# Group-Wise Semantic Mining for Weakly Supervised Semantic Segmentation

**Xueyi Li[1], Tianfei Zhou[2*], Jianwu Li[1], Yi Zhou[3], Zhaoxiang Zhang[4]**

[1] Beijing Key Laboratory of Intelligent Information Technology,
School of Computer Science and Technology, Beijing Institute of Technology, China
[2] Computer Vision Laboratory, ETH Zurich, Switzerland
[3] School of Computer Science and Engineering, Southeast University, China
[4] Center for Research on Intelligent Perception and Computing, CASIA, China
xueyili@bit.edu.cn    tianfei.zhou@vision.ee.ethz.ch    ljw@bit.edu.cn

## Abstract

Acquiring sufficient ground-truth supervision to train deep visual models has been a bottleneck over the years due to the data-hungry nature of deep learning. This is exacerbated in some structured prediction tasks, such as semantic segmentation, which requires pixel-level annotations. This work addresses weakly supervised semantic segmentation (WSSS), with the goal of bridging the gap between image-level annotations and pixel-level segmentation. We formulate WSSS as a novel group-wise learning task that explicitly models semantic dependencies in a group of images to estimate more reliable pseudo ground-truths, which can be used for training more accurate segmentation models. In particular, we devise a graph neural network (GNN) for group-wise semantic mining, wherein input images are represented as graph nodes, and the underlying relations between a pair of images are characterized by an efficient co-attention mechanism. Moreover, in order to prevent the model from paying excessive attention to common semantics only, we further propose a graph dropout layer, encouraging the model to learn more accurate and complete object responses. The whole network is end-to-end trainable by iterative message passing, which propagates interaction cues over the images to progressively improve the performance. We conduct experiments on the popular PASCAL VOC 2012 and COCO benchmarks, and our model yields state-of-the-art performance. Our code is available at: *https://github.com/Lixy1997/Group-WSSS*.

## Introduction

Semantic segmentation is a fundamental task in computer vision, aiming to assign a semantic category to each pixel in an image. It can benefit a wide variety of applications including autonomous driving, image editing and medical diagnosis. With the recent renaissance of deep neural networks, semantic segmentation has achieved tremendous progress. However, most of the leading approaches (Long, Shelhamer, and Darrell 2015; Wang et al. 2019b; Zhou et al. 2020; Wang et al. 2021b) are fully supervised, requiring massive amounts of pixel-level annotated training data, which are extremely expensive and time-consuming to obtain. In contrast, the weak supervision alternatives, *e.g.,* image-level tags (Pathak, Krahenbuhl, and Darrell 2015; Kolesnikov and

Lampert 2016; Qi et al. 2016; Wei et al. 2016; Chaudhry, Dokania, and Torr 2017; Ahn and Kwak 2018; Fan et al. 2018), scribbles (Lin et al. 2016; Vernaza and Chandraker 2017) or bounding-box annotations (Dai, He, and Sun 2015; Khoreva et al. 2017; Song et al. 2019), are less costly. Thus, it is of interest to explore the potential of these weak supervision cues in providing a data-efficient solution for semantic segmentation. In this paper, we aim to address weakly supervised semantic segmentation (WSSS) under the supervision of image-level tags, which can be obtained effortlessly.

WSSS based on image tags is extremely challenging because fine-grained pixel-level annotations, which are typically required for semantic segmentation, are difficult to obtain from class labels. The pioneering work, (Zhou et al. 2016), proposes an efficient and straightforward way to solve this by recognizing the discriminative regions specific to a given category using class activation maps (CAMs), which are then refined to obtain pseudo ground-truths for supervising a semantic segmentation network. Along this line, a number of approaches have been proposed to improve the estimation of CAMs so that they cover the full extent of objects rather than only the most representative parts. For example, some approaches (Wei et al. 2017; Kolesnikov and Lampert 2016; Choe and Shim 2019) manipulate internal feature maps to guide the network to perceive easily ignored but essential parts, while others (Hou et al. 2018; Chang et al. 2020; Fan et al. 2020a; Wang et al. 2020b) adopt self-ensembling or self-supervision to improve localization.

However, the mainstream methods above are merely based on ***single images*** (Figure 1 (a)), ignoring the valuable semantic context existing in a group of images. The very recent studies (Fan et al. 2020b; Sun et al. 2020) utilized Siamese networks to model the relations between a pair of images, leading to a ***pair-wise*** solution (Figure 1 (b)). These approaches have proven effective in locating more accurate object regions. However, seeking relations between two images at a time is still limited in capturing substantial semantic context. Accordingly, we introduce a more promising, and fundamentally different ***group-wise*** solution (Figure 1 (c)) which comprehensively mines richer semantics from a group of images. Our main motivation is that the availability of group images containing instances of the *same semantic classes* can make up for the absence of detailed supervisory information. From this perspective, we hypothesize that it is
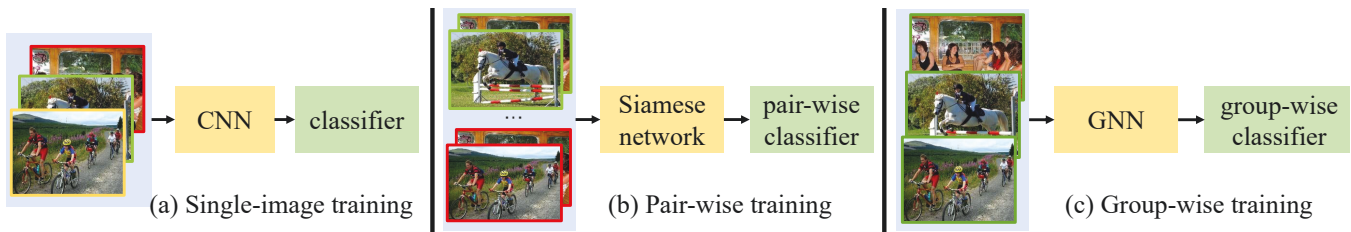
---

Figure 1: Architecture comparison of existing frameworks *vs.* Ours. (a) Single-image models feed each image one by one into the network for training, using standard classifiers (*e.g.,* VGG). (b) Pair-wise methods extract features from a pair of images using a Siamese network, and make predictions using a pair-wise classifier which has learned the correlation between the image pair. (c) We propose a group-wise method that accepts an arbitrary number of images as input. The input images are *iteratively* processed by a GNN to enable substantial information exchange, and a group-wise classifier is then adopted for prediction.

desirable to take advantage of all available information for WSSS, including not only individual image properties, but also group-level synergetic relationships.

Based on the above analysis, we propose a novel deep learning model for WSSS. Unlike previous pair-wise approaches, our model is targeted at group-wise semantic mining to capture more comprehensive relations among input images. Specifically, we develop an efficient, end-to-end trainable graph neural network (GNN), and conduct recursive reasoning for group-wise semantic understanding. In our graph, the nodes represent a group of input images, and edges describe pair-wise relations between two connected images. We consider two images as connected only if they share common semantic objects with each other, and their relation is then characterized by an elaborately designed co-attention mechanism. Through iterative message passing, the information from individual elements can be efficiently integrated and broadcasted over the graph structure. In this way, our model is capable of leveraging explicit semantic dependencies among images to obtain better node representations. However, this graph reasoning strategy mainly focuses on co-occurring semantics in a group of images, ignoring isolated objects. To address this, we further introduce a graph dropout layer, which can be seamlessly integrated into the GNN for iterative inference. The graph dropout layer selectively suppresses the most salient objects, forcing the network to be biased toward their counterparts.

Our method has two appealing characteristics over single-image and pair-wise methods. **First**, it is capable of learning semantic relations from an arbitrary number of images using a flexible GNN framework. The GNN empowers our model to inherit the complementary strengths of neural networks in learning capability and graphical models in structure representations. **Second**, our model adopts multi-step, iterative inference to progressively improve image representations. This is more favorable than directly producing representations by one-step inference in previous approaches

In summary, our main contributions are three-fold: **First**, we demonstrate the advantages of group-wise semantic mining for WSSS, and proffer a graph-aware solution for effective inference. **Second**, we develop a graph dropout layer to promote the missing categories, leading to more accurate localization. **Third**, we evaluate the proposed approach on

two large-scale benchmarks, *i.e.,* PASCAL VOC 2012 (Everingham et al. 2010) and COCO (Lin et al. 2014), and the results demonstrate its superiority.

## Related Work

**Weakly Supervised Semantic Segmentation.** Recent years have seen a surge of interest in semantic segmentation under weak supervision (*e.g.,* image-level labels, scribbles, bounding boxes), greatly reducing human efforts in manual labeling. In particular, methods operating with image-level labels have attracted the most attention since they require minimal annotation efforts. Most of these methods follow a popular pipeline that trains an image classifier using image-level labels, and exploits CAMs to highlight the most discriminative object regions for a particular semantic category as its pseudo ground-truth. However, CAMs are weak in revealing complete object regions, resulting in poor segmentation performance. Some pioneering efforts address this difficulty by learning pixel affinities (Ahn and Kwak 2018), erasing the most discriminative parts (Wei et al. 2017; Choe and Shim 2019; Lee et al. 2019), optimizing intraclass discrimination (Fan et al. 2020a), or applying region growing (Kolesnikov and Lampert 2016; Wei et al. 2018; Huang et al. 2018) to capture the full extent of objects. However, these methods are confined to using only limited image-level information. More recent approaches thus follow the self-supervised paradigm to acquire additional supervisions (Shimoda and Yanai 2019; Wang et al. 2020b), or rely on Siamese networks to capture semantic relations between a pair of images (Fan et al. 2020b; Sun et al. 2020).

In this paper, we take a further step toward discovering higher-order relations among images. A graph model is designed to encode such relationships. Through graph reasoning, our model iteratively refines object representations by accepting informative knowledge from other images.

**Graph Neural Networks.** Graph neural networks were proposed in (Scarselli et al. 2008), and have since gained widespread attention due to their superiority in dealing with flexible graph-structured data. GNNs typically model the graph elements (*e.g.,* nodes, edges) and approximation inference as learnable neural networks, and conduct iterative reasoning to explicitly discover the relations among nodes. They have achieved wide success in a variety of fields, in-
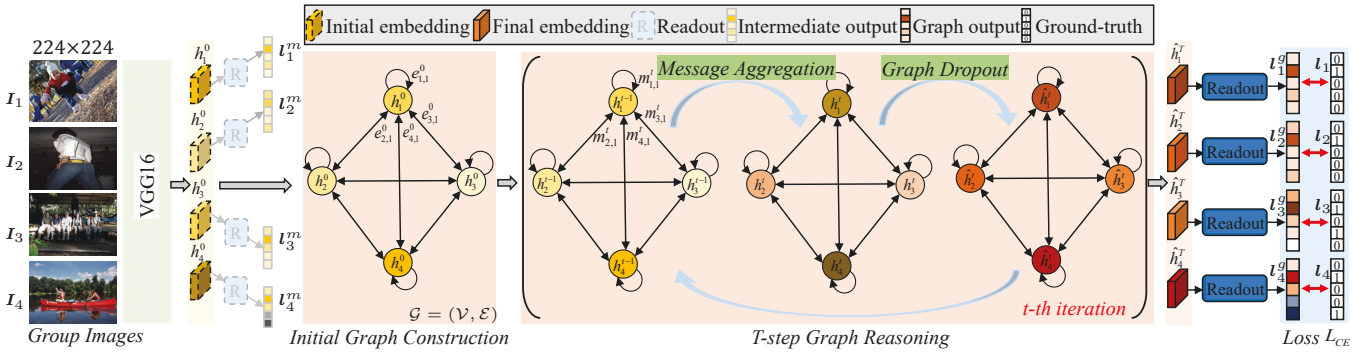
Figure 2: Overview of the proposed group-wise semantic mining network during the training phase. Given a group of images (*i.e.,* $\{\boldsymbol{I}_i\}_{i=1}^4$), our model uses VGG16 to extract convolutional features (*i.e.,* $\boldsymbol{h}_i^0\}_{i=1}^4$), which are used as the initial embeddings for graph construction. Next, our model conducts $T$-step graph reasoning to iteratively refine the features by *message passing* (Eq. (8)), *message aggregation* (Eq. (2)), and *graph dropout* (Eq. (10)). The final features (*i.e.,* $\{\hat{\boldsymbol{h}}_i^T\}_{i=1}^4$) are fed into a readout function (Eq. (9)) to get the predictions (*i.e.,* $\{\boldsymbol{l}_i^g\}_{i=1}^4$).

cluding molecular biology (Gilmer et al. 2017), computer vision (Qi et al. 2017; Lu et al. 2020; Marino, Salakhutdinov, and Gupta 2017; Wang et al. 2019a; Santoro et al. 2017; Zhou et al. 2021a; Wang et al. 2021a), medical image analysis (Zhou et al. 2021b) and machine learning (Veličković et al. 2018; Qu, Bengio, and Tang 2019). Inspired by these efforts, we build an image-level graph network to model their semantic relations for the WSSS task. Assisted by a graph dropout layer, our model can generate more accurate pseudo ground-truths for semantic segmentation.

## Methodology

In this section, we elaborate on the proposed model for WSSS. Given training images with only image-level labels, current efforts operate on two sub-tasks to achieve pixel-wise predictions. The first one is *pseudo ground-truth generation*, which relies on an image classification network to localize discriminative regions. The other one is *semantic segmentation*, which conducts dense predictions using a fully convolutional network (FCN) under the supervision of pseudo labels. Our approach also follows this pipeline. However, unlike previous approaches that treat each single image independently, our model aims to mine common semantic patterns from multiple images by graph inference. In this way, our model can alleviate the incomplete-annotation problem in WSSS and produce more accurate pseudo labels.

### Preliminary: Graph Neural Networks

We start by revisiting the basic concept of GNNs. We define a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ by its node set $\mathcal{V} = \{v_1, \ldots, v_n\}$ and edge set $\mathcal{E} = \{e_{i,j} = (v_i, v_j)|v_i, v_j \in \mathcal{V}\}$. We assume that each node $v_i$ is associated with a feature embedding vector $\boldsymbol{h}_i$, and each edge $e_{i,j}$ has an edge representation $\boldsymbol{e}_{i,j}$. During inference, GNNs iteratively improve the feature representations at a node by aggregating its neighborhood features. Specifically, a GNN maps the graph $\mathcal{G}$ to the node outputs through two phases: a message passing phase and a readout phase. The message passing phase is defined in terms of a *message function* $\mathcal{F}_M$, whose input is a node's features and

output is a message, and an *aggregation function* $\mathcal{F}_A$, whose input is a set of messages and output is the updated features. Suppose we conduct $T$ rounds of message passing; the $t$-th round for a node $v_i$ can be described as:

$$\text{message passing:} \quad \boldsymbol{m}_i^t = \sum_{v_j \in \mathcal{N}_i} \mathcal{F}_M^t(\boldsymbol{h}_i^{t-1}, \boldsymbol{h}_j^{t-1}, \boldsymbol{e}_{i,j}), \quad (1)$$

$$\text{message aggregation:} \quad \boldsymbol{h}_i^t = \mathcal{F}_A(\boldsymbol{h}_i^{t-1}, \boldsymbol{m}_i^t), \quad (2)$$

where for $v_i$, the message function firstly summarizes the information (*i.e.,* $\boldsymbol{m}_i^t$) from its neighbors $\mathcal{N}_i$, and then uses it to update the node state. Then, in the readout phase, a task-specific readout function $\mathcal{F}_R$ operates on the final node representation $\boldsymbol{h}_i^T$ to produce a node output:

$$\text{readout phase:} \quad o_i = \mathcal{F}_R(\boldsymbol{h}_i^T). \quad (3)$$

Next, we will present the details of the proposed graph-based semantic mining model for pseudo ground-truth generation in WSSS.

### Group-Wise Semantic Mining Network

**Problem Definition:** Given a collection of training samples, our first goal is to generate corresponding pseudo ground-truths, which will later be used to supervise semantic segmentation networks. To achieve this, we formulate the problem as graph-based semantic co-mining among multiple images. Formally, we denote $\mathcal{I} = \{(\boldsymbol{I}_i, \boldsymbol{l}_i)\}_{i=1}^N$ as the training data, where $\boldsymbol{I}_i \in \mathbb{R}^{w \times h \times 3}$ is an image and $\boldsymbol{l}_i \in \{0, 1\}^L$ is the corresponding image-level ground-truth with $L$ possible semantic categories. During training, we *selectively* sample $K$ images $\{\boldsymbol{I}_i\}_{i=1}^K$ as a mini-batch, and model their relations as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the image $\boldsymbol{I}_i$ is denoted as node $v_i \in \mathcal{V}$, and the relation between $v_i$ and $v_j$ is represented by edge $e_{i,j} \in \mathcal{E}$. To better capture more comprehensive common semantics, we consider two nodes $v_i$ and $v_j$ to be linked only if there is at least one semantic category shared between them. Besides, we assume that every node has a *self-edge*, *e.g.,* $e_{i,i}$ for $v_i$.

Given the above definitions, our network aims to conduct pseudo ground-truth generation in a graph learning scheme, under the full supervision of image-level labels as well as the implicit semantic relations among different images. In this manner, our model can capture richer semantic information and obtain more accurate pseudo labels. Next, we describe the details of each component in our model.

**Node Embedding:** As an initial step, we abstract a high-level feature representation for each input image. Formally, given $\boldsymbol{I}_i$, we extract features $\boldsymbol{h}_i \in \mathbb{R}^{W \times H \times C}$ from the convolutional stages of a standard classification network (*e.g.*, VGG (Simonyan and Zisserman 2015)). The embedding of node $v_i$ is then initialized by $\boldsymbol{h}_i$, which is a $(W, H, C)$-dimensional tensor preserving full spatial details for more effective pixel-level matching during graph reasoning.

**Edge Embedding:** For each edge $e_{i,j}$ connecting $v_i$ to $v_j$, we aim to learn an edge embedding $\boldsymbol{e}_{i,j}^t$ at each iteration $t$ to find the correct semantic correspondence between the two nodes. This is achieved by dense matching over node embeddings using the following bilinear model:

$$\boldsymbol{e}_{i,j}^t = \boldsymbol{h}_i^t \boldsymbol{W} \boldsymbol{h}_j^{t\top} \in \mathbb{R}^{WH \times WH}, \tag{4}$$

where $\boldsymbol{h}_i^t \in \mathbb{R}^{WH \times C}$ and $\boldsymbol{h}_j^t \in \mathbb{R}^{WH \times C}$ are flattened into matrix representations for computational convenience. $\boldsymbol{W} \in \mathbb{R}^{C \times C}$ is a trainable weight matrix. In Eq. (4), $\boldsymbol{e}_{i,j}^t$ encodes the similarity between $\boldsymbol{h}_i^t$ and $\boldsymbol{h}_j^t$ for all pairs of spatial locations. For the edge $e_{j,i}$, its embedding at iteration $t$ is simply calculated as $\boldsymbol{e}_{j,i}^t = \boldsymbol{e}_{i,j}^{t\top}$.

It should be noted that Eq. (4) introduces a large number of parameters, increasing the computational cost. To alleviate this, $\boldsymbol{W}$ is approximately factorized into two low-rank matrices $\boldsymbol{P} \in \mathbb{R}^{C \times \frac{C}{d}}$ and $\boldsymbol{Q} \in \mathbb{R}^{C \times \frac{C}{d}}$, where $d$ $(d > 1)$ is a reduction ratio. Then, Eq. (4) can be rewritten as:

$$\boldsymbol{e}_{i,j}^t = \boldsymbol{h}_i^t \boldsymbol{P} \boldsymbol{Q}^\top \boldsymbol{h}_j^{t\top} \in \mathbb{R}^{WH \times WH}. \tag{5}$$

Eq. (5) has significant advantages over Eq. (4) in both model parameters and computational cost: 1) it reduces the number of parameters by $2/d$ times; 2) it only requires $(2WHC^2 + W^2 H^2 C)/d$ multiplication operations, instead of the $WHC^2 + W^2 H^2 C$ in Eq. (4).

In addition, for each self-edge $e_{i,i}$, its embedding $\boldsymbol{e}_{i,i}$ captures the self-relation over the node representation $\boldsymbol{h}_i$. We compute $\boldsymbol{e}_{i,i}^t$ at iteration $t$ by self-attention (Vaswani et al. 2017; Wang et al. 2018a), which can effectively capture long-range semantic dependencies. In particular, the self-attention calculates the response at a position by attending to all the positions within the same node embedding:

$$\boldsymbol{e}_{i,i}^t = \text{softmax}(\phi_f(\boldsymbol{h}_i^t)\phi_g^\top(\boldsymbol{h}_i^t))\phi_h(\boldsymbol{h}_i^t) + \boldsymbol{h}_i^t \in \mathbb{R}^{W \times H \times C}, \tag{6}$$

where $\phi_{\{f,g,h\}}$ are $1 \times 1$ convolutional operators. As seen, we also consider it to be a residual layer in Eq. (6), which can effectively preserve information in the original feature map.

**Message Passing:** Given the node and edge embeddings, our model iteratively updates the hidden states of graph nodes by applying message functions to collect information from their neighboring nodes. More specifically, for a node $v_i$, it absorbs knowledge along two types of edges: 1) a self-edge $e_{i,i}$ that encodes rich context-aware knowledge in $v_i$;

and 2) other edges $\{e_{j,i}\}_j$ that connect $v_j$ to $v_i$. For the former, our message function directly reads the message from $e_{i,i}$, *i.e.*, $\boldsymbol{m}_{i,i}^t = \boldsymbol{e}_{i,i}^{t-1}$; while for the latter, the messages are summarized as:

$$\boldsymbol{m}_{j,i}^t = \text{softmax}_r(\boldsymbol{e}_{i,j}^{t-1})\boldsymbol{h}_j^{t-1} \in \mathbb{R}^{WH \times C}, \tag{7}$$

where $\text{softmax}_r$ denotes the row-wise softmax operation. In Eq. (7), we accumulate knowledge from $\boldsymbol{h}_j^{t-1}$, which is weighted based on the similarity between $\boldsymbol{h}_i^{t-1}$ and $\boldsymbol{h}_j^{t-1}$. $\boldsymbol{m}_{j,i}^t$ is then reshaped to a $(W, H, C)$-dimensional tensor. Then, we can easily summarize the message for $v_i$ at the $t$-th iteration as:

$$\boldsymbol{m}_i^t = \sum_{v_j \in \mathcal{N}_i} \boldsymbol{m}_{j,i}^{t-1} + \boldsymbol{m}_{i,i}^{t-1}. \tag{8}$$

Next, the aggregation function $A$ updates the hidden states of nodes, as given in Eq. (2). In our method, $A$ is instantiated by a ConvGRU network (Ballas et al. 2016), which is an extension of the GRU update function used in (Gilmer et al. 2017). In this way, the message passing algorithm runs for $T$ steps before convergence, iteratively collecting messages and updating node embeddings.

**Readout Phase:** Having repeated the above process for $T$ time steps, we obtain the final node embedding $\boldsymbol{h}_i^T \in \mathbb{R}^{W \times H \times C}$ for $v_i$. Then, the readout function $R$ is applied to the features $\boldsymbol{h}_i^T$ for image classification:

$$\boldsymbol{l}_i^g = \mathcal{F}_R(\boldsymbol{h}_i^T) = \text{GAP}(\phi_r(\boldsymbol{h}_i^T)) \in \mathbb{R}^L, \tag{9}$$

where $\phi_r$ is a class-aware convolutional layer with kernel size $1 \times 1$ that obtains a feature map with $L$ channels, and GAP denotes a *global average pooling* layer which produces the final classification outputs.

**Pseudo Ground-Truth Generation by Self-Ensembling:** Once the classification results are obtained (Eq. (9)), we discover the discriminative image regions for a particular category following (Jiang et al. 2019). These regions are further thresholded to generate pseudo ground-truths.

Besides, as shown in Figure 2, for each input image, our network produces two outputs based on raw convolutional features $\boldsymbol{h}_i^0$ as well as enriched features $\boldsymbol{h}_i^T$. This not only introduces additional deeply supervised constraints (Lee et al. 2015) which could benefit the performance, but also enables the results to be further improved by ensembling the CAMs of multiple outputs. We found that the pseudo ground-truths from different outputs are well complementary with each other, and self-ensembling them by averaging can further improve the performance (see Table 3).

## Graph Dropout Layer

The above graph reasoning scheme enables our model to discover common semantics present in different images (Eq. (5)). The features of these semantics can be accordingly enriched by summarizing all the information from other images (Eq. (8)). However, standalone categories, which may exist only in a single image, are almost ignored in this procedure. To address this, we introduce a graph dropout layer to force the network to pay more attention to these categories. Formally, given the feature map $\boldsymbol{h}_i^t \in \mathbb{R}^{W \times H \times C}$ at the $t$-th

| Methods | Pub. | Val | Test |
|---|---|---|---|
| *MEFF (Ge, Yang, and Yu 2018) | CVPR18 | - | 55.6% |
| *GAIN (Li et al. 2018) | CVPR18 | 55.3% | 56.8% |
| *MDC (Wei et al. 2018) | CVPR18 | 60.4% | 60.8% |
| *RRM (Zhang et al. 2020) | AAAI20 | 60.7% | 61.0% |
| †MCOF (Wang et al. 2018b) | CVPR18 | 60.3% | 61.2% |
| †SeeNet (Hou et al. 2018) | NIPS18 | 63.1% | 62.8% |
| †DSRG (Huang et al. 2018) | CVPR18 | 61.4% | 63.2% |
| †AffinityNet (Ahn and Kwak 2018) | CVPR18 | 61.7% | 63.7% |
| †SS-WSSS (Araslanov and Roth 2020) | CVPR20 | 62.7% | 64.3% |
| †SSNet (Zeng et al. 2019) | ICCV19 | 63.3% | 64.3% |
| †IRNet (Ahn, Cho, and Kwak 2019) | CVPR19 | 63.5% | 64.8% |
| †CIAN (Fan et al. 2020b) | AAAI20 | 64.3% | 65.3% |
| †FickleNet (Lee et al. 2019) | CVPR19 | 64.9% | 65.3% |
| †IAL (Wang et al. 2020a) | IJCV20 | 64.3% | 65.4% |
| †SSDD (Shimoda and Yanai 2019) | ICCV19 | 64.9% | 65.5% |
| †SEAM (Wang et al. 2020b) | CVPR20 | 64.5% | 65.7% |
| †SubCat (Chang et al. 2020) | CVPR20 | 66.1% | 65.9% |
| †OAA+ (Jiang et al. 2019) | ICCV19 | 65.2% | 66.4% |
| †RRM (Zhang et al. 2020) | AAAI20 | 66.3% | 66.5% |
| †BES (Chen et al. 2020) | ECCV20 | 65.7% | 66.6% |
| †EME (Fan, Zhang, and Tan 2020) | ECCV20 | 67.2% | 66.7% |
| †MCIS (Sun et al. 2020) | ECCV20 | 66.2% | 66.9% |
| †ICD (Fan et al. 2020a) | CVPR20 | 67.8% | 68.0% |
| *Ours (VGG16) | – | 63.3% | 63.6% |
| †Ours (ResNet101) | – | **68.2%** | **68.5%** |

Table 1: Quantitative comparison of different methods on PASCAL VOC 2012 *val* and *test* in terms of mIoU. *: VGG backbone. †: ResNet backbone.

| Methods | Pub. | Val |
|---|---|---|
| BFBP (Saleh et al. 2016) | ECCV16 | 20.4% |
| SEC (Kolesnikov and Lampert 2016) | ECCV16 | 22.4% |
| DSRG (Huang et al. 2018) | CVPR18 | 26.0% |
| IAL (Wang et al. 2020a) | IJCV20 | 27.7% |
| Ours | – | **28.4%** |

Table 2: Quantitative comparison of different methods on COCO *val* in terms of mIoU. All methods use VGG16 as the backbone.

iteration, we average it along the channel dimension to obtain $o_i^t \in \mathbb{R}^{W \times H}$. Then, we generate a mask $s_i^t \in \mathbb{R}^{W \times H}$ as follows:

$$s_i^t = \begin{cases} \text{sigmoid}(o_i^t), & \text{if } r < \delta_r; \\ o_i^t \mathbf{1}(o_i^t < \max(o_i^t) * \delta_d), & \text{otherwise.} \end{cases} \quad (10)$$

Here, the parameter $\delta_r$ is a drop rate threshold, determining whether to carry out the dropout operation or not. The parameter $r$ is a scalar generated from a random generator. If $r < \delta_r$, $s_i^t$ is an importance map which supports the activations in $h_i^t$; otherwise, the layer drops the highly activated semantic regions to emphasize standalone semantics. $\mathbf{1}(x)$ is a matrix indicator function which returns 1 for the true elements in $x$, and 0 otherwise. The $\max(\cdot)$ operation calculates the maximum value for a 2D tensor. $\delta_d$ is a threshold controlling the dropout. Finally, we enhance the feature maps by:

$$\hat{h}_i^t = h_i^t \otimes s_i^t, \quad (11)$$

where $\otimes$ denotes spatial-wise multiplication. Note that $\hat{h}_i^t$ is then used to replace original features $h_i^t$ in the next iteration.

### Detailed Network Architecture

Our model is comprised of two sub-networks: a *classification network* for group-wise pseudo ground-truth generation and a *segmentation network* for semantic segmentation.

**Classification Network.** We choose VGG16 (Simonyan and Zisserman 2015) as the backbone, which is pre-trained on

ImageNet (Deng et al. 2009). We replace the last convolutional layer in VGG16 by dilated convolutions with a rate of 2, and the feature maps from this layer are taken as the initial node representations for the GNN. For each image $I_i$, our network has two outputs: an intermediate output $l_i^m$ which is directly obtained from the backbone (Figure 2), and a final output $l_i^g$ after graph reasoning (Figure 2). Then, the loss function of the classification network for image $i$ is:

$$\mathcal{L} = \mathcal{L}_{\text{CE}}(l_i^g, l_i) + \lambda \mathcal{L}_{\text{CE}}(l_i^m, l_i), \quad (12)$$

where $\mathcal{L}_{\text{CE}}$ indicates the standard sigmoid cross entropy loss, and $\lambda$ balances the two losses.

After training, we obtain the CAMs for each training image from the two classification layers mentioned earlier, and combine them to obtain foreground object seeds. Besides, we also follow conventional practices (Jiang et al. 2019; Fan et al. 2020a) to estimate background seeds using an off-the-shelf salient object detection model (Hou et al. 2017). The final pseudo labels are generated by combining the foreground and background seeds.

**Segmentation Network.** Following (Chang et al. 2020; Fan et al. 2020b), we choose DeepLab-v2 (Chen et al. 2017) as the segmentation network due to its superior performance in fully supervised semantic segmentation tasks.

## Experiments

### Experimental Setup

**Datasets:** We conduct our experiments on two datasets: PASCAL VOC 2012 (Everingham et al. 2010) and COCO (Lin et al. 2014). 1) **PASCAL VOC 2012** is currently the most popular benchmark for WSSS. The dataset contains 20 semantic categories (*e.g.,* person, bicycle, cow) and one background category. Following standard protocol (Huang et al. 2018; Lee et al. 2019; Wang et al. 2020b), extra data from SBD (Hariharan et al. 2011) is also used for training, leading to a total of 10,582 training images. We evaluate our model on the standard validation and test sets, which have 1,449 and 1,456 images, respectively. 2) **COCO** is a more challenging benchmark with 80 semantic classes. Since more complex contextual relations exist among these categories, it is interesting to examine the performance of our model in this dataset. Following (Wang et al. 2020a), we use the default train/val splits (80k images for training and 40k for validation) in the experiment.

**Evaluation Metric:** For fair comparison, we utilize a widely used metric (Wang et al. 2018b; Choe and Shim 2019; Sun

et al. 2020), *mean Intersection-over-Union (mIoU)*, for evaluation. The scores on the test set of PASCAL VOC are obtained from the official evaluation server.

**Training Details:** 1) *Greedy Mini-Batch Sampling.* During training, we design a heuristic, greedy strategy to sample $K$ training images in each mini-batch. Starting from a randomly sampled image $I_i$, we further find another $K-1$ images, each of which shares as many common semantic objects with $I_i$ as possible. These $K$ images are then used to build a $K$-node GNN. This sampling strategy enables our model to better explore rich relationships among groups of images and improve the results. 2) *Training Settings.* For the classification network, the number of nodes $K$ and message passing steps $T$ in the GNN are separately set to 4 and 3 by default. The input image size is $224 \times 224$. The entire network is trained using the SGD optimizer with initial learning rates of 1e-3 for the backbone and 1e-2 for the GNN, which are reduced by 0.1 every five epochs. The total number of epochs, momentum and weight decay are set to 15, 0.9, and 5e-4, respectively. The $\lambda$ in Eq. (12) is empirically set to 0.4 and the $d$ in Eq. (5) is set to 4. For the segmentation network, we follow the training setting in (Chen et al. 2017), but use the generated pseudo ground-truths as the supervision.

## Performance on PASCAL VOC 2012

We evaluate the proposed approach on PASCAL VOC 2012 against current top-performing WSSS methods that only operate with image-level labels. Following conventions, we evaluate the performance of our model using VGG16 (Simonyan and Zisserman 2015) and ResNet101 (He et al. 2016) as the backbones, respectively. As reported in Table 1, our model with ResNet101 achieves the best performance, scoring **68.2%** and **68.5%** on the *val* and *test* sets, respectively. It significantly outperforms the current leading approach, *i.e.,* ICD (Fan et al. 2020a), by **+0.4%** and **+0.5%** on the two evaluation sets.

In addition, Table 1 also shows that the proposed approach outperforms both pair-wise models (*i.e.,* CIAN (Fan et al. 2020b) and MCIS (Sun et al. 2020)), and all single-image based models (*e.g.,* RRM (Zhang et al. 2020), OAA+ (Jiang et al. 2019)), by a large margin. The reason lies in that existing methods exploit limited context in image collection, while our approach can learn more effective inter-image representations with GNNs.

In Figure 3, we also provide sample results for representative images in PASCAL VOC 2012 *val*. The images cover various challenging factors in WSSS, such as multiple objects, different semantic categories, small objects, and cluttered background. We see that our model can deal with these difficulties well, resulting in appealing segmentation results.

## Performance on COCO

We further examine the performance of our model on COCO. As reported in Table 2, our model achieves the best mIoU score (*i.e.,* **28.4%**) on the validation set, outperforming the second-best result, *i.e.,* IAL (Wang et al. 2020a), by **0.7%**. This further proves the superiority of our model.

| Aspect | | Variant | | mIoU |
|---|---|---|---|---|
| **Full Model** | | $T=3, K=4$ $\delta_r = 0.8, \delta_d = 0.7$ | | **68.2%** |
| Graph Reasoning | Node Number | $K=3$ | | 68.1% |
| | | $K=5$ | | 67.8% |
| | | $K=6$ | | 67.6% |
| | Message Passing | $T=2$ | | 67.8% |
| | | $T=4$ | | 68.0% |
| | | $T=5$ | | 68.0% |
| | Graph Dropout | $\delta_r = 0.8$ | $\delta_d = 0.9$ | 68.0% |
| | | | $\delta_d = 0.5$ | 67.7% |
| | | $\delta_r = 0.6$ | $\delta_d = 0.7$ | 66.8% |
| | | $\delta_r = 0.4$ | | 63.6% |
| | | *w/o* dropout | | 67.7% |
| Self-Ensembling | | intermediate output | | 64.1% |
| | | graph output | | 67.8% |
| | | self-ensembling | | 68.2% |

Table 3: Diagnostic experiments of our model on PASCAL VOC 2012 *val* in terms of mIoU. For all variants, we use ResNet101 as the backbone.

## Diagnostic Experiments

We further conduct diagnostic analysis on PASCAL VOC 2012 *val* set to verify the effectiveness of the essential modules in our approach. We use ResNet101 as the default backbone for all the studies. The performance of our full model with default parameters is given in the first row of Table 3.

**Number of Nodes $K$:** We first investigate the effect of the node number $K$ used in the GNN, which indicates the number of images in a group. As shown in Table 3, the model achieves comparably high performance with three or four nodes. However, when more nodes are added, the performance decreases significantly. This can be attributed to the trade-off between meaningful semantic relations and noise brought by group images. When $K=3$ or $4$, the semantic relations can be fully exploited to improve the integral regions of objects. However, when more images are further considered, meaningful semantic cues reach a bottleneck and noise, introduced by imperfect localization of the classifier, dominates, thus leading to performance degradation.

**Number of Message Passing Steps $T$:** We further evaluate the impact of the message passing steps by comparing the performance with different $T$ ranging from 2 to 5. From Table 3, we observe that the mIoU score is significantly improved when $T$ varies from 2 to 3. The performance decreases slightly when considering more steps. Therefore, we set $T=3$ as default for message passing.

**Graph Dropout Layer:** To verify the effectiveness of the proposed graph dropout layer, we design multiple experiments to search the optimal configuration of parameters *drop-rate* and *drop-th*. We observe that both parameters have great influences on the performance. As observed in Table 3, our model reaches the best performance at $\delta_r = 0.8$ and $\delta_d = 0.7$. If $\delta_d$ is higher (*e.g.,* 0.9), most discriminative regions will be kept, and thus ignored regions will remain unactivated. In contrast, if the $\delta_d$ is lower, the regions with high responses will be excessively dropped, leading to degraded classification accuracy.

Figure 3: Qualitative results on PASCAL VOC 2012 *val*. From top to bottom: input images, ground-truths, and our results.
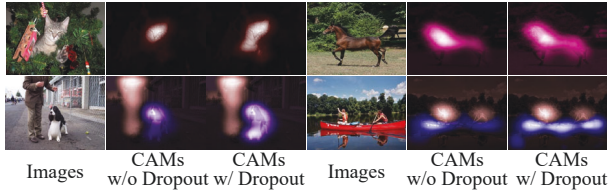


| Images | CAMs w/o Dropout | CAMs w/ Dropout | Images | CAMs w/o Dropout | CAMs w/ Dropout |

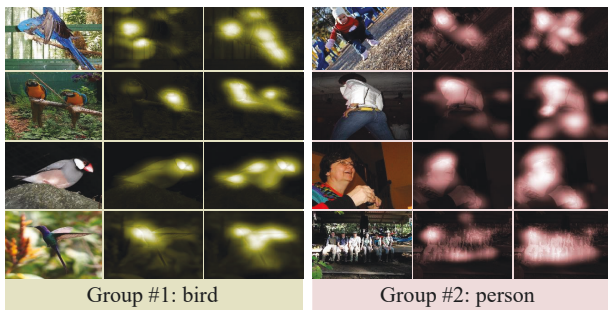Figure 4: Visual comparisons of CAMs generated *w/* or *w/o* the graph dropout layer.



Group #1: bird   Group #2: person

Figure 5: Visual comparisons of CAMs. Here we provide the results of two groups of images. For each group, we show the input images, CAMs from the *intermediate readout layer* and CAMs from the *graph readout layer* (from left to right). Our model clearly provides more accurate CAMs after group-wise graph reasoning.

In addition, the parameter $\delta_r$ controls whether to drop the responses or not during training. As shown in Table 3, a $\delta_r$ of $0.8$ helps to achieve the best mIoU score. Such a setting not only maintains the classification ability of the network by keeping discriminative regions with a high probability, but also drives the network to mildly attend to other regions. We can also see that by setting $\delta_r$ to smaller values (*e.g.,* $0.6$ or $0.4$), the performance encounters a significant decrease.

Moreover, we examine the performance of our model without the graph dropout layer. As seen, without the dropout layer, the performance of our model decreases by $0.5\%$ in terms of mIoU, which reveals its importance.

Finally, we illustrate some examples of the final CAMs generated *with* or *without* the graph dropout layer. As shown in Figure 4, without the dropout layer, the network only fo-

cuses on the most discriminative parts (*e.g.,* heads of the cat and the horse). This is improved with our dropout layer, which helps to highlight non-discriminative object regions.

**Self-Ensembling:** In addition to the supervision on the final outputs, we also introduce deep supervision signals on the intermediate features. Such multi-level supervision has proven effective for improving the performance of various vision tasks. Besides, this enables us to combine the multiple outputs with low cost to further boost the performance. Here, we examine the self-ensembling strategy by building three network variants, *i.e., intermediate output*, *graph output* and *self-ensembling*, in which the final CAMs are separately extracted from the intermediate readout layer, graph-aware readout layer, and their ensemble, respectively. As shown in Table 3, the *intermediate output* only obtains an mIoU score of $64.1\%$, greatly lagging behind the $67.8\%$ obtained by the *graph output*. This demonstrates that through iterative graph reasoning, our model can improve the image representations by integrating information from group images, leading to huge performance gains. Furthermore, the self-ensembling strategy boosts the performance to $68.2\%$.

In Figure 5, we illustrate two groups of images with their CAMs from the *intermediate readout layer* and *graph readout layer*. As seen, in both groups, the CAMs are well-refined to cover more complete foreground regions after graph reasoning. Besides, in many cases, the CAMs from two output layers complement with each other well, enabling better results to be obtained by self-ensembling.

## Conclusion

In this paper, we have introduced a group-wise learning model for weakly supervised semantic segmentation (WSSS). We formulate the task within a graph neural network (GNN), which operates on a group of images and explores their semantic relations for representation learning. By iterative reasoning, our model provides better pseudo ground-truths, further improving the semantic segmentation results. Experimental results on PASCAL VOC 2012 and COCO demonstrate that the proposed approach performs favorably against the state-of-the-art methods.

## Acknowledgments

# References

Ahn, J.; Cho, S.; and Kwak, S. 2019. Weakly supervised learning of instance segmentation with inter-pixel relations. In *CVPR*, 2209–2218.

Ahn, J.; and Kwak, S. 2018. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *CVPR*, 4981–4990.

Araslanov, N.; and Roth, S. 2020. Single-Stage Semantic Segmentation from Image Labels. In *CVPR*, 4253–4262.

Ballas, N.; Yao, L.; Pal, C.; and Courville, A. 2016. Delving deeper into convolutional networks for learning video representations. In *ICLR*.

Chang, Y.-T.; Wang, Q.; Hung, W.-C.; Piramuthu, R.; Tsai, Y.-H.; and Yang, M.-H. 2020. Weakly-Supervised Semantic Segmentation via Sub-Category Exploration. In *CVPR*, 8991–9000.

Chaudhry, A.; Dokania, P. K.; and Torr, P. H. 2017. Discovering class-specific pixels for weakly-supervised semantic segmentation. *arXiv preprint arXiv:1707.05821* .

Chen, L.; Wu, W.; Fu, C.; Han, X.; and Zhang, Y. 2020. Weakly Supervised Semantic Segmentation with Boundary Exploration. In *ECCV*.

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE TPAMI* 40(4): 834–848.

Choe, J.; and Shim, H. 2019. Attention-based dropout layer for weakly supervised object localization. In *CVPR*, 2219–2228.

Dai, J.; He, K.; and Sun, J. 2015. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *ICCV*, 1635–1643.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*, 248–255.

Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *IJCV* 88(2): 303–338.

Fan, J.; Zhang, Z.; Song, C.; and Tan, T. 2020a. Learning Integral Objects With Intra-Class Discriminator for Weakly-Supervised Semantic Segmentation. In *CVPR*, 4283–4292.

Fan, J.; Zhang, Z.; and Tan, T. 2020. Employing Multi-Estimations for Weakly-Supervised Semantic Segmentation. In *ECCV*.

Fan, J.; Zhang, Z.; Tan, T.; Song, C.; and Xiao, J. 2020b. CIAN: Cross-Image Affinity Net for Weakly Supervised Semantic Segmentation. In *AAAI*, 10762–10769.

Fan, R.; Hou, Q.; Cheng, M.-M.; Yu, G.; Martin, R. R.; and Hu, S.-M. 2018. Associating inter-image salient instances for weakly supervised semantic segmentation. In *ECCV*, 367–383.

Ge, W.; Yang, S.; and Yu, Y. 2018. Multi-evidence filtering and fusion for multi-label classification, object detection and semantic segmentation based on weakly supervised learning. In *CVPR*, 1277–1286.

Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; and Dahl, G. E. 2017. Neural message passing for quantum chemistry. In *ICML*.

Hariharan, B.; Arbeláez, P.; Bourdev, L.; Maji, S.; and Malik, J. 2011. Semantic contours from inverse detectors. In *ICCV*, 991–998.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Identity mappings in deep residual networks. In *ECCV*, 630–645.

Hou, Q.; Cheng, M.-M.; Hu, X.; Borji, A.; Tu, Z.; and Torr, P. H. 2017. Deeply supervised salient object detection with short connections. In *ICCV*, 3203–3212.

Hou, Q.; Jiang, P.; Wei, Y.; and Cheng, M.-M. 2018. Self-erasing network for integral object attention. In *NeurIPS*, 549–559.

Huang, Z.; Wang, X.; Wang, J.; Liu, W.; and Wang, J. 2018. Weakly-supervised semantic segmentation network with deep seeded region growing. In *CVPR*, 7014–7023.

Jiang, P.-T.; Hou, Q.; Cao, Y.; Cheng, M.-M.; Wei, Y.; and Xiong, H.-K. 2019. Integral object mining via online attention accumulation. In *ICCV*, 2070–2079.

Khoreva, A.; Benenson, R.; Hosang, J.; Hein, M.; and Schiele, B. 2017. Simple does it: Weakly supervised instance and semantic segmentation. In *CVPR*, 876–885.

Kolesnikov, A.; and Lampert, C. H. 2016. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *ECCV*, 695–711.

Lee, C.-Y.; Xie, S.; Gallagher, P.; Zhang, Z.; and Tu, Z. 2015. Deeply-supervised nets. In *Artificial intelligence and statistics*, 562–570.

Lee, J.; Kim, E.; Lee, S.; Lee, J.; and Yoon, S. 2019. Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference. In *CVPR*, 5267–5276.

Li, K.; Wu, Z.; Peng, K.-C.; Ernst, J.; and Fu, Y. 2018. Tell me where to look: Guided attention inference network. In *CVPR*, 9215–9223.

Lin, D.; Dai, J.; Jia, J.; He, K.; and Sun, J. 2016. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *CVPR*, 3159–3167.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *ECCV*, 740–755.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*, 3431–3440.

Lu, X.; Wang, W.; Danelljan, M.; Zhou, T.; Shen, J.; and Van Gool, L. 2020. Video object segmentation with episodic graph memory networks. In *ECCV*.

Marino, K.; Salakhutdinov, R.; and Gupta, A. 2017. The more you know: Using knowledge graphs for image classification. In *CVPR*, 20–28.

Pathak, D.; Krahenbuhl, P.; and Darrell, T. 2015. Constrained convolutional neural networks for weakly supervised segmentation. In *ICCV*, 1796–1804.

Qi, X.; Liao, R.; Jia, J.; Fidler, S.; and Urtasun, R. 2017. 3D graph neural networks for rgbd semantic segmentation. In *ICCV*, 5199–5208.

Qi, X.; Liu, Z.; Shi, J.; Zhao, H.; and Jia, J. 2016. Augmented feedback in semantic segmentation under image level supervision. In *ECCV*, 90–105.

Qu, M.; Bengio, Y.; and Tang, J. 2019. GMNN: Graph Markov Neural Networks. In *ICML*, 5241–5250.

Saleh, F.; Aliakbarian, M. S.; Salzmann, M.; Petersson, L.; Gould, S.; and Alvarez, J. M. 2016. Built-in foreground/background prior for weakly-supervised semantic segmentation. In *ECCV*, 413–432.

Santoro, A.; Raposo, D.; Barrett, D. G.; Malinowski, M.; Pascanu, R.; Battaglia, P.; and Lillicrap, T. 2017. A simple neural network module for relational reasoning. In *NeurIPS*, 4967–4976.

Scarselli, F.; Gori, M.; Tsoi, A. C.; Hagenbuchner, M.; and Monfardini, G. 2008. The graph neural network model. *IEEE TNN* 20(1): 61–80.

Shimoda, W.; and Yanai, K. 2019. Self-supervised difference detection for weakly-supervised semantic segmentation. In *ICCV*, 5208–5217.

Simonyan, K.; and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. In *ICLR*.

Song, C.; Huang, Y.; Ouyang, W.; and Wang, L. 2019. Box-driven class-wise region masking and filling rate guided loss for weakly supervised semantic segmentation. In *CVPR*, 3136–3145.

Sun, G.; Wang, W.; Dai, J.; and Van Gool, L. 2020. Mining Cross-Image Semantics for Weakly Supervised Semantic Segmentation. In *ECCV*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NeurIPS*, 5998–6008.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2018. Graph attention networks. In *ICLR*.

Vernaza, P.; and Chandraker, M. 2017. Learning random-walk label propagation for weakly-supervised semantic segmentation. In *CVPR*, 7158–7166.

Wang, W.; Lu, X.; Shen, J.; Crandall, D. J.; and Shao, L. 2019a. Zero-shot video object segmentation via attentive graph neural networks. In *ICCV*, 9236–9245.

Wang, W.; Zhang, Z.; Qi, S.; Shen, J.; Pang, Y.; and Shao, L. 2019b. Learning compositional neural information fusion for human parsing. In *ICCV*, 5703–5713.

Wang, W.; Zhou, T.; Qi, S.; Shen, J.; and Zhu, S.-C. 2021a. Hierarchical Human Semantic Parsing with Comprehensive Part-Relation Modeling. *IEEE TPAMI* .

Wang, W.; Zhou, T.; Yu, F.; Dai, J.; Konukoglu, E.; and Van Gool, L. 2021b. Exploring Cross-Image Pixel Contrast for Semantic Segmentation. *arXiv preprint arXiv:2101.11939* .

Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018a. Non-local neural networks. In *CVPR*, 7794–7803.

Wang, X.; Liu, S.; Ma, H.; and Yang, M.-H. 2020a. Weakly-Supervised Semantic Segmentation by Iterative Affinity Learning. *IJCV* 1–14.

Wang, X.; You, S.; Li, X.; and Ma, H. 2018b. Weakly-supervised semantic segmentation by iteratively mining common object features. In *CVPR*, 1354–1362.

Wang, Y.; Zhang, J.; Kan, M.; Shan, S.; and Chen, X. 2020b. Self-supervised Equivariant Attention Mechanism for Weakly Supervised Semantic Segmentation. In *CVPR*, 12275–12284.

Wei, Y.; Feng, J.; Liang, X.; Cheng, M.-M.; Zhao, Y.; and Yan, S. 2017. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *CVPR*, 1568–1576.

Wei, Y.; Liang, X.; Chen, Y.; Shen, X.; Cheng, M.-M.; Feng, J.; Zhao, Y.; and Yan, S. 2016. Stc: A simple to complex framework for weakly-supervised semantic segmentation. *IEEE TPAMI* 39(11): 2314–2320.

Wei, Y.; Xiao, H.; Shi, H.; Jie, Z.; Feng, J.; and Huang, T. S. 2018. Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation. In *CVPR*, 7268–7277.

Zeng, Y.; Zhuge, Y.; Lu, H.; and Zhang, L. 2019. Joint learning of saliency detection and weakly supervised semantic segmentation. In *ICCV*, 7223–7233.

Zhang, B.; Xiao, J.; Wei, Y.; Sun, M.; and Huang, K. 2020. Reliability Does Matter: An End-to-End Weakly Supervised Semantic Segmentation Approach. In *AAAI*, 12765–12772.

Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A. 2016. Learning deep features for discriminative localization. In *CVPR*, 2921–2929.

Zhou, T.; Li, J.; Wang, S.; Tao, R.; and Shen, J. 2020. MATNet: Motion-Attentive Transition Network for Zero-Shot Video Object Segmentation. *IEEE TIP* 29: 8326–8338.

Zhou, T.; Qi, S.; Wang, W.; Shen, J.; and Zhu, S.-C. 2021a. Cascaded Parsing of Human-Object Interaction Recognition. *IEEE TPAMI* .

Zhou, Y.; Zhou, T.; Zhou, T.; Fu, H.; Liu, J.; and Shao, L. 2021b. Contrast-Attentive Thoracic Disease Recognition with Dual-Weighting Graph Reasoning. *IEEE TMI* .