Dynamic Graph Message Passing Networks

Li Zhang[1]   Dan Xu[1]   Anurag Arnab[2*]   Philip H.S. Torr[1]

[1]University of Oxford   [2]Google Research

{lz, danxu, phst}@robots.ox.ac.uk   aarnab@google.com

Abstract

Modelling long-range dependencies is critical for scene understanding tasks in computer vision. Although CNNs have excelled in many vision tasks, they are still limited in capturing long-range structured relationships as they typically consist of layers of local kernels. A fully-connected graph is beneficial for such modelling, however, its computational overhead is prohibitive. We propose a dynamic graph message passing network, that significantly reduces the computational complexity compared to related works modelling a fully-connected graph. This is achieved by adaptively sampling nodes in the graph, conditioned on the input, for message passing. Based on the sampled nodes, we dynamically predict node-dependent filter weights and the affinity matrix for propagating information between them. Using this model, we show significant improvements with respect to strong, state-of-the-art baselines on three different tasks and backbone architectures. Our approach also outperforms fully-connected graphs while using substantially fewer floating-point operations and parameters.

1. Introduction

Capturing long-range dependencies is crucial for complex scene understanding tasks such as semantic segmentation, instance segmentation and object detection. Although convolutional neural networks (CNNs) have excelled in a wide range of scene understanding tasks [20, 30, 15], they are still limited by their ability to capture these long-range interactions. To improve the capability of CNNs in this regard, a recent, popular model Non-local networks [34] proposes a generalisation of the attention model of [31] and achieves significant advance in several computer vision tasks.

Non-local networks essentially model pairwise structured relationships among all feature elements in a feature map to produce the attention weights which are used for feature aggregation. Considering each feature element as a node in a graph, Non-local networks effectively model a fully-connected feature graph and thus have a quadratic inference complexity with respect to the number of the feature elements. This is infeasible for dense prediction tasks on high-resolution imagery, as commonly encountered in semantic segmentation [9]. Moreover, in dense prediction tasks, capturing relations between all pairs of pixels is usually unnecessary due to the redundant information contained within the image (Fig. 1). Simply subsampling the feature map to reduce the memory requirements is also suboptimal, as such naïve subsampling would result in smaller objects in the image not being represented adequately.

Graph convolution networks (GCNs) [19, 12] – which propagate information along graph-structured input data – can alleviate the computational issues of non-local networks to a certain extent. However, this stands only if local neighbourhoods are considered for each node. Employing such local-connected graphs means that the long-range contextual information needed for complex vision tasks such as segmentation and detection [29, 28, 3] will only be partially captured. Along this direction, GraphSAGE [13] introduced an efficient graph learning model based on graph sampling. However, the proposed sampling method considered a uniform sampling strategy along the spatial dimension of the input, and was independent of the actual input. Consequently, the modelling capacity was restricted as it assumed a static input graph where the neighbours for each node were fixed and filter weights were shared among all nodes.

To address the aforementioned shortcomings, we propose a novel dynamic graph message passing network (DGMN) model, targeting effective and efficient deep representational learning with joint modeling of two key dynamic properties as illustrated in Fig. 1. Our contribution is twofold: (i) We dynamically sample the neighbourhood of a node from the feature graph, conditioned on the node features. Intuitively, this learned sampling allows the network to efficiently gather long-range context by only selecting a subset of the most relevant nodes in the graph; (ii) Based on the nodes that have been sampled, we further dynamically predict node-dependant, and thus *position specific*, filter weights and also the affinity matrix, which are used to propagate information among the feature nodes via message passing. The dynamic

---

| (a) Fully-connected message passing | (b) Locally-connected message passing | (c) Dynamic graph message passing |

Figure 1: Contextual information is crucial for complex scene understanding tasks. To recognise the "boathouse", one needs to consider the "boat" and the "water" next to it. Fully-connected message passing models (a) are able to obtain this information, but are prohibitively expensive. Furthermore, they capture a lot of redundant information (*i.e.*"trees" and "sky"). Locally-connected models (b) are more efficient, but miss out on important context. Our proposed approach (c), dynamically samples a small subset of relevant feature nodes based on a *learned* dynamic sampling scheme, *i.e.* the *learned* position-specific random walk (indicated by the white dashed arrow lines), and also dynamically predicts filter weights and affinities (indicated by unique edge and square colors.), which are both conditioned on the sampled feature nodes.

weights and affinities are especially beneficial to specifically model each sampled feature context, leading to more effective message passing. Both of these dynamic properties are jointly optimised in a single model, and we modularise the DGMN as a network layer for simple deployment into existing networks.

We demonstrate the proposed model on the tasks of semantic segmentation, object detection and instance segmentation on the challenging Cityscapes [9] and COCO [26] datasets. We achieve significant performance improvements over the fully-connected Non-local model [34], while using substantially fewer floating point operations (FLOPs). Significantly, one variant of our model with dynamic filters and affinities (*i.e.* the second dynamic property) achieves similar performance to Non-local while only using 9.4% of its FLOPs and 25.3% of its parameters. Furthermore, "plugging" our module into existing networks, we show considerable improvements with respect to strong, state-of-the-art baselines on three different tasks and backbone architectures.

2. Related work

An early technique for modelling context for computer vision tasks involved conditional random fields. In particular, the DenseCRF model [21] was popular as it modelled interactions between all pairs of pixels in an image. Although such models have been integrated into neural networks [43, 1, 2, 37], they are limited by the fact that the pairwise potentials are based on simple handcrafted features, Moreover, they mostly model discrete label spaces, and are thus not directly applicable in the feature learning task since feature variables are typically continuous. Coupled with the fact that CRFs are computationally expensive, CRFs are no longer used for most computer vision tasks.

A complementary technique for increasing the receptive field of CNNs was to use dilated convolutions [5, 39]. With dilated convolutions, the number of parameters does not change, while the receptive field grows exponentially if the dilation rate is linearly increased in successive layers. Other modifications to the convolution operation include deformable convolution [10, 44], which learns the offset with respect to a predefined grid from which to select input values. However, the weights of the deformable convolution filters do not depend on the selected input, and are in fact shared across all different positions. In contrast, our dynamic sampling aims to sample over the whole feature graph to obtain a large receptive field, and the predicted affinities and the weights for message passing are *position specific* and *conditioned* on the dynamically sampled nodes. Our model is thus able to better capture position-based semantic context to enable more effective message passing among feature nodes.

The idea of sampling graph nodes has previously been explored in GraphSAGE [13]. Crucially, GraphSAGE simply uniformly samples nodes. In contrast, our sampling strategy is *learned* based on the node features. Specifically, we first sample the nodes uniformly in the spatial dimension, and then dynamically predict *walks* of each node conditioned on the node features. Furthermore, GraphSAGE does not consider our second important property, *i.e.* the dynamic prediction of the affinities and the message passing kernels.

We also note that [18] developed an idea of "dynamic convolution", that is predicting a dynamic convolutional filter for each feature position. More recently, [35] further reduced the complexity of this operation in the context of natural language processing with lightweight grouped convolutions. Unlike [18, 35], we present a graph-based formulation, and jointly learn dynamic weights and dynamic affinities, which are conditioned on an *adaptively sampled* neighbourhood for each feature node in the graph using the proposed dynamic

sampling strategy for effective message passing.

3. Dynamic graph message passing networks

3.1. Problem definition and notation

Given an input feature map interpreted as a set of feature vectors, *i.e.* $\mathbf{F} = \{\mathbf{f}_i\}_{i=1}^N$ with $\mathbf{f}_i \in \mathbb{R}^{1 \times C}$, where N is the number of pixels and C is the feature dimension, our goal is to learn a set of refined latent feature vectors $\mathbf{H} = \{\mathbf{h}_i\}_{i=1}^N$ by utilising hidden structured information among the feature vectors at different pixel locations. $\mathbf{H}$ has the same dimension as the observation $\mathbf{F}$. To learn such structured representations, we convert the feature map into a graph domain by constructing a feature graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, A\}$ with $\mathcal{V}$ as its nodes, $\mathcal{E}$ as its edges and A as its adjacency matrix. Specifically, the nodes of the graph are represented by the latent feature vectors, *i.e.* $\mathcal{V} = \{\mathbf{h}_i\}_{i=1}^N$, and $A \in \mathbb{R}^{N \times N}$ is a binary or learnable matrix with self-loops describing the connections between nodes. In this work, we propose a novel dynamic graph message passing network [12] for deep representation learning, which refines each graph feature node by passing messages on the graph $\mathcal{G}$. Different from existing message passing neural networks considering a fully- or locally-connected static graph [34, 12], we propose a dynamic graph network model with two dynamic properties, *i.e. dynamic sampling* of graph nodes to approximate the full graph distribution, and *dynamic prediction* of node-conditioned filter weights and affinities, in order to achieve more efficient and effective message passing.

3.2. Graph message passing neural networks for deep representation learning

Message passing neural networks (MPNNs) [12] present a generalised form of graph neural networks such as graph convolution networks [19], gated graph sequential networks [24] and graph attention networks [32]. In order to model structured graph data, in which latent variables are represented as nodes on an undirected or directed graph, feed-forward inference is performed through a message passing phase followed by a readout phase upon the graph nodes. The message passing phase usually takes T iteration steps to update feature nodes, while the readout phase is for the final prediction *e.g.* graph classification with updated nodes. In this work, we focus on the message passing phase for learning efficient and effective feature refinement, since well-represented features are critical in all downstream tasks. The message passing phase consists of two steps, *i.e.* a message calculation step M^t and a message updating step U^t. Given a latent feature node $\mathbf{h}_i^{(t)}$ at an iteration t, for computational efficiency, we consider a locally connected node field with $v_i \subset \mathcal{V}$ and $v_i \in \mathbb{R}^{(K \times C)}$, where $K \ll N$ is the number of sampled nodes in v_i. Thus we can define the message

calculation step for node i operated locally as

$$
\begin{aligned}
\mathbf{m}_i^{(t+1)} &= M^t \left(A_{i,j}, \{\mathbf{h}_1^{(t)}, \cdots, \mathbf{h}_K^{(t)}\}, \mathbf{w}_j \right) \\
&= \sum_{j \in \mathcal{N}(i)} A_{i,j} \mathbf{h}_j^{(t)} \mathbf{w}_j,
\end{aligned} \tag{1}
$$

where $A_{i,j} = A[i,j]$ describes the connection relationship *i.e.* the affinity between latent nodes $\mathbf{h}_i^{(t)}$ and $\mathbf{h}_j^{(t)}$, $\mathcal{N}(i)$ denotes a self-included neighborhood of the node $\mathbf{h}_i^{(t)}$ which can be derived from v_i and $\mathbf{w}_j \in \mathbb{R}^{C \times C}$ is a transformation matrix for message calculation on the hidden node $\mathbf{h}_j^{(t)}$. The message updating function U^t then updates the node $\mathbf{h}_i^{(t)}$ with a linear combination of the calculated message and the observed feature $\mathbf{f}_i$ at the node position i as:

$$
\mathbf{h}_i^{(t+1)} = U^t \left(\mathbf{f}_i, \mathbf{m}_i^{(t+1)} \right) = \sigma \left(\mathbf{f}_i + \alpha_i^m \mathbf{m}_i^{(t+1)} \right), \tag{2}
$$

where α_i^m of a learnable parameter for scaling the message, and the operation $\sigma(\cdot)$ is a non-linearity function *e.g.* ReLU. By iteratively performing message passing on each node with T steps, we obtain a refined feature map $\mathbf{H}^{(T)}$ as output.

3.3. From a fully-connected graph to a dynamic sampled graph

A fully-connected graph typically contains many connections and parameters, which, in addition to computational overhead, results in redundancy in the connections, and also makes the network optimisation more difficult especially when dealing with limited training data. Therefore, as in Eq. 1, a local node connection field is considered in the graph message passing network. However, in various computer vision tasks, such as detection and segmentation, learning deep representations capturing both local and global receptive fields is important for the model performance [29, 28, 23]. To maintain a large receptive field while utilising much fewer parameters than the fully-connected setting, we further explore dynamic sampling strategies in our proposed graph message passing network. We develop a uniform sampling scheme, which we then extend to a predicted random walk sampling scheme, aimed at reducing the redundancy found in a fully-connected graph. This sampling is performed in a dynamic fashion, meaning that for a given node $\mathbf{h}_i$, we aim to sample an optimal subset of v_i from $\mathcal{V}$ to update $\mathbf{h}_i$ via message passing as shown in Fig. 2.

Multiple uniform sampling for dynamic receptive fields. Uniform sampling is a commonly used strategy for graph node sampling [22] based on Monte-Carlo estimation. To approximate the distribution of $\mathcal{V}$, we consider a set of S uniform sampling rates φ with $\varphi = \{\rho_q\}_{q=1}^S$, where ρ_q is a sampling rate. Let us assume that the latent feature nodes are located in a P-dimensional space $\mathbb{R}^P$. For instance, $P = 2$ for images considering the x- and y-axes. For each
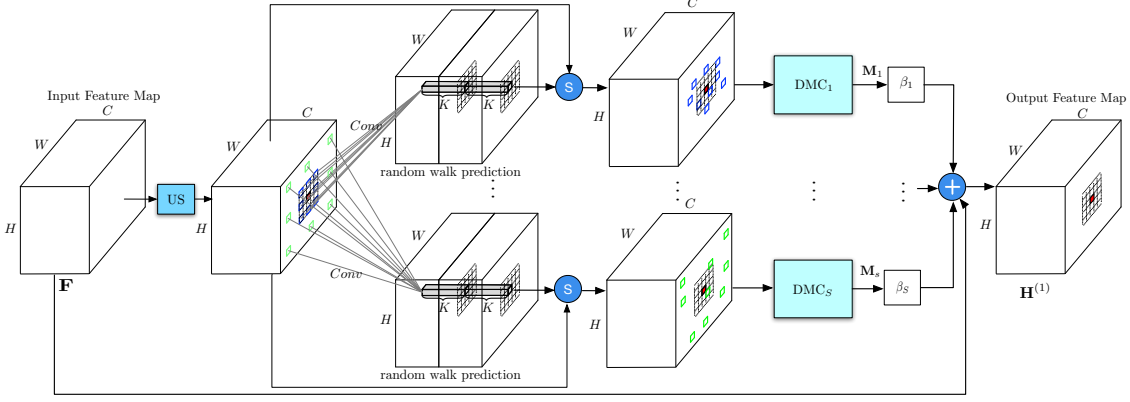
Figure 2: Overview of our proposed dynamic graph message passing network (DGMN). The neighbourhood used to update the feature representation of each node (we show a single node with a red square) is predicted dynamically conditioned on each input. This is done by first uniformly sampling (denoted by "US") a set of S neighbourhoods around each node. Each neighbourhood contains K (*e.g.* 3×3) sampled nodes. Here, the blue nodes were sampled with a low sampling rate, and the green ones with a high sampling rate. Walks are predicted (conditioned on the input) from these uniformly sampled nodes, denoted by the Ⓢ symbol representing the random walk sampling operation described in Sec. 3.3. $\text{DMC}_1, \cdots, \text{DMC}_S$ and $\beta_1, \cdots, \beta_S$ denotes S dynamic message calculation operations and S message scaling parameters, respectively. The DMC module is detailed in Figure 3. The symbol $\oplus$ indicates an element-wise addition operation.

latent node $\mathbf{h}_i$, a total of K neighbouring nodes are sampled from $\mathbb{R}^P$. The receptive field of v_i is thus determined by ρ_q and K. Note that the sampling rate ρ_q corresponds to the "dilation rate" often used in convolution [39] and is thus able to capture a large receptive field whilst maintaining a small number of connected nodes. Thus we can achieve much lower computational overhead compared with fully-connected message passing in which typically all N nodes are used when one of the nodes is updated. Each node receives S complementary messages from distinct receptive fields for updating as

$$\mathbf{m}_i^{(t+1)} = \sum_q \sum_{j \in \mathcal{N}_q(i)} \beta_q A_{i,j}^q \mathbf{h}_j^{(t)} \mathbf{w}_j^q, \qquad (3)$$

where β_q is a weighting parameter for the message from the q-th sampling rate and $q = 1, \cdots, S$. A^q denotes an adjacency matrix formed under a sampling rate ρ_q, with $A_{i,j}^q$, $\mathbf{w}_j^q$ and $\mathcal{N}_q(i)$ defined analogously. The uniform sampling scheme acts as a linear sampler based on the spatial distribution while not considering the original feature distribution of the hidden nodes, *i.e.* sampling independently of the node features. Eq. 2 can still be used to update the nodes.

Learning position-specific random walks for node-dependant adaptive sampling. To take into account the feature data distribution when sampling nodes, we further present a random walk strategy upon the uniform sampling. Walks around the uniformly sampled nodes could sample the graph in a non-linear and adaptive manner, and we believe that it could facilitate learning better approximation of the original feature distribution. The "random" here refers to the fact that the walks are predicted in a data-driven fashion from stochastic gradient descent. Given a matrix, $v_i^q \in \mathbb{R}^{K \times C}$, constructed from K uniformly sampled nodes under a sam-

pling rate, ρ_q, the random walk of each node is further estimated based on the feature data of the sampled nodes. Given the P-dimensional space where the nodes distribute ($P = 2$ for images), let us denote $\triangle \mathbf{d}_j^q \in \mathbb{R}^{P \times 1}$ as predicted walks from a uniformly sampled node $\mathbf{h}_j$ with $j \in \mathcal{N}_q(i)$. The node walk prediction can then be performed using a matrix transformation as

$$\triangle \mathbf{d}_j^q = \mathbf{W}_{i,j}^q v_i^q + \mathbf{b}_{i,j}^q, \qquad (4)$$

where $\mathbf{W}_{i,j}^q \in \mathbb{R}^{P \times (K \times C)}$ and $\mathbf{b}_{i,j}^q \in \mathbb{R}^{P \times 1}$ are the matrix transformation parameters, which are learned separately for each node v_i^q. With the predicted walks, we can obtain a new set of adaptively sampled nodes $v_i'^q$, and generate the corresponding adjacency matrix A'^q, which can be used to calculate the messages as

$$\mathbf{m}_i^{(t+1)} = \sum_q \sum_{j \in \mathcal{N}_q(i)} \beta_q A_{i,j}'^q \varrho \left(\mathbf{h}_j'^{(t)} | \mathcal{V}, j, \triangle \mathbf{d}_j^q \right) \mathbf{w}_j^q, \qquad (5)$$

where the function $\varrho(\cdot)$ is a bilinear sampler [17] which samples a new feature node $\mathbf{h}_j'^{(t)}$ around $\mathbf{h}_j^{(t)}$ given the predicted walk $\triangle \mathbf{d}_j^q$ and the whole set of graph vertexes $\mathcal{V}$.

3.4. Joint learning of node-conditioned dynamic filters and affinities

In the message calculation formulated in Eq. 5, the set of weights $\{\mathbf{w}_j^q\}_{j=1}^K$ of the filter is shared for each adaptively sampled node field $v_i'^q$. However, since each $v_i'^q$ essentially defines a node-specific local feature context, it is more meaningful to use a node-conditioned filter to learn the message for each hidden node $\mathbf{h}_i'^{(t)}$. In additional to the filters for
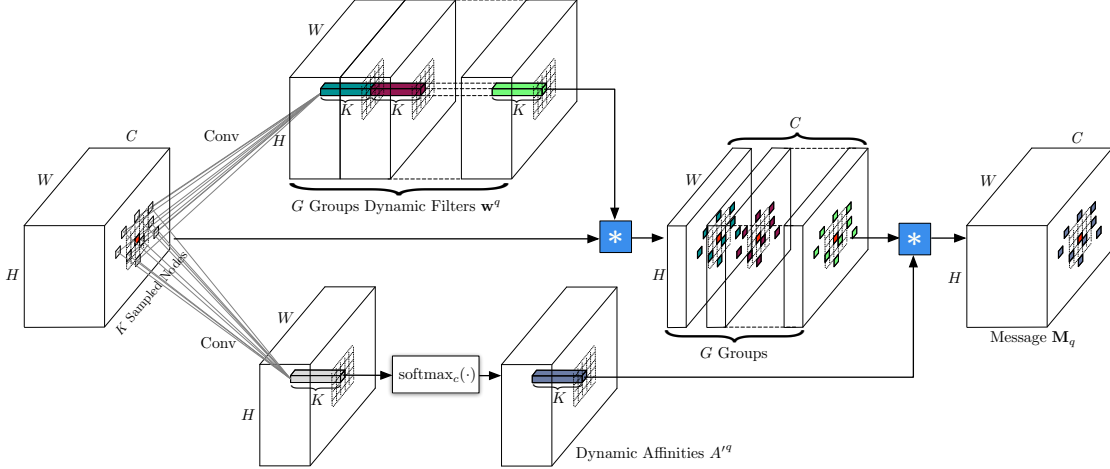
Figure 3: Schematic illustration of the proposed dynamic message passing calculation (DMC) module. The small red square indicates the receiving node whose message is calculated from its neighbourhood, *i.e.* the sampled K (*e.g.* 3×3) features nodes. The module accepts a feature map as input and produces its corresponding message map. The symbol $*$ denotes group convolution operation using the dynamically predicted and position specific group kernels and affinities.

the message calculation in Eq. 5, the affinity $A'_{i,j}$ of any pair of nodes $\mathbf{h}'^{(t)}_i$ and $\mathbf{h}'^{(t)}_j$ could be also be predicted and should also be conditioned on the node field v'^q_i, since the affinity reweights the message passing only in v'^q_i. As shown in Fig. 3, we thus use matrix transformations to simultaneously estimate the dynamic filter and affinity which are both conditioned on v'^q_i,

$$\{\mathbf{w}^q_j, A'^q_{i,j}\} = \mathbf{W}^{k,A}_{i,j} v'^q_i + \mathbf{b}^{k,A}_{i,j}, \qquad (6)$$

$$A'^q_{i,j} = \text{softmax}_c(A'^q_{i,j}) = \frac{\exp(A'^q_{i,j})}{\sum_{l \in \mathcal{N}_q(i)} \exp(A'^q_{i,l})}, \quad (7)$$

where the function $\text{softmax}_c(\cdot)$ denotes a softmax operation along the channel axis, which is used to perform a normalisation on the estimated affinity $A'^q_{i,j} \in \mathbb{R}^1$. $\mathbf{W}^{k,A}_{i,j} \in \mathbb{R}^{(G \times C+1) \times (K \times C)}$ and $\mathbf{b}^{k,A}_{i,j} \in \mathbb{R}^{(G \times C+1)}$ are matrix transformation parameters. To reduce the number of the filter parameters, we consider grouped convolutions [8] with a set of G groups split from the total C feature channels, and $G \ll C$, *i.e.* each group of C/G feature channels shares the same set of filter parameters. The predicted dynamic filter weights and the affinities are then used in Eq. 5 for dynamic message calculation.

3.5. Modular instantiation

Figures 2 and 3 shows how our proposed dynamic graph message passing network (DGMN) can be implemented in a neural network. The proposed module accepts a single feature map $\mathbf{F}$ as input, which can be derived from any CNN layer. $\mathbf{H}^{(0)}$ denotes an initial state of the latent feature map, $\mathbf{H}$, and is initialised with $\mathbf{F}$. $\mathbf{H}$ and $\mathbf{F}$ have the same dimension, *i.e.* $\mathbf{F}, \mathbf{H} \in \mathbb{R}^{H \times W \times C}$, where H, W and C are the height, width and the number of feature channels of

the feature map respectively. We first define a set of S uniform sampling rates (we show two uniform sampling rates in Fig. 2 for clarity). The uniform and the random walk sampler sample the nodes from the full graph and return the node indices for subsequent dynamic message calculation (DMC) in Fig. 3. The matrix transformation $\mathbf{W}^q_{i,j}$ to estimate the node random walk in Eq. 4 is implemented by a 3×3 convolution layer [10]. Note that other sampling strategies could also be flexibly employed in our framework.

The sampled feature nodes are processed along two data paths: one for predicting the node-dependant dynamic affinities $A'^q \in \mathbb{R}^{H \times W \times K}$ and another path for dynamic filters $\mathbf{w}^q \in \mathbb{R}^{H \times W \times K \times G}$ where K (*e.g.* , 3×3) is the kernel size for the receiving node. The matrix transformation $\mathbf{W}^{k,A}_{i,j}$ used to jointly predict the dynamic filters and affinities in Eq 6 is implemented by a 3×3 convolution layer. Message $\mathbf{M}_q \in \mathbb{R}^{H \times W \times C}$ corresponding to the q-th sampling rate is then scaled to perform a linear combination with the observed feature map $\mathbf{F}$, to produce a refined feature map $\mathbf{H}^{(1)}$ as output. To balance performance and efficiency, as in existing graph-based feature learning models [34, 7, 25, 40], we also perform $T = 1$ iteration of message updating.

3.6. Discussion

Our approach is related to deformable convolution [10, 44] and Non-local [33], but has several key differences:

A fundamental difference to deformable convolution is that it only learns the offset dependent on the input feature while the filter weights are fixed for all inputs. In contrast, our model learns the random walk, weight and affinity as all being dependent on the input. This property makes our weights and affinities position-specific whereas deformable convolution shares the same weight across all convolution positions in the feature map. Moreover, [10, 44] only con-

sider 3×3 local neighbours at each convolution position. In contrast, our model, for each position, learns to sample a set of K nodes (where $K \gg 9$) for message passing globally from the whole feature map. This allows our model to capture a larger receptive field than deformable convolution.

Whilst Non-local also learns to refine deep features, it uses a self-attention matrix to guide the message passing between each pair of feature nodes. In contrast, our model learns to sample graph feature nodes to capture global feature information efficiently. This dynamic sampling reduces computational overhead, whilst still being able to improve upon the accuracy of Non-local across multiple tasks as shown in the next section.

4. Experiments

4.1. Experimental setup

Tasks and datasets. We evaluate our proposed model on two challenging public benchmarks, *i.e.* Cityscapes [9] for semantic segmentation, and COCO [26] for object detection and instance segmentation. Both datasets have hidden test sets which are evaluated on a public evaluation server. We follow the standard protocol and evaluation metrics used by these public benchmarks. More details can be found in supplementary material.

Baseline models. For semantic segmentation on Cityscapes, our baseline is Dilated-FCN [39] with a ResNet-101 backbone pretrained on ImageNet. A randomly initialised 3×3 convolution layer, together with batch normalisation and ReLU is used after the backbone to produce a dimension-reduced feature map of 512 channels which is then fed into the final classifier. For the task of object detection and instance segmentation on COCO, our baseline is Mask-RCNN [14, 27] with FPN and ResNet/ResNeXt [15, 36] as a backbone architecture. Unless otherwise specified, we use a single scale and test with a single model for all experiments without using other complementary performance boosting "tricks". We train models on the COCO training set and test on the validation and test-dev sets.

Across all tasks and datasets, we consider Non-local networks [34] as an additional baseline. To have a direct comparison with the Deformable Convolution method [10, 44], we consider two baselines: (i) "deformable message passing", which is a variant of our model using randomly initialised deformable convolutions for message calculation, but without using the proposed dynamic sampling and dynamic weights/affinities strategies, and (ii) the original deformable method which replaces the convolutional operations as in [10, 44].

Implementation details. For our experiments on Cityscapes, our DGMN module is randomly initialised and inserted between the 3×3 convolution layer and the final classifier. For the experiments on COCO, we insert one

	mIoU (%)	Params	FLOPs
Dilated FCN [39]	75.0	–	–
+ Deformable [44]	78.2	+1.31M	+12.34G
+ ASPP [6]	78.9	+4.42M	+38.45G
+ Non-local [34]	79.0	+2.88M	+73.33G
+ DGMN w/ DA	76.5	+0.57M	+5.32G
+ DGMN w/ DA+DW	79.1	+0.73M	+6.88G
+ DGMN w/ DA+DW+DS	**80.4**	+2.61M	+24.55G

Table 1: Ablation study on the Cityscapes validation set for semantic segmentation. All models have a ResNet-101 backbone and are evaluated at a single scale.

or multiple randomly initialised DGMN modules into the backbone for deploying our approach for feature learning. Our models and baselines for all of our COCO experiments are trained with the typical "1x" training settings from the public Mask R-CNN benchmark [27].

When predicting dynamic filter weights, we used the grouping parameter $G = 4$. For our experiments on Cityscapes, the sample rates are set to $\varphi = \{1, 6, 12, 24, 36\}$. For experiments on COCO, we use smaller sampling rates of $\varphi = \{1, 4, 8, 12\}$. The effect of this hyperparameter and additional implementation details are described in the supplementary material.

4.2. Model analysis

To demonstrate the effectiveness of the proposed components of our model, we conduct ablation studies of: (i) DGMN w/ DA, which adds the dynamic affinity (DA) strategy onto the DGMN base model; (ii) DGMN w/ DA+DW, which further adds the proposed dynamic weights (DW) prediction; (iii) DGMN w/ DA+DW+DS, which is our full model with the dynamic sampling (DS) scheme added upon DGMN w/ DA+DW. Note that DGMN Base was described in Sec. 3.2, DS in Sec. 3.3, and DA and DW in 3.4.

Effectiveness of the dynamic sampling strategy. Table 1 shows quantitative results of semantic segmentation on the Cityscapes validation set. DGMN w/ DA+DW+DS clearly outperforms DGMN w/ DA+DW on the challenging segmentation task, meaning that the feature-conditioned adaptive sampling based on learned random walks is more effective compared to a spatial uniform sampling strategy when selecting nodes. More importantly, all variants of our module for both semantic segmentation and object detection that use dynamic sampling (Tab. 1 and Tab. 2) achieve higher performance than a fully-connected model (*i.e.* Non-local [34]) with substantially fewer FLOPs. This emphasises the performance benefits of our dynamic graph sampling model. Visualisations of the nodes dynamically sampled by our model are shown in Fig. 4.

Effectiveness of joint learning the dynamic filters and affinities. As shown in Tab. 1, DGMN w/ DA is 1.5 points better than Dilated FCN baseline with only a slight increase
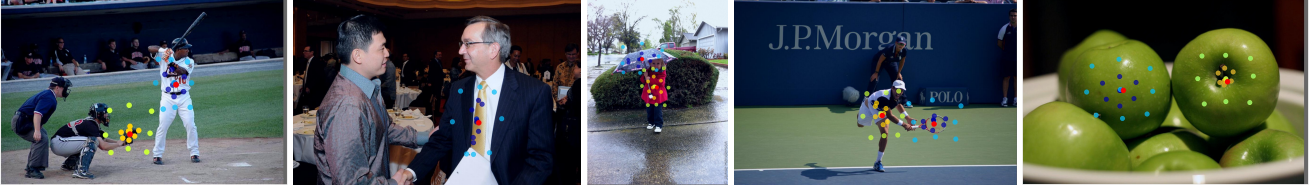
Figure 4: Visualisation of the nodes sampled via *learning* the random walks with our network (trained for instance segmentation on COCO). The red point indicates a receiving node i. Different colour families (*i.e.* yellow and blue) indicate the learned position specific weights and affinities of the sampled nodes. Different colours in the same family show the sampled nodes with different sampling rates for the same receiving node.

	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
Mask R-CNN baseline	37.8	59.1	41.4	34.4	55.8	36.6
+ GCNet [4]	38.1	60.0	41.2	34.9	56.5	37.2
+ Deformable Message Passing	38.7	60.4	42.4	35.0	56.9	37.4
+ Non-local [34]	39.0	61.1	41.9	35.5	58.0	37.4
+ CCNet [16]	39.3	-	-	36.1	-	-
+ DGMN	**39.5**	**61.0**	**43.3**	**35.7**	**58.0**	**37.9**
+ GCNet (C5) [4]	38.7	61.1	41.7	35.2	57.4	37.4
+ Deformable (C5) [44]	39.9	-	-	34.9	-	-
+ DGMN (C5)	**40.2**	**62.0**	**43.4**	**36.0**	**58.3**	**38.2**

Table 2: Quantitative results of different models on the COCO 2017 validation set for object detection (AP^b) and instance segmentation (AP^m). C5 denotes inserting DGMN after all 3×3 convolutional layers in *res5*. All methods are based on the Mask R-CNN with ResNet-50 as backbone.
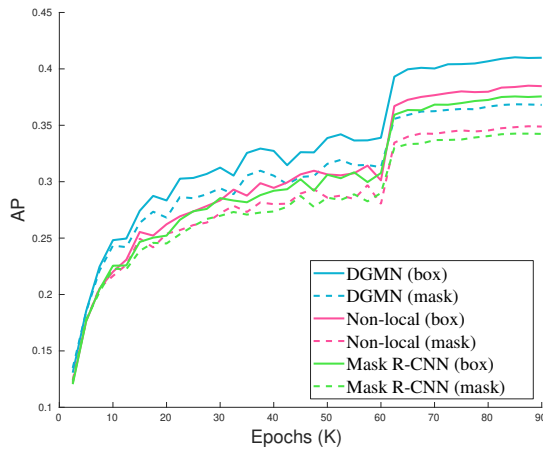

Figure 5: Validation curves of AP^{box} and AP^{mask} on COCO for Mask-RCNN baseline, Non-local and the proposed DGMN. The number of training epochs is 90K.

in FLOPs and parameters, showing the benefit of using predicted dynamic affinities for reweighting the messages in message passing. By further employing the estimated dynamic filter weights for message calculation, the performance increases substantially from a mIoU of 76.5% to 79.1%, which is almost the same as the 79.2% of the Non-local model [34]. Crucially, our approach only uses 9.4% of the FLOPs and 25.3% of the parameters compared to Non-local. These results clearly demonstrate our motivation of jointly learning the dynamic filters and dynamic affinities from sampled graph nodes.

Comparison with other baselines. For semantic segmenta-

tion on Cityscapes, we clearly outperform the ASPP module of Deeplab v3 [6] which also increases the receptive field by using multiple dilation rates. Notably, we improve upon the most related method, Non-local [34] which models a fully-connected graph, whilst using only 33% of the FLOPs of [34]. This suggests that a fully-connected graph models redundant information, and further confirms the performance and efficiency of our model.

For fair comparison with Non-local [34] as well as other alternatives on COCO, we insert one randomly initialised DGMN module right before the last residual block of *res4* [34] (Tab. 2). We also compare to GCNet [4] and CC-Net [16] which both aim to reduce the complexity of the fully connected Non-local model. Our proposed DGMN model substantially improves upon these strong baselines and alternatives. Figure 5 further shows the validation curves of our method and different baselines using the standard AP^{Box} and AP^{Mask} measures for semantic and instance segmentation respectively. Our method is consistently better than the Non-local and Mask R-CNN baselines throughout training.

Effectiveness of multiple DGMN modules. We further show the effectiveness of our approach for representation learning by inserting multiple of our DGMN modules into the ResNet-50 backbone. Specifically, we add our full DGMN module *after* all 3×3 conv layers in *res5* which we denote as "C5". The second part of Tab. 2 shows that our model significantly improves upon the Mask R-CNN baseline with the improvements of 2.4 points for the AP^{box} on object detection, and 1.6 points for the AP^{mask} on instance segmentation. Furthermore, when we insert the GCNet module [4] in the same locations for comparison, our model achieves better performance too. A straightforward comparison to the deformable method of [44] is the deformable message passing baseline in which we disable the proposed dynamic sampling and the dynamic weights and affinities learning strategies. In Tab. 2, our model significantly improves upon the deformable message passing method, which is a direct evidence of the effectiveness of jointly modeling the dynamic sampling and dynamic filters and affinities for feature learning. Furthermore, we also consider inserting the improved Deformable Conv method [44] in C5, which is complementary to our model since it is plugged *before* 3×3
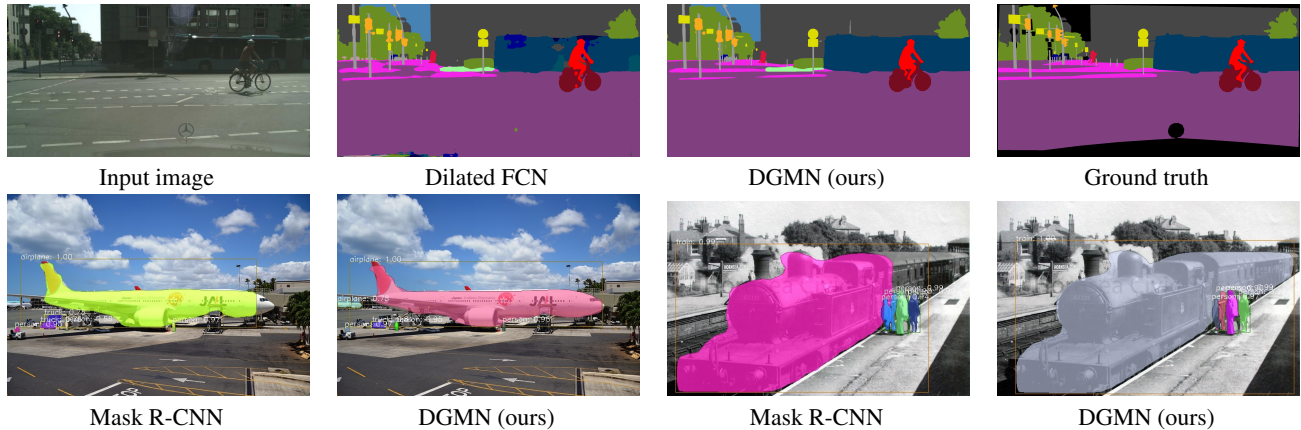
Figure 6: Qualitative examples of our results for semantic segmentation on Cityscapes (first row), and object detection and instance segmentation on COCO (second row)

	Backbone	mIoU (%)
PSPNet [41]	ResNet 101	78.4
PSANet [42]	ResNet 101	80.1
DenseASPP [38]	DenseNet 161	80.6
GloRe [7]	ResNet 101	80.9
Non-local [34]	ResNet 101	81.2
CCNet [16]	ResNet 101	81.4
DANet [11]	ResNet 101	81.5
DGMN (Ours)	ResNet 101	**81.6**

Table 3: Comparison to state-of-the-art for semantic segmentation on Cityscapes. All methods are trained with the finely-annotated data from the training and validation sets.

	Backbone	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}
Mask R-CNN baseline		38.0	59.7	41.5	34.6	56.5	36.6
+ DGMN (C5)	ResNet 50	40.2	62.5	43.9	36.2	59.1	38.4
+ DGMN (C4, C5)		**41.0**	**63.2**	**44.9**	**36.8**	**59.8**	**39.1**
Mask R-CNN baseline	ResNet 101	40.2	61.9	44.0	36.2	58.6	38.4
+ DGMN (C5)		**41.9**	**64.1**	**45.9**	**37.6**	**60.9**	**40.0**
Mask R-CNN baseline	ResNeXt 101	42.6	64.9	46.6	38.3	61.6	40.8
+ DGMN (C5)		**44.3**	**66.8**	**48.4**	**39.5**	**63.3**	**42.1**

Table 4: Quantitative results via plugging our DGMN module on different backbones on the COCO 2017 `test-dev` set for object detection (AP^{box}) and instance segmentation (AP^{mask}).

layers to replace the convolution operations. Our approach also achieves better performance than it.

4.3. Comparison to State-of-the-art

Performance on Cityscapes test set. Table 3 compares our approach with state-of-the-art methods on Cityscapes. Note that all methods are trained using only the fine annotations and evaluated on the public evaluation server as test-set annotations are withheld from the public. As shown in the table, DGMN (ours) achieves an mIoU of 81.6%, surpassing all previous works. Among competing methods, GloRe [7], Non-local [34], CCNet [16] and DANet [11] are the most related to us as they all based on graph neural network modules. Note that we followed common practice and employed several complementary strategies used in semantic segmentation to boost performance, including Online Hard Example Mining (OHEM) [11], Multi-Grid [6] and Multi-Scale (MS) ensembling [41]. The contribution of each strategy on the

final performance is reported in the supplementary.

Performance on COCO 2017 test set. Table 4 presents our results on the COCO test-dev set, where we inserted our module on multiple backbones. By inserting DGMN into all layers of C4 and C5, we substantially improve the performance of Mask R-CNN, observing a gain of of 3.0 and 2.2 points on the AP^{box} and the AP^{mask} of object detection and instance segmentation respectively. We observe similar improvements when using the ResNet-101 or ResNeXt-101 backbones as well, showing that our proposed DGMN module generalises to multiple backbone architectures.

Note that the Mask-RCNN baseline with ResNet-101 has 63.1M parameters and 354 GFLOPs. Our DGMN (C4, C5) model with ResNet-50 outperforms it whilst having only 51.1M parameters and 297.1 GFLOPs. This further shows that the improvements from our method are due to the model design, and not only the increased parameters and computation. Further comparisons to the state-of-the-art are included in the supplementary.

5. Conclusion

We proposed Dynamic Graph Message Passing Networks, a novel graph neural network module that dynamically determines the graph structure for each input. It learns dynamic sampling of a small set of relevant neighbours for each node, and also predicts the weights and affinities dependant on the feature nodes to propagate information through this sampled neighbourhood. This formulation significantly reduces the computational cost of static, fully-connected graphs such as Non-local [34] which contain many redundancies. This is demonstrated by the fact that we are able to clearly improve upon the accuracy of Non-local, and several state-of-art baselines, on three complex scene understanding problems.

References

[1] Anurag Arnab, Sadeep Jayasumana, Shuai Zheng, and Philip HS Torr. Higher order conditional random fields in deep neural networks. In *ECCV*, 2016. 2

[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. 2

[3] Xue Bai and Guillermo Sapiro. Geodesic matting: A framework for fast interactive image and video segmentation and matting. *IJCV*, 2009. 1

[4] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. *arXiv preprint arXiv:1904.11492*, 2019. 7

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected CRFs. In *ICLR*, 2015. 2

[6] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 6, 7, 8

[7] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Shuicheng Yan, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. In *CVPR*, 2019. 5, 8

[8] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, 2017. 5

[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 *CVPR*, 2016. 1, 2, 6

[10] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017. 2, 5, 6

[11] Jun Fu, Jing Liu, Haijie Tian, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, 2019. 8

[12] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *ICML*, 2017. 1, 3

[13] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, 2017. 1, 2

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 6

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 6

[16] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *ICCV*, 2019. 7, 8

[17] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *NeurIPS*, 2015. 4

[18] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. In *NeurIPS*, 2016. 2

[19] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017. 1, 3

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 1

[21] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *NeurIPS*, 2011. 2

[22] Jure Leskovec and Christos Faloutsos. Sampling from large graphs. In *SIGKDD*, 2006. 3

[23] Xiangtai Li, Li Zhang, Ansheng You, Maoke Yang, Kuiyuan Yang, and Yunhai Tong. Global aggregation then local distribution in fully convolutional networks. In *BMVC*, 2019. 3

[24] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. In *ICLR*, 2015. 3

[25] Guosheng Lin, Chunhua Shen, Ian Reid, and Anton van den Hengel. Deeply learning the messages in message passing inference. In *NeurIPS*, 2015. 5

[26] 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*, 2014. 2, 6

[27] Francisco Massa and Ross Girshick. maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch. https://github.com/facebookresearch/maskrcnn-benchmark, 2018. 6

[28] Aude Oliva and Antonio Torralba. The role of context in object recognition. *Trends in cognitive sciences*, 2007. 1, 3

[29] Andrew Rabinovich, Andrea Vedaldi, Carolina Galleguillos, Eric Wiewiora, and Serge Belongie. Objects in context. In *ICCV*, 2007. 1, 3

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 1

[32] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018. 3

[33] Panqu Wang, Pengfei Chen, Ye Yuan, Ding Liu, Zehua Huang, Xiaodi Hou, and Garrison Cottrell. Understanding convolution for semantic segmentation. In *WACV*, 2018. 5

[34] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 1, 2, 3, 5, 6, 7, 8

[35] Felix Wu, Angela Fan, Alexei Baevski, Yann N Dauphin, and Michael Auli. Pay less attention with lightweight and dynamic convolutions. In *ICLR*, 2019. 2

[36] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017. 6

[37] Dan Xu, Wanli Ouyang, Xavier Alameda-Pineda, Elisa Ricci, Xiaogang Wang, and Nicu Sebe. Learning deep structured multi-scale features using attention-gated crfs for contour prediction. In *NeurIPS*, 2017. 2

[38] Maoke Yang, Kun Yu, Chi Zhang, Zhiwei Li, and Kuiyuan Yang. Denseaspp for semantic segmentation in street scenes. In *CVPR*, 2018. 8

[39] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 2, 4, 6

[40] Li Zhang, Xiangtai Li, Anurag Arnab, Kuiyuan Yang, Yunhai Tong, and Philip HS Torr. Dual graph convolutional network for semantic segmentation. In *BMVC*, 2019. 5

[41] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 8

[42] 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 *ECCV*, 2018. 8

[43] 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 *ICCV*, 2015. 2

[44] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *CVPR*, 2019. 2, 5, 6, 7