Deep Grouping Model for Unified Perceptual Parsing

Zhiheng Li[1] Wenxuan Bao[2*] Jiayang Zheng[1] Chenliang Xu[1]
[1]University of Rochester [2]Tsinghua University
{zhiheng.li,jiayang.zheng,chenliang.xu}@rochester.edu bwx16@mails.tsinghua.edu.cn

Abstract

The perceptual-based grouping process produces a hierarchical and compositional image representation that helps both human and machine vision systems recognize heterogeneous visual concepts. Examples can be found in the classical hierarchical superpixel segmentation or image parsing works. However, the grouping process is largely overlooked in modern CNN-based image segmentation networks due to many challenges, including the inherent incompatibility between the grid-shaped CNN feature map and the irregular-shaped perceptual grouping hierarchy. Overcoming these challenges, we propose a deep grouping model (DGM) that tightly marries the two types of representations and defines a bottom-up and a top-down process for feature exchanging. When evaluating the model on the recent Broden+ dataset for the unified perceptual parsing task, it achieves state-of-the-art results while having a small computational overhead compared to other contextual-based segmentation models. Furthermore, the DGM has better interpretability compared with modern CNN methods.

1. Introduction

Deep CNN methods have achieved substantial performance improvement compared with non-CNN methods in the field of semantic segmentation [29, 5]. Many of them can achieve even better performance by incorporating *good practices* that have long been discovered in non-CNN methods, *e.g.*, multiscale features [59, 48, 44] and contextual information [57, 55, 19, 16, 58]. However, recent works still have some key limitations. First, many CNN-based methods are solely driven by the cross-entropy loss computed against ground-truth pixel labels, lacking an explicit modeling of the perceptual grouping process, which is an integral part in the human visual system [4]. Second, most modelings are still focusing on regular-shaped feature maps, which creates not only significant overhead in a multi-scale representation when considering feature-to-feature attention but also is sub-optimal for modeling irregular-shaped semantic regions on the image.

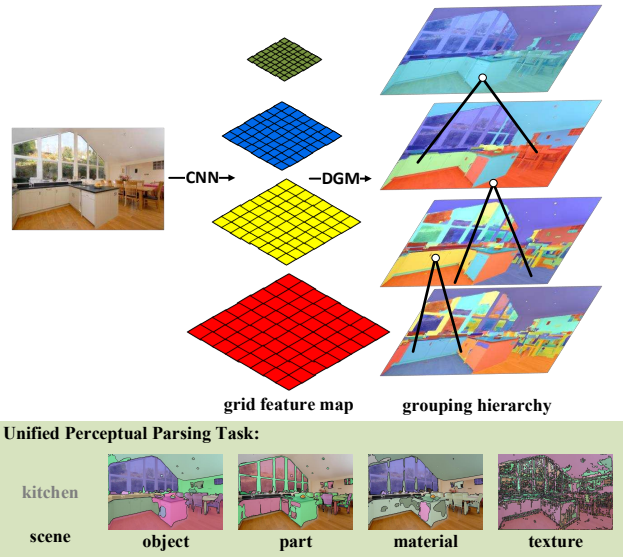

Figure 1: Perceptual grouping process. From fine to coarse: neighboring pixels form a part; parts group into an object; and objects combine into a contextual region. The DGM aims to marry a CNN with the grouping hierarchy for unified perceptual parsing of images. The grouping hierarchy is dynamically computed based on the CNN features, and the CNN features are enhanced by the grouping cues from the graph hierarchy. The model is applied to unified perceptual parsing task to show superiority of DGM.

To overcome these limitations, we revisit the classical perceptual grouping methods, *e.g.*, superpixel segmentation [37, 13, 31, 35] and image parsing [41, 38, 53], which were extensively studied before the predominance of CNNs in segmentation. The seminal work by Tu *et al.* [41] represents an image as a hierarchical graph, a.k.a. *parsing graph*. In their depicted example, an image of *a football match scene* is first decomposed into three elements: person, sports field, and spectator, then these elements are further decomposed, *e.g.*, the person consists of face and body texture. Such a graph is both compositional (*e.g.*, lower-level semantics induce grouping cues for higher-level semantics)

*The work was performed while Wenxuan Bao was a visiting student at University of Rochester.

and decompositional (*e.g.*, higher-level semantics provide feature support for lower-level semantics), and it varies upon the input image. In this work, we explore whether it is beneficial to inject such a perceptual grouping process explicitly in modern CNN frameworks for a unified image parsing of the scene (see Fig. 1 for an example).

Three challenges arise when incorporating the perceptual grouping process as a hierarchical graph in a deep CNN. First, there is feature incompatibility between the grid-shaped CNN feature maps and irregular-shaped graph nodes, not to mention how to benefit one from the other. Second, it is unclear how to dynamically grow the grouping hierarchy based on different levels of feature semantics extracted from the image. Although superpixel segmentation map provides a plausible initial grouping based on low-level textural and edge cues, high-level semantics of larger receptive fields are needed when growing parts into objects. Third, a holistic understanding of the scene is required when considering the unified pcerceptual parsing task. For example, knowing the scene-level *kitchen* label helps clarify *countertop* against *desk*. It is easy to do in a CNN but difficult in a parsing graph hierarchy.

To tackle the challenges as mentioned above, we propose a novel *Deep Grouping Model (DGM)*, which contains a few modules that are general enough to adapt to many CNNs. The *Expectation-Maximization Graph Pooling* (*EMGP*) module and *Projection* module transform multi-resolution feature maps into a multi-level graph by grouping different regions on the feature map in a bottom-up fashion (*i.e.*, from high- to low-resolution). They have several advantages. Since the model groups pixels and regions iteratively, the number of nodes in the graph is far smaller than the number of pixels on a feature map, which reduces computational overhead. The relationship between different levels of the hierarchy are learned during grouping, rather than assuming a uniform distribution such as in bilinear interpolation or adaptive average pooling on a grid feature map [48, 59]. Furthermore, the contextual information at one level of hierarchy can be quantified via edge weights in a graph, which is sparser than fully-connected non-local block [45, 55], leading to a lower overhead.

We put forward a *Top-down Message Passing (TDMP)* module, which propagates contextual information from the top-level graph to the bottom level graph by utilizing grouping results from *EMGP*. In this way, higher level context can be propagated *adaptively* to the corresponding irregular-shaped regions. For instance, object context features (*e.g.*, human) at higher-level graph will be propagated to its corresponding parts (*e.g.*, arms, legs, torso, etc.) at lower-level graph. Similarly, global scene context can also be propagated down to lower-level graph containing objects. Our proposed *TDMP* module is especially useful in the multi-task settings, where lower-level features enhanced by high-level semantics are able to produce better results. At the end, we use *Re-projection* module to re-project features from the hierarchical graph back to multi-resolution grid feature maps, which are used for down-stream tasks.

In order to prove the effectiveness of the proposed model, we apply our model on unified perceptual parsing task, a challenging task to recognize diverse perceptual concepts, including object (or stuff) segmentation, parts segmentation, scene classification, material segmentation, and texture prediction. We use the recent Broden+ dataset [2], a large-scale dataset combining five different datasets with heterogeneous task labels, that is designed for the unified perceptual parsing task. Our method is trained in a multi-task learning fashion, and we evaluate our model on each subtask. Results show that our method achieves the state-of-the-art on Broden+ dataset in every subtask.

Furthermore, the proposed DGM provides better interpretability thanks to the hierarchical graph representation. By using the grouping result , DGM can be applied to other two applications: 1) click propagation, 2) explainability with Grad-CAM, which are the building blocks in recent works on interactive segmentation [52, 30] and weakly-supervised segmentation [47, 46, 24].

2. Related Work

Grouping-based Method. Grouping-based segmentation method is extensively utilized before the deep learning methods. Ren *et al*. [37] propose grouping pixels into superpixels using Gestalt cues. Hierarchical grouping methods [1, 35, 42, 49, 50, 51] are also proposed for both image segmentation and video segmentation tasks. More recently, some deep learning methods start using grouping in the segmentation task. Gadde *et al*. [17] use superpixels to upsample CNN's low resolution prediction to the original image size. [40, 23] use deep feature rather than traditional low-level cues to predict superpixel map. Two works are closely related to our work. [21] puts forward local relation layer to model pixel-pair affinity in a predefined 7×7 square neighborhood, while our proposed model considers the neighborhood adaptively in an irregular-shaped region. Liang *et al*. [27] propose structure-envolving LSTM where Graph LSTM [28] is used for updating node features. In their work, only one pair of nodes is merged each time when a coarser graph is generated. Compared with [27], our model groups nodes more quickly thus reduces computational overhead. Farabet *et al*. [12] use multi-scale convolutional feature and conditional random field to regulate the probability of each pixel in segmentation prediction. In contrast, our work learns both grouping hierarchy and top-down message passing at feature level in a end-to-end fashion.

Graph Neural Network. Some recent works employ Graph Neural Network on segmentation task. Liang *et al*. [26] map feature maps to a concept tree to enable concept reason-

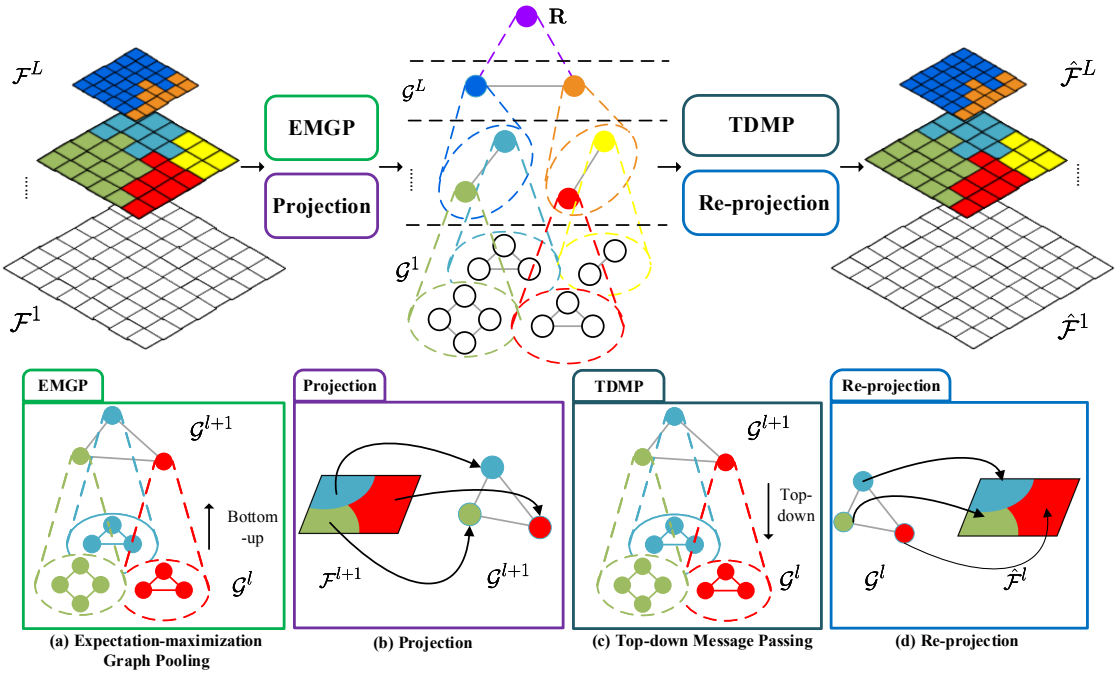

Figure 2: An overview of the proposed Deep Grouping Model (DGM).

ing. Other works [25, 8] project feature map to graph via linear transformation with learned anchor vectors or convolutional weights, which may be successful in classifying single pixel's semantic meaning but does not consider similarity between pairs of pixels to group them into a region. Ying *et al*. [54] propose a differentiable pooling method through predicting pooling weights by GraphSAGE [18], but the method does not consider pairwise similarity between graph nodes and the number of clusters is also fixed. In comparison, our model considers pairwise affinity among nodes and supports a dynamic number of clustering centers.

Contextual Modeling. Given the success of self-attention mechanism in many recognition tasks [45], recent work introduces self-attention module in the semantic segmentation field from different perspectives. Yuan *et al*. [55] propose object context pooling module. Fu *et al*. [15] apply attention mechanism on both position and channel. The aforementioned non-local based context modeling method creates large overhead since similarity between each pair of grid needs to be computed on the feature map. He *et al*. [19] introduces adaptive context module to model the affinity between region feature and pixel feature, where the region feature is computed from average pooling on square patch. In comparison with non-local based method and adaptive context module, our method models the context between nodes at different levels of the graph hierarchy, which not only leads to lower overhead but also allow contextual information flow to irregular-shaped regions.

3. Deep Grouping Model (DGM)

The proposed DGM represents an image as a hierarchical graph (see Fig. 2). The L-level multiscale feature maps $\{\mathcal{F}^l \mid l = 1, \ldots, L\}$ are extracted from different layers' output of a CNN, where $\mathcal{F}^1$ has a large resolution with more low-level details and $\mathcal{F}^L$ is in the lowest resolution containing more high-level semantics [56]. Correspondingly, we denote graph feature at the l-th level as $\mathcal{G}^l = \langle \mathbf{V}^l, \mathbf{E}^l \rangle$, where $\mathbf{V}^l$ and $\mathbf{E}^l$ denote vertex features and adjacency matrix, respectively. First, we initialize the bottom level graph $\mathcal{G}^1 = \langle \mathbf{V}^1, \mathbf{E}^1 \rangle$ from pre-computed superpixel $\mathcal{S}$ and bottom level grid feature map $\mathcal{F}^1$. Concretely, vertex features come from superpixel pooling, *i.e*., each node takes the mean of the features in the corresponding superpixel region of the feature map (formal definition can be seen in supplementary material). Unweighted adjacency matrix $\mathbf{E}^1$ is defined from the region adjacency graph of the superpixel $\mathcal{S}$ [39], which is much sparser compared with fully-connected non-local operation [45, 55]. Notice that only $\mathbf{E}^1$ is unweighted adjacency matrix, while upper-level adjacency matrices $\mathbf{E}^l (l > 1)$ are weighted adjacency matrices (more details in Sec. 3.1).

Bottom-up process. The bottom-up process is aiming at transforming multi-resolution grid feature maps $\{\mathcal{F}^l \mid l = 1, \ldots, L\}$ to hierarchical graph representation $\{\mathcal{G}^l \mid l = 1, \ldots, L\}$ (see Fig. 2), where $\mathcal{G}^l$ not only dynamically composes information from lower level graph $\mathcal{G}^{l-1}$ (Fig. 2(a)), but also receives high-level semantics from feature map

$\mathcal{F}^l$ (Fig. 2(b)). To this end, the proposed *Expectation-Maximization Graph Pooling (EMGP)* module and *Projection* module do the aforementioned jobs, respectively.

Top-down process. From another perspective, high-level semantics can also help low-level representation. To this end, *Top-down Message Passing* (*TDMP*) module propagates messages from the top-level graph to the bottom-level graph (Fig. 2(c)).

Finally, in order to make DGM compatible with modern CNN framework, we use a *Re-projection* module to transform hierarchical graph $\{\mathcal{G}^l\}$ back to multi-level grid-shape feature map $\{\hat{\mathcal{F}}^l \mid l = 1, \dots, L\}$ (Fig. 2(d)), which will be used in down-stream tasks.

3.1. Bottom-up Graph Hierarchy Construction

The bottom-up process transforms $\{\mathcal{F}^l\}$ to multi-level graph features $\{\mathcal{G}^l = \langle \mathbf{V}^l, \mathbf{E}^l \rangle \mid l = 1, \dots, L\}$ from the bottom level to the top level (*i.e.*, l is in an increasing order when constructing the graph hierarchy). Concretely, in order to construct $\mathcal{G}^{l+1}$ from $\mathcal{G}^l$, the modules *EMGP* and *Projection* run successively.

Expectation-Maximization Graph Pooling (EMGP). The goal of *EMGP* is to pool graph $\mathcal{G}^l$ to $\mathcal{G}^{l+1}$ with less number of nodes, *i.e.*, $|\mathbf{V}^{l+1}| < |\mathbf{V}^l|$ (see Fig. 2(a)). Following the EM framework [10], we initialize $\bar{\mathbf{V}}^{l+1}$ with uniformly sampled vertices from $\mathbf{V}^l$, then update pooled graph vertex features $\bar{\mathbf{V}}^{l+1}$ in K iterations:

$$\mathbf{P}^l_{ij} = \frac{1}{\mathbf{Z}^l_j} \exp\left(-\frac{||\mathbf{V}^l_i - \bar{\mathbf{V}}^{l+1}_j||^2}{\sigma^2}\right) , \tag{1}$$

$$\bar{\mathbf{V}}^{l+1} = (\mathbf{P}^l)^\intercal \mathbf{V}^l , \tag{2}$$

where $\mathbf{P}^l \in \mathbb{R}^{|\mathbf{V}^l| \times |\bar{\mathbf{V}}^{l+1}|}$ computes the affinity of vertices between the levels l and $l+1$ via a Gaussian kernel with bandwidth σ and $\mathbf{Z}^l \in \mathbb{R}^{|\bar{\mathbf{V}}^{l+1}|}$ is a normalization term:

$$\mathbf{Z}^l_j = \sum_i^{|\mathbf{V}^l|} \exp\left(-\frac{||\mathbf{V}^l_i - \bar{\mathbf{V}}^{l+1}_j||^2}{\sigma^2}\right) . \tag{3}$$

After K-iteration updates of vertex features, following Ying *et al*. [54], the adjacency matrix of higher level graph $\mathbf{E}^{l+1}$ can be computed by:

$$\mathbf{E}^{l+1} = (\mathbf{P}^l)^\intercal \mathbf{E}^l \mathbf{P}^l . \tag{4}$$

Notice that our method is different from the "differentiable pooling" method proposed in [54]. Instead of predicting pooling weights $\mathbf{P}^l$ through a stack of graph convolutional layers, our method uses EM to make the prediction. Therefore, our method not only considers similarity between each pair of nodes, but also can change $|\mathbf{V}^{l+1}|$ dynamically according to the content of the image. For example, an image of a simple scene with small number of objects or uniform textual, *e.g.*, the *sky*, can be represented by a small $|\mathbf{V}^{l+1}|$ in the graph.

Projection. Although the pooled node features $\bar{\mathbf{V}}^{l+1}$ summarize the lower level graph through a linear combination of the lower level graph nodes $\mathbf{V}^l$, they do not necessarily contain higher level semantics. To incorporate higher level semantics, the *Projection* module projects the feature map $\mathcal{F}^{l+1}$ to pooled node features $\bar{\mathbf{V}}^{l+1}$, outputting node feature $\mathbf{V}^{l+1}$.

A straightforward design could be constructing a bipartite graph between $\mathcal{F}^{l+1}$ and $\bar{\mathbf{V}}^{l+1}$ and use graph convolution to propagate high-level semantics, where pixels on the feature map $\mathcal{F}^{l+1}$ are treated as nodes and directed edges are pointing from $\mathcal{F}^{l+1}$ to $\bar{\mathbf{V}}^{l+1}$. However, such design not only creates large overhead due to large number of pixels on the feature map, but also the edge weights of the bipartite graph is undefined. Therefore, we define auxiliary nodes $\mathbf{U}^{l+1}$, obtained from superpixel pooling on feature map $\mathcal{F}^{l+1}$ by the bottom-level superpixel map $\mathcal{S}$, to address the aforementioned problems. Since both $\mathbf{U}^{l+1}$ and $\mathbf{V}^1$ are computed from the same superpixel map $\mathcal{S}$, $\mathbf{U}^{l+1}$ has the same number of vertices as $\mathbf{V}^1$, *i.e.*, $|\mathbf{U}^{l+1}| = |\mathbf{V}^1|$. However, $\mathbf{U}^{l+1}$ contains high-level semantics as it is pooled from the feature map $\mathcal{F}^{l+1}$.

A quasi-bipartite graph from $\mathbf{U}^{l+1}$ to $\bar{\mathbf{V}}^{l+1}$ can be constructed. Since $\mathbf{U}^{l+1}$ can also be hierarchically grouped to $\mathbf{V}^{l+1}$ as how $\mathbf{V}^1$ are merged to $\mathbf{V}^{l+1}$, we reuse $\{\mathbf{P}^l\}$ predicted by *EMGP* to construct the adjacency matrix of the quasi-bipartite directed graph. Concretely, we compute the cumulative product $\prod_{k=1}^{l} \mathbf{P}^k \in \mathbb{R}^{|\mathbf{V}^1| \times |\mathbf{V}^{l+1}|}$, which can be regarded as graph pooling weights that directly pool $\mathbf{V}^1$ (or the auxiliary nodes $\mathbf{U}^{l+1}$) to $\mathbf{V}^{l+1}$. To enable vertices $\bar{\mathbf{V}}^{l+1}$ retain the information through *EMGP*, self-loops are added to $\bar{\mathbf{V}}^{l+1}$, resulting in the final adjacency matrix $\mathbf{I} + \prod_{k=1}^{l} \mathbf{P}^k$ of the bipartite graph. Therefore, the bipartite graph is formally defined as $\left(\mathbf{U}^{l+1}, \bar{\mathbf{V}}^{l+1}, \mathbf{I} + \prod_{k=1}^{l} \mathbf{P}^k\right)$, where directed edges are pointing from $\mathbf{U}^{l+1}$ to $\bar{\mathbf{V}}^{l+1}$.

Next, we use graph convolution to allow message passing from $\mathbf{U}^{l+1}$ to $\mathbf{V}^{l+1}$:

$$\mathbf{V}^{l+1} = \mathit{GConv}\left(\mathbf{U}^{l+1} \cup \bar{\mathbf{V}}^{l+1}, \mathbf{I} + \prod_{k=1}^{l} \mathbf{P}^k\right) , \tag{5}$$

where *GConv* stands for graph convolution. Following the mean aggregator proposed in GraphSAGE [18], we use weighted average aggregator GraphSAGE as the graph convolution layer:

$$\mathbf{h}_v = \sigma\left(\mathbf{W} \cdot \sum_{u \in \mathcal{N}(v)} \mathbf{w}(u, v) \cdot \mathbf{h}_u\right) , \tag{6}$$

where $\mathbf{h}_v$ stands for the feature of vertex v, σ is the sigmoid function, $\mathbf{W}$ is a learnable weight matrix, and $\mathcal{N}(v)$ defines the neighboring nodes of vertex v. Here, $\mathbf{w}(u,v)$ is the weight of the directed edge from u to v, which can be found in the given adjacency matrix (*i.e.*, $\mathbf{I}+\prod_{k=1}^{l}\mathbf{P}^k$ as in Eq. 5). Thus, the updated graph node features $\mathbf{V}^{l+1}$ contain features of both high-level semantics $\mathcal{F}^{l+1}$ and the feature summarization from its lower level graph $\mathcal{G}^l$.

Global Vector. After the construction of $\mathcal{G}^L$, we obtain the global vector representation $\mathbf{R}$ (see the top node in Fig. 2) of the scene by:

$$\mathbf{R} = \mathit{READOUT}\big(\mathbf{V}^L\big) \ , \tag{7}$$

where *READOUT* function is used for combining features of a graph in many GNN methods [54, 43]. Here we use average pooling as the *READOUT* function. In other words, $\mathbf{R} = \frac{1}{|\mathbf{V}^L|}\sum_i^{|\mathbf{V}^L|}\mathbf{V}_i^L$. $\mathbf{R}$ can also be regarded as a graph at level $L+1$ without edges, *i.e.*, $\mathbf{R} = \mathcal{G}^{L+1} = \langle \mathbf{V}^{L+1}, \emptyset \rangle$. Since $\mathbf{R}$ is a vector representation of the image, it can be supervised by image classification tasks, *e.g.*, a scene category label for the image.

3.2. Top-down Message Passing (TDMP)

To further enable high-level semantics to help low-level features, the *TDMP* module iteratively updates each level of graph features from the top-level graph $\mathbf{R} = \mathcal{G}^{L+1}$ to the bottom level graph $\mathcal{G}^1$ through message passing, outputting updated multi-level graph features. It serves much like the "decomposition" process as motivated in Introduction.

Concretely, given $\mathbf{V}^{l+1}$ (already updated) and $\mathbf{V}^l$ (to be updated), a quasi-bipartite graph is constructed (see Fig. 2(c)), where directed edges are pointing from $\mathbf{V}^{l+1}$ to $\mathbf{V}^l$. Intuitively, high-level semantics should be transmitted to their corresponding lower-level regions. For example, the whole human body feature at the $(l+1)$-th level should be sent to human parts (*e.g.*, arms, legs) at the l-th level . Thus, by reusing the grouping results in the bottom-up process, edges $\tilde{\mathbf{P}}^l \in \mathbb{R}^{|\mathbf{V}^l|\times|\mathbf{V}^{l+1}|}$ can be obtained by:

$$\tilde{\mathbf{P}}^l_{ij} = \frac{1}{\tilde{\mathbf{Z}}^l_i}\exp\big(-\frac{||\mathbf{V}^l_i - \mathbf{V}^{l+1}_j||^2}{\sigma^2}\big) \ , \tag{8}$$

where $\tilde{\mathbf{Z}}^l \in \mathbb{R}^{|\mathbf{V}^l|}$ is a normalization term:

$$\tilde{\mathbf{Z}}^l_i = \sum_j^{|\mathbf{V}^{l+1}|}\exp\big(-\frac{||\mathbf{V}^l_i - \mathbf{V}^{l+1}_j||^2}{\sigma^2}\big) \ . \tag{9}$$

After adding self-loops to $\mathbf{V}^l$, a graph convolution layer is applied to achieve the top-down message passing:

$$\mathbf{V}^l := \mathit{GConv}\big(\mathbf{V}^{l+1}\cup\mathbf{V}^l, \mathbf{I}+\tilde{\mathbf{P}}^l\big) \ , \tag{10}$$

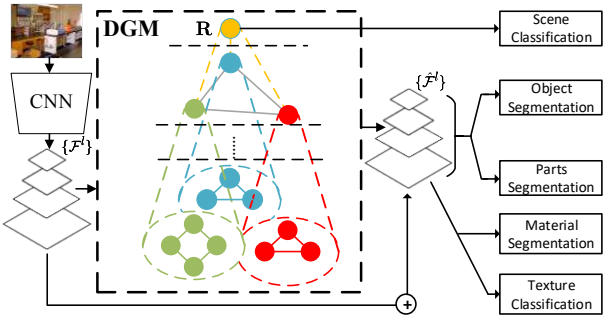

Figure 3: Full Model for the Unified Perceptual Parsing Task.

where $\mathbf{V}^l$ is the updated vertex feature at the lth level and *GConv* is defined the same as in Eq. 6.

3.3. Re-projection from Graph to Grid Features

Finally, we re-project the updated vertex features $\{\mathbf{V}^l\}$ back to the grid features resulting in $\{\hat{\mathcal{F}}^l\}$. The *re-prejection* can be regarded as a mirror module of *projection*. Analogous to the *projection* module, at each level l, a quasi-bipartite directed graph $\big(\mathbf{V}^l, \mathbf{U}^l, \mathbf{I}+\prod_{k=1}^{l}\tilde{\mathbf{P}}^k\big)$ (see Fig. 2(d)) is built from superpixel pooling features $\mathbf{U}^l$, updated vertex features $\mathbf{V}^l$, and the adjacency matrix that comes from the self-loops of $\mathbf{U}^l$ and the cumulative product $\prod_{k=1}^{l}\tilde{\mathbf{P}}^k$. Here, edges are pointing from $\mathbf{V}^l$ to $\mathbf{U}^l$. Then, we apply graph convolution to re-project the features:

$$\hat{\mathbf{U}}^l = \mathit{GConv}\big(\mathbf{V}^l\cup\mathbf{U}^l, \mathbf{I}+\prod_{k=1}^{l}\tilde{\mathbf{P}}^k\big) \ , \tag{11}$$

where $\hat{\mathbf{U}}^l$ is the vertex feature receiving information from the graph and has the same number of superpixels defined in superpixel map $\mathcal{S}$. Lastly, $\hat{\mathbf{U}}^l$ is copied to pixel regions defined by the superpixel map $\mathcal{S}$, outputting the updated grid feature map $\hat{\mathcal{F}}^l$.

4. Unified Perceptual Parsing with DGM

To fully verify the effectiveness of the hierarchical graph representation, we apply deep grouping model (DGM) on the unified perceptual parsing (UPP) task, a challenging task introduced by Xiao *et al*. [48]. Aiming at recognizing heterogeneous perceptual concepts of an image, UPP combines tasks of scene classification, object segmentation, parts segmentation, material segmentation, and texture recognition, requiring good modeling on features at different granularities.

To this end, we insert DGM to a backbone model (see Fig. 3), which outputs $\{\hat{\mathcal{F}}^l \,|\, l = 1,\ldots,L\}$. With the residual connection [20] from $\{\mathcal{F}^l \mid l = 1...L\}$, we obtain multi-resolution grid feature maps $\{\mathcal{F}^l + \hat{\mathcal{F}}^l \mid l = 1,\ldots,L\}$. Fol-

lowing the architecture proposed by Xiao *et al*. [48], after bilinear interpolating all feature maps to the same size, we concatenate all L levels of grid features $\{\mathcal{F}^l + \hat{\mathcal{F}}^l | l = 1, \ldots, L\}$ for object segmentation and part segmentation. In material segmentation, we only use the bottom-level grid feature $\mathcal{F}^1 + \hat{\mathcal{F}}^1$ for prediction by following the architecture of UPerNet [48].

For scene classification, we first apply global average pooling on the original top-level feature map $\mathcal{F}^L$ (not shown in the figure). Then, it is residual connected with the graph *READOUT* feature $\mathbf{R}$ for scene classification.

Limited by the dataset in UPP task [2], only texture images with image-level labels are provided. Therefore, for texture recognition, the model classifies texture images with the feature come from global average pooling on the bottom grid features $\mathcal{F}^1 + \hat{\mathcal{F}}^1$ in training and quantitative evaluation. However, we can also apply the texture classification layer on each pixel to generate texture segmentation results on natural images.

To summarize, the final loss of the full model on the unified perceptual parsing task is defined by:

$$\mathcal{L} = \lambda_s \mathcal{L}_s + \lambda_t \mathcal{L}_t + \lambda_o \mathcal{L}_o + \lambda_p \mathcal{L}_p + \lambda_m \mathcal{L}_m, \qquad (12)$$

where $\mathcal{L}_s$ and $\mathcal{L}_t$ are cross-entropy losses between prediction and image labels for scene classification and texture classification, respectively. $\mathcal{L}_o$, $\mathcal{L}_p$, $\mathcal{L}_m$ are cross-entropy losses at each pixel between the prediction and ground-truth for object segmentation, part segmentation, and material segmentation, respectively. Following [48], coefficients of each loss term are $\lambda_s = 0.25, \lambda_t = 1, \lambda_o = 1, \lambda_p = 0.5, \lambda_m = 1$.

5. Experiments

5.1. Dataset and Evaluation Metrics

The Broden+ dataset [2] is used for training and evaluation on the unified perceptual parsing task [48]. The dataset is comprised of five large datasets: ADE20k [60], PASCAL Context [32], PASCAL-Part [7], OpenSurfaces [3], and DTD [9] datasets. For each subtask in unified perceptual parsing, the data source comes from the union of the datasets that contain subtask's labels. For example, object/stuff segmentation task will be trained on and evaluated on the union of ADE20k and PASCAL-Context datasets. In this way, not only the number of tasks is large, but also the number of categories is larger since datasets are merged together, which makes the unified perceptual parsing task extremely challenging. In terms of evaluation metrics, the scene classification task and texture classification task are evaluated via top-1 accuracy (Top-1 Acc.). The object/stuff segmentation, parts segmentation and material segmentation are evaluated by mIoU and pixel accuracy (P.A.).

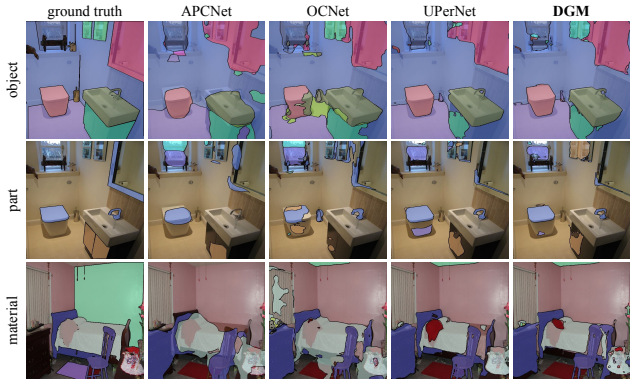

Figure 4: Qualitative comparison on Broden+ Dataset.

5.2. Implementation Details

We follow the experimental settings in [48]. During training, we resize the image's shorter side to a size that is randomly chosen from $300, 375, 450, 525, 600$ and keep its aspect ratio. The shorter side of the image is resized to 450 pixels in the evaluation stage. Following [5], we use "poly" learning rate policy $(1 - \frac{iter}{max\_iter}^{power})$ to adjust learning rate during training, and the initial learning rate is 0.02, where $max\_iter = 2 \times 10^5$ and $power = 0.9$. The batch size is 8, and the model is trained on 4 GPUs.

MCG [35, 1] is used for extracting superpixels for training DGM, which is further merged greedily to make sure that the number of superpixel is at most 512. In terms of the DGM architecture, the input multi-resolution feature map $\mathcal{F}^l$ comes from C_1 to C_4 layers' output from ResNet [20]. Accordingly, we set the level of graph $L = 4$ in our experiment. All GraphSAGE [18] layers in DGM are followed by L_2 normalization and ReLU [33]. The *EMGP* module pools the graph to half the number of nodes in upper-level graph (*i.e.*, $|\bar{\mathbf{V}}^{l+1}| = |\mathbf{V}^l|/2$). The number of iteration K in *EMGP* is set as 5 in training and 10 in evaluation.

Our code is based on the PyTorch framework [34]. Specifically, the PyTorch Geometric [14] is used to implement graph operations in DGM. Following UPerNet [48], in the experiment on Broden+ dataset, all tasks except the texture classification task are trained jointly. When training the texture classification task, the model's parameters are fixed except the texture classification branch.

5.3. Comparison with the State-of-the-art

Results of all tasks in the unified perceptual parsing (UPP) task are shown in Tab. 1. Since the dataset is fairly recent and only UPerNet [48] reports its results, we replicate OCNet [55] and APCNet [19]'s results[1] on the Broden+ dataset, as they represent state-of-the-art contextual modeling methods based on non-local block and region-based context mod-

[1]To ensure a fair comparison, we used the authors' released code for OCNet. Since no released code for APCNet, we did our best to replicate.

Tasks	Method	Object		Part		Scene	Material		Texture
		mIoU	P.A.	mIoU	P.A.	Top-1 Acc.	mIoU	P.A.	Top-1 Acc.
O+P+S	APCNet	21.25	71.71	23.39	41.07	68.50	-	-	-
	OCNet	22.62	74.58	28.51	48.92	68.50	-	-	-
	UPerNet	23.83	**77.23**	30.10	48.34	71.35	-	-	-
	DGM w/o ↓	24.58	74.76	31.23	**51.17**	71.24	-	-	-
	DGM	**24.76**	75.15	**31.26**	50.55	**71.87**	-	-	-
O+P+S+M+T	APCNet	20.37	71.01	22.32	40.08	68.45	43.88	79.95	50.35
	OCNet	20.21	**77.09**	25.75	43.78	66.92	48.20	80.70	51.95
	UPerNet	23.36	**77.09**	28.75	46.92	70.87	54.19	84.45	57.44*
	DGM w/o ↓	24.05	74.21	29.94	49.49	70.24	54.52	84.41	58.15
	DGM	**24.37**	74.99	**30.28**	**49.70**	**71.03**	**54.58**	**84.62**	**60.10**

Table 1: Comparing with state-of-the-art methods on Broden+ dataset. O+P+S means object segmentation task, part segmentation task, and scene classification task are used in training and evaluation. O+P+S+M+T incrementally add material segmentation task and texture classification task in training and evaluation stages. *Based on the authors' released model, we continue to train UPerNet and get better results on texture classification than the reported number (35.10) in [48].

Method	mIoU	Pixel Accuracy
UPerNet	42.66	81.01
+DGM w/o ↓	**43.64**	81.11
+DGM	43.51	**81.13**
HRNetv2	43.20	81.47
+DGM w/o ↓	**43.86**	**81.55**
+DGM	43.46	81.53
DeepLabV3	44.1	81.1
+DGM w/o ↓	44.31	**81.36**
+DGM	**44.86**	81.35
CCNet [22]	45.22	-
APCNet [19]	45.38	-
OCNet [55]	45.45	-

Table 2: Results on ADE20k validation set.

eling in semantic segmentation, respectively. The backbone of UPerNet, OCNet and our proposed DGM is ResNet50, and APCNet's backbone is the dilated ResNet50 [48, 19, 55]. Backbones' weights are initialized with ImageNet [11] pre-trained models. More results comparing with GCU [25] and HRNetv2 [44] backbone are included in Appendix.

Results shows that our model (**DGM** in Tab. 1) outperforms all other methods, achieving the state-of-the-art result on Broden+ in every subtask. Although DGM did not achieve the best performance in terms of pixel accuracy on the object segmentation subtask, we suspect that the pixel accuracy measure is easily biased by the imbalanced number of pixels among different classes, while mIoU is a better and more meaningful evaluation metric for segmentation.

In the qualitative evaluation, our model can achieve more reasonable results. For example, in Fig. 4, compared with other methods, our model successfully segments both cabinet (in green) and toilet (in pink) in object segmentation. Our model's parts segmentation has smaller false prediction on the toilet. Finally, on the material segmentation, our model shows sharp boundary on the legs of wood chair.

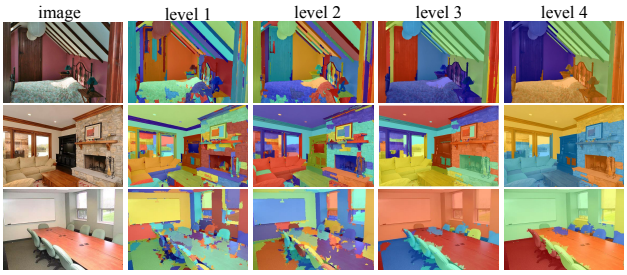

Figure 5: Visualization of perceptual groupings generated by DGM. A color represents a graph vertex. Note that the same color between different levels are not related.

5.4. Ablation Study

Single-task training. To ablate the effect of multi-task training, we train our model on ADE20k only focusing on the semantic segmentation task. We use three backbone models to train and evaluate our model: UPerNet, HRNetv2 [44], and DeepLabV3 [6]. Our DGM is general enough to be an add-on module for many segmentation networks. More details of how DGM is added will be illustrated in the supplementary material. Results in Tab. 2 (see **+DGM**) show that DGM can increase the performance for every backbone model. Admittedly, OCNet and APCNet show better performance on ADE20k. Our model serves a better role in the more challenging unified perceptual parsing where a joint representation for multiple tasks is needed.

Top-down message passing. To evaluate the role of *TDMP*, we evaluate DGM model without *TDMP* (denoted as **DGM w/o ↓**). In the Broden+ dataset, DGM w/o ↓ shows weaker performance compared with the full model, proving the effectiveness of context modeling of *TDMP*. In the single-

Model	FLOPs (Δ)	#Params (Δ)
OCP	161.4G	15.179M
RCCA(R=2)	16.5G	23.931M
DGM w/o↓	**9.3G**	**3.417M**
DGM	**10.8G**	**4.468M**

Table 3: Compare overhead of contextual modules.

task ADE20k, DGM w/o ↓ performs weaker on DeepLabV3 backbone and achieves even better performance than the full model when UPerNet and HRNetv2 are the backbones. We suspect that the top-down message passing may not provide valuable information to lower-level graph features when only one task is trained and evaluated. In comparison, *TDMP* helps lower-level graph features for better prediction on part segmentation and material segmentation (see Tab. 1).

5.5. Grouping Visualization

To verify the quality of perceptual grouping, the grouping results are visualized in Fig. 5. Details of grouping visualization will be illustrated in the supplementary material. As shown in Fig. 5, DGM gradually merges conceptually-related regions as it goes to higher levels in the hierarchy. For example, in the second row, sofa gradually merges with tables to the main area in the living room.

5.6. Overhead

We compare the overhead with other contextual modeling methods: recurrent criss-cross attention (RCCA) module proposed in CCNet [22] and object context pooling module (OCP) in OCNet [55]. For a fair comparison, the size of the input images to all methods is 769 × 769. In Tab. 3, we show the difference of FLOPs and the number of parameters before and after adding the contextual modeling module to the network. For our proposed DGM model, we use ResNet50 [48] as the backbone when evaluating the overhead. The results show that our model has significantly lower overhead compared with non-local base OCP module. Note that RCCA is the state-of-art method targeting at reducing overhead in contextual modeling. Our method beats RCCA module of CCNet. In Tab. 3, we also show the overhead of our method without using *TDMP* (DGM w/o ↓). The result shows that *TDMP* only creates little overhead.

6. Applications

We further show that DGM enables novel applications due to the added interpretability of the perceptual grouping process, which is difficult to achieve by using other segmentation networks.

Click Propagation. In interactive segmentation, a user adds positive click on the object and negative click on the background, which are used to segment the selected instance on the image. One critical process of recent interactive segmentation methods [30, 52] is augmenting user's click by propagating it to other related areas on the image. Since our model produces a compositional-hierarchical graph, related areas can be dynamically computed through the learning process, rather than treating it as a pre-processing step. As shown in Fig. 6, given a user's click, our model first selects a superpixel. Then, it can propagate to higher levels by using the $\mathbf{P}^l$ defined in Eq. 1. For example, positive click is propagated to the entire shower kit in Fig. 6 top-left, and negative click will not be propagated to the bathtub in Fig. 6 bottom-right. More details are in the supplementary material.

Figure 6: Visualization of click propagation. Bottom-right is negative click while others are all positive clicks.

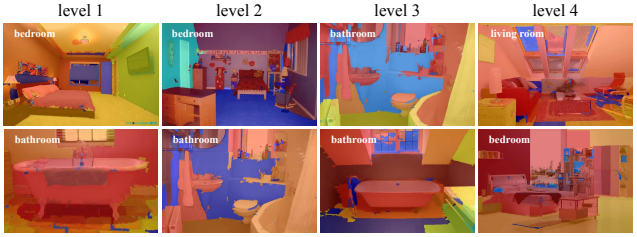

Figure 7: Visualization of Grad-CAM. Red-to-blue denotes the decreasing activation. Scene labels are shown.

Explainability with Grad-CAM. We use Grad-CAM on graph [36] to localize activated vertices at each level of the hierarchy (more details in supplementary material). By using the gradient back-propagated from the ground-truth scene label, our model localizes semantically discriminative regions on the image. For example, the bed is highlighted with sharp boundary in Fig. 7 bedroom.

7. Conclusion

We propose Deep Grouping Model to marry a CNN segmentation network with the perceptual grouping process, which outperforms state-of-the-art methods on unified perceptual parsing task with little overhead. Meanwhile, our proposed model is of good interpretability and is useful in other tasks. We believe such hierarchical graph representation is of great potential to be applied to many other tasks.

Acknowledgments. This work was supported in part by NSF 1741472, 1764415, 1813709, and 1909912. The article solely reflects the opinions and conclusions of its authors but not the funding agents.

References

[1] Pablo Arbelaez, Jordi Pont-Tuset, Jonathan T. Barron, Ferran Marques, and Jitendra Malik. Multiscale combinatorial grouping. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 2, 6

[2] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2, 6

[3] Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. Opensurfaces: A richly annotated catalog of surface appearance. *ACM Trans. Graph.*, 32(4):111:1–111:17, July 2013. 6

[4] Timothy F Brady, Anna Shafer-Skelton, and George A Alvarez. Global ensemble texture representations are critical to rapid scene perception. *Journal of Experimental Psychology: Human Perception and Performance*, 43(6):1160, 2017. 1

[5] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, April 2018. 1, 6

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

[7] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 6

[8] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Yan Shuicheng, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3

[9] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 6

[10] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977. 4

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2009. 7

[12] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1915–1929, 2012. 2

[13] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181, 2004. 1

[14] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019. 6

[15] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3

[16] Jun Fu, Jing Liu, Yuhang Wang, Yong Li, Yongjun Bao, Jinhui Tang, and Hanqing Lu. Adaptive context network for scene parsing. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 1

[17] Raghudeep Gadde, Varun Jampani, Martin Kiefel, Daniel Kappler, and Peter V. Gehler. Superpixel convolutional networks using bilateral inceptions. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 597–613, Cham, 2016. Springer International Publishing. 2

[18] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 1024–1034. Curran Associates, Inc., 2017. 3, 4, 6

[19] Junjun He, Zhongying Deng, Lei Zhou, Yali Wang, and Yu Qiao. Adaptive pyramid context network for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 3, 6, 7

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 5, 6

[21] Han Hu, Zheng Zhang, Zhenda Xie, and Stephen Lin. Local relation networks for image recognition. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 2

[22] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 7, 8

[23] Varun Jampani, Deqing Sun, Ming-Yu Liu, Ming-Hsuan Yang, and Jan Kautz. Superpixel sampling networks. In *The European Conference on Computer Vision (ECCV)*, September 2018. 2

[24] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2

[25] Yin Li and Abhinav Gupta. Beyond grids: Learning graph representations for visual recognition. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 9225–9235. Curran Associates, Inc., 2018. 3, 7

[26] 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 Infor-*

mation Processing Systems 31, pages 1853–1863. Curran Associates, Inc., 2018. 2

[27] Xiaodan Liang, Liang Lin, Xiaohui Shen, Jiashi Feng, Shuicheng Yan, and Eric P. Xing. Interpretable structure-evolving lstm. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[28] Xiaodan Liang, Xiaohui Shen, Jiashi Feng, Liang Lin, and Shuicheng Yan. Semantic object parsing with graph lstm. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 125–143, Cham, 2016. Springer International Publishing. 2

[29] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 1

[30] Soumajit Majumder and Angela Yao. Content-aware multi-level guidance for interactive instance segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2, 8

[31] Alastair P Moore, Simon JD Prince, Jonathan Warrell, Umar Mohammed, and Graham Jones. Superpixel lattices. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 1

[32] 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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 6

[33] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010. 6

[34] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017. 6

[35] J. Pont-Tuset, P. Arbeláez, J. T. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping for image segmentation and object proposal generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(1):128–140, Jan 2017. 1, 2, 6

[36] Phillip E. Pope, Soheil Kolouri, Mohammad Rostami, Charles E. Martin, and Heiko Hoffmann. Explainability methods for graph convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 8

[37] Ren and Malik. Learning a classification model for segmentation. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 10–17 vol.1, Oct 2003. 1, 2

[38] Joseph Tighe and Svetlana Lazebnik. Superparsing. *International Journal of Computer Vision*, 101(2):329–349, 2013. 1

[39] A. Tremeau and P. Colantoni. Regions adjacency graph applied to color image segmentation. *IEEE Transactions on Image Processing*, 9(4):735–744, April 2000. 3

[40] Wei-Chih Tu, Ming-Yu Liu, Varun Jampani, Deqing Sun, Shao-Yi Chien, Ming-Hsuan Yang, and Jan Kautz. Learning superpixels with segmentation-aware affinity loss. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[41] Zhuowen Tu, Xiangrong Chen, Alan L Yuille, and Song-Chun Zhu. Image parsing: Unifying segmentation, detection, and recognition. *International Journal of computer vision*, 63(2):113–140, 2005. 1

[42] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171, Sep 2013. 2

[43] Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. In *International Conference on Learning Representations*, 2019. 5

[44] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, March 2020. 1, 7

[45] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2, 3

[46] Xiang Wang, Shaodi You, Xi Li, and Huimin Ma. Weakly-supervised semantic segmentation by iteratively mining common object features. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[47] Yunchao Wei, Jiashi Feng, Xiaodan Liang, Ming-Ming Cheng, Yao Zhao, and Shuicheng Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[48] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *The European Conference on Computer Vision (ECCV)*, September 2018. 1, 2, 5, 6, 7, 8

[49] Chenliang Xu and Jason J. Corso. Actor-action semantic segmentation with grouping process models. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[50] Chenliang Xu, Spencer Whitt, and Jason J. Corso. Flattening supervoxel hierarchies by the uniform entropy slice. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2013. 2

[51] Chenliang Xu, Caiming Xiong, and Jason J. Corso. Streaming hierarchical video segmentation. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, pages 626–639, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. 2

[52] Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas S. Huang. Deep interactive object selection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2, 8

[53] Jian Yao, Sanja Fidler, and Raquel Urtasun. Describing the scene as a whole: Joint object detection, scene classification

and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 1

[54] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. In *Advances in Neural Information Processing Systems*, pages 4800–4810, 2018. 3, 4, 5

[55] Yuhui Yuan and Jingdong Wang. Ocnet: Object context network for scene parsing. *arXiv preprint arXiv:1809.00916*, 2018. 1, 2, 3, 6, 7, 8

[56] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 818–833, Cham, 2014. Springer International Publishing. 3

[57] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1

[58] Hang Zhang, Han Zhang, Chenguang Wang, and Junyuan Xie. Co-occurrent features in semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1

[59] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 2

[60] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127(3):302–321, Mar 2019. 6