model	IoU
co-FCN [15]	60.1
Reduced-DFCN8s [19]	60.9
PL [3]	61.2
A-MCG-Conv-LSTM [9]	61.2
OSLSM [18]	61.3
SG-One [27]	63.1
CANet [26]	66.2
Ours	**69.9**

Table 7: Comparison with the state-of-the-art 1-shot segmentation results on PASCAL VOC 2012 dataset, regarding the evaluation metric proposed in [15]

solve the k-shot learning task easily by modeling all support images into the support graph. Specifically, the foreground regions from different support images together constitute the support graph so that the attention distribution is over all foreground locations in the support set. Thus, our algorithm is able to handle 1-shot and k-shot cases with the same model. In comparison, most previous works can only handle one-shot cases and adopt naive fusion methods to fuse individual 1-shot results. For example, [15, 27, 3] average the global descriptors generated by different support images. Zhang *et al*. [26] trains an additional branch to weight the k global descriptors before averaging them. Shaban *et al*. [18] adopts logic OR operation to fuse individual predicted binary masks.

We report the 5-shot segmentation results for a fair comparison with previous works. The results are shown in Table. 8. **PGNet-Fusion** denotes the baseline method that we use the one-shot model to make predictions with each support image and average the 5 predicted masks. **PGNet-Graph** denotes the proposed method that models all support images into the graph. **PGNet-Graph-MS** denotes our proposed model with multi-scale input test operated on the query images. We can see from the results that our graph-based 5-shot learning method is more effective than naive fusion solutions and our final 5-shot result significantly outperforms the state-of-the-art performance under two evaluation metrics.

Model (5-shot)	fold-0	fold-1	fold-2	fold-3	mean
co-FCN [15]	37.5	50.0	44.1	33.9	41.4
OSLSM [18]	35.9	58.1	42.7	39.1	43.9
Reduced-DFCN8s [19]	45.3	51.4	44.9	39.5	45.3
SG-One [27]	41.9	58.6	48.6	39.4	47.1
CANet [26]	55.5	67.8	51.9	53.2	57.1
PGNet-Fusion	53.6	65.6	49.9	49.9	54.8
PGNet-Graph	54.9	67.4	51.8	53.0	56.8
PGNet-Graph-MS	**57.7**	**68.7**	**52.9**	**54.6**	**58.5**

(a) Evaluation metric of mean class-wise IoU adopted in [18]

Model (5-shot)	IoU
co-FCN [15]	60.2
OSLSM [18]	61.5
A-MCG-Conv-LSTM [9]	62.2
PL [3]	62.3
SG-One [27]	65.9
Reduced-DFCN8s [19]	66.0
CANet [26]	69.6
PGNet-Graph-MS	**70.5**

(b) Evaluation metric of IoU adopted in [15]

Table 8: Comparison with the state-of-the-art 5-shot segmentation performance on PASCAL VOC 2012 dataset. Our network outperforms previous methods under both evaluation metrics.

6. Conclusions

We have presented Pyramid Graph Networks for one-shot image segmentation. Compared with previous methods solely based on a global supporting vector, our attentive graphical models establish connections between elements across structure represented data that allow each unlabeled pixel to selectively aggregate guidance information from support image regions. Our pyramid structure models various sizes of regions as the graph nodes to enable graph reasoning at different scale and semantic levels. Experiments on PASVAL VOC 2012 datasets under two evaluation metrics show that our proposed method significantly outperforms the baseline methods and achieves a new state-of-the-art performance.

Acknowledgements

This work is supported by the National Research Foundation Singapore under its AI Singapore Programme [AISG-RP-2018-003] and the MOE Tier-1 research grant [RG126/17 (S)]. We would like to thank NVIDIA for GPU donation.

References

[1] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolu-

tion, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018. 5, 6

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 1

[3] Nanqing Dong and Eric Xing. Few-shot semantic segmentation with prototype learning. In *BMVC*, 2018. 1, 2, 3, 7, 8

[4] Jun Feng, Minlie Huang, Yang Yang, et al. Gake: graph aware knowledge embedding. In *International Conference on Computational Linguistics*, pages 641–651, 2016. 3

[5] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, pages 1126–1135. JMLR. org, 2017. 2

[6] Victor Garcia and Joan Bruna. Few-shot learning with graph neural networks. *arXiv preprint arXiv:1711.04043*, 2017. 3

[7] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *ECCV*, pages 297–312. Springer, 2014. 5

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1, 5

[9] Tao Hu, Pengwan Yang, Chiliang Zhang, Gang Yu, Yadong Mu, and Cees GM Snoek. Attention-based multi-context guiding for few-shot semantic segmentation. In *AAAI*, 2019. 2, 8

[10] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015. 2

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 1

[12] John Boaz Lee, Ryan A Rossi, Sungchul Kim, Nesreen K Ahmed, and Eunyee Koh. Attention models in graphs: A survey. *arXiv preprint arXiv:1807.07984*, 2018. 3

[13] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014. 1

[14] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. *arXiv preprint arXiv:1707.03141*, 2017. 2

[15] Kate Rakelly, Evan Shelhamer, Trevor Darrell, Alyosha Efros, and Sergey Levine. Conditional networks for few-shot semantic segmentation. In *ICLR Workshop*, 2018. 1, 2, 3, 7, 8

[16] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017. 2

[17] Seongok Ryu, Jaechang Lim, and Woo Youn Kim. Deeply learning molecular structure-property relationships using graph attention neural network. *arXiv preprint arXiv:1805.10988*, 2018. 3

[18] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. In *BMVC*, 2017. 1, 2, 3, 5, 7, 8

[19] Mennatullah Siam and Boris Oreshkin. Adaptive masked weight imprinting for few-shot segmentation. *arXiv preprint arXiv:1902.11123*, 2019. 8

[20] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NIPS*, 2017. 2

[21] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *CVPR*, 2019. 2

[22] Kiran K Thekumparampil, Chong Wang, Sewoong Oh, and Li-Jia Li. Attention-based graph neural network for semi-supervised learning. *arXiv preprint arXiv:1803.03735*, 2018. 3

[23] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. 2, 3, 4

[24] Zhouxia Wang, Tianshui Chen, Jimmy Ren, Weihao Yu, Hui Cheng, and Liang Lin. Deep reasoning with knowledge graph for social relationship understanding. *arXiv preprint arXiv:1807.00504*, 2018. 3

[25] Flood Sung Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018. 2

[26] Chi Zhang, Guosheng Lin, Fayao Liu, Rui Yao, and Chunhua Shen. Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In *CVPR*, 2019. 1, 2, 3, 7, 8

[27] Xiaolin Zhang, Yunchao Wei, Yi Yang, and Thomas Huang. Sg-one: Similarity guidance network for one-shot semantic segmentation. *arXiv preprint arXiv:1810.09091*, 2018. 1, 2, 3, 7, 8