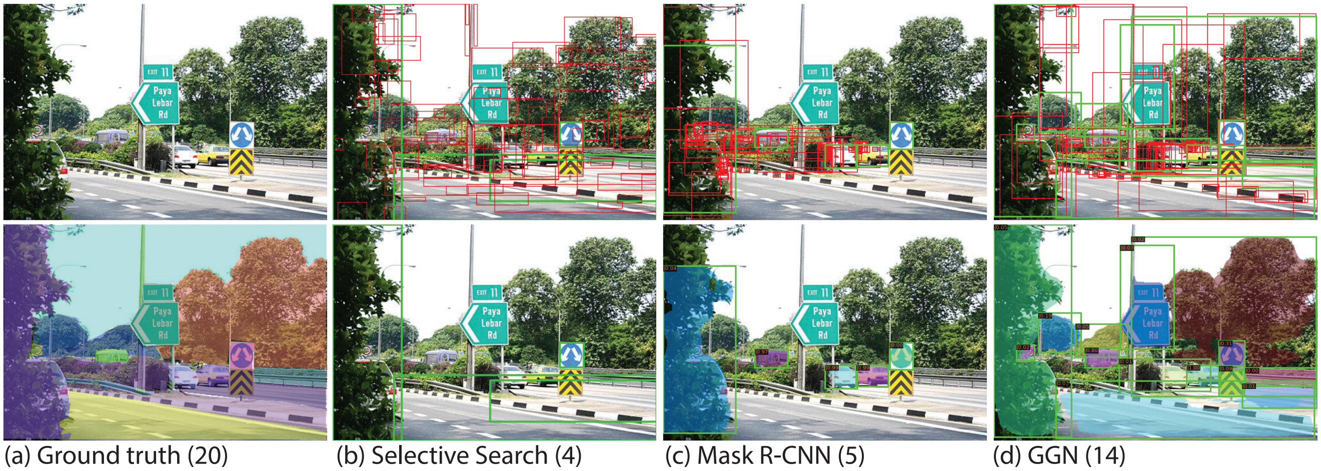
Open-World Instance Segmentation: Exploiting Pseudo Ground Truth From Learned Pairwise Affinity

Weiyao Wang[†], Matt Feiszli[†], Heng Wang[†], Jitendra Malik[†§], Du Tran[†]
[†] Meta AI Research [§] UC Berkeley
{weiyaowang,mdf,hengwang,trandu}@fb.com, malik@berkeley.edu
sites.google.com/view/generic-grouping/



(a) Ground truth (20) (b) Selective Search (4) (c) Mask R-CNN (5) (d) GGN (14)

Figure 1. **Comparison of GGN with different baselines**. (a) an input image from ADE20K [65] (upper row) with the ground truth object masks overlaid (lower row). Three different approaches: (b) Selective Search (SS) [55], (c) Mask R-CNN [23], and (d) our Generic Grouping Network (GGN) are applied on the same image to predict the top 100 proposals. The upper images provide all top-100 proposals predicted by three approaches with false and true positive proposals visualized in red and green boxes, respectively. The lower images provide only true positive proposals. The number of true positive proposals or ground truth objects are denoted in parentheses. Among 20 ground truth objects, SS recalls only 4, Mask R-CNN detects 5, and GGN retrieves 14. SS is a bottom-up non-parametric approach, thus has no notion of objectness. Mask R-CNN can make whole object proposals; however it still fails to detect objects that are not seen during training. Our GGN can predict whole object proposals and generalize to unseen categories.

Abstract

Open-world instance segmentation is the task of grouping pixels into object instances without any pre-determined taxonomy. This is challenging, as state-of-the-art methods rely on explicit class semantics obtained from large labeled datasets, and out-of-domain evaluation performance drops significantly. Here we propose a novel approach for mask proposals, Generic Grouping Networks (GGNs), constructed without semantic supervision. Our approach combines a local measure of pixel affinity with instance-level mask supervision, producing a training regimen designed to make the model as generic as the data diversity allows. We introduce a method for predicting Pairwise Affinities (PA), a learned local relationship between pairs of pixels. PA generalizes very well to unseen categories. From PA we construct a large set of pseudo-ground-truth instance masks; combined with human-annotated instance masks we train GGNs and significantly outperform the SOTA on open-world instance segmentation on various benchmarks including COCO, LVIS, ADE20K, and UVO. Code is available on project website.

1. Introduction

Instance segmentation is the task of grouping pixels into object instances [23]. In the closed-world setup, the task is to detect and segment objects from a predefined taxonomy. In contrast, the open-world setting requires segmenting objects of arbitrary categories. For a model trained in a closed-world setup, this means segmenting not only the "seen" categories (those presented at training time) but also the "unseen" categories (not seen during training) [28, 58].

There is generally a large performance gap between the seen and unseen domains. Leading computer vision systems today have tightly coupled recognition and segmentation; these systems are unable to segment out objects that they cannot recognize (e.g. Fig 1 (c)). Comparing Average Recall (AR@100) of Mask R-CNN [23] trained on 80 COCO [34] classes vs a subset of 20 classes, AR@100 of 60 classes out of training taxonomy drops from 49.6% to 19.9% when no mask of these classes is provided in training data. The "unseen" gap remains large if we train on larger taxonomy (e.g., 1,000+ classes in training data) Table 5). In contrast, humans can readily group and segment objects which they cannot categorize - few of us can identify the 6500 Passerine bird species, but we can readily segment out a perching bird from a tree branch. Or use another often-quoted example: our familiarity with a generic quadruped body plan enables us to segment out horses, donkeys and zebras, and even an okapi when first encountered.

On the other hand, models which were common in computer vision in 2000-2015 (e.g., [1, 5, 7, 19, 55, 62]), before deep learning for supervised object detection took off, were quite category-agnostic. They didn't work as well as, say, Mask R-CNN on a category for which it has trained, but they worked across the board (e.g. Figure 1 (b)). The goal was to come up with a moderately sized set of object proposals which included the true objects. The emphasis was on recall; precision was secondary. MCG [5] is an illustrative example. It starts with local grouping which produces a set of elementary regions of coherent color and texture, "super-pixels". These typically over-segment objects; e.g. a person might be broken up into a face, a torso, legs, parts of clothing, shadows, etc. MCG then assembles regions into objects by considering various groupings of regions, and ranks them on some "objectness" score. While some learning is involved in both edge detection and objectness ranking, the method works primarily with hand-crafted features and a small number of parameters, quite unlike the deep learning zeitgeist.

How do we get the best of both worlds? A modern instance segmentation system (eg. Mask R-CNN) would do well if given comprehensive training data containing a large number of examples from all visual categories. While we have a practically infinite supply of raw natural images, obtaining mask annotation is very expensive. Multiple approaches have emerged to handle this data problem. Self-supervised learning [11, 12, 21, 44] is the most well-known; self-learning [49, 50, 63, 67] is another approach, based on the classical idea of adding high-confidence guessed labels to previously unlabeled data, and then combining this "pseudo ground truth" data with real ground truth. We exploit this second strategy.

Our approach begins with a learned pairwise affinity predictor (Figure 2a), followed by a module which extracts and ranks segments (Figure 2b, essentially a very simplified version of MCG [5]). We can run this on any image dataset without using annotations; we extract the highest ranked segments as "pseudo ground truth" candidate objects. This is a large and category-agnostic set; we add it to our (much smaller) datasets of curated annotations, to train a Mask R-CNN instance segmentation module. Ideally this model should become more generic and class-agnostic (Figure 2c). Indeed, this simple approach produces impressive gains compared to closed-world training on the same backbone (Mask R-CNN) (Table 5, Table 6, Table 7): **+11%** on VOC to Non-VOC cross-category evaluation, **+3.9%** on COCO to LVIS cross-category evaluation, **+5.8%** on COCO to ADE20K and **+5.2%** on COCO to UVO.

Our contributions in this paper include:

- A novel approach, **G**eneric **G**rouping **N**etworks (GGNs), for open-world instance segmentation; GGN exploits additional pseudo ground truth supervision generated from learned pixel-level pairwise affinities.
- Comprehensive ablation experiments which provide insights about GGNs and the problem of open-world instance segmentation.
- GGNs achieve state-of-the-art performance in open-world instance segmentation on various benchmarks including COCO, LVIS, ADE20K, and UVO.

2. Related works

Object and instance segmentation. Before the success of deep learning, object segmentation approaches typically worked by grouping local regions into whole objects. Popular approaches include graph-based [16, 19], Normalized Cut [25], Graph Cut [8], Multiscale Combinatorial Grouping [5], and Selective Search [55]. Since deep learning, end-to-end approaches proved their success on problems such as semantic segmentation [39,41], instance segmentation [23], panoptic segmentation [30, 56]. Despite sharing the common problem of segmentation, our approach is different in the open-world setup: instead of assuming a closed-world taxonomy, our work aims at detecting and segmenting both seen and unseen objects.

Pairwise affinity based approaches. Pairwise affinity is used in most graph-based segmentation methods [8, 24, 25] as an important term defining a relation graph of pixels for segmentation. The pixel-level pairwise affinity can be either hand-constructed [8, 24, 25] or learned [17, 18, 29, 35, 37, 42, 54]. Similar to pairwise affinity, object boundary detection [43, 61] is a dual problem but offers weaker supervision (sec 5) and cannot be trained as-is on non-exhaustive

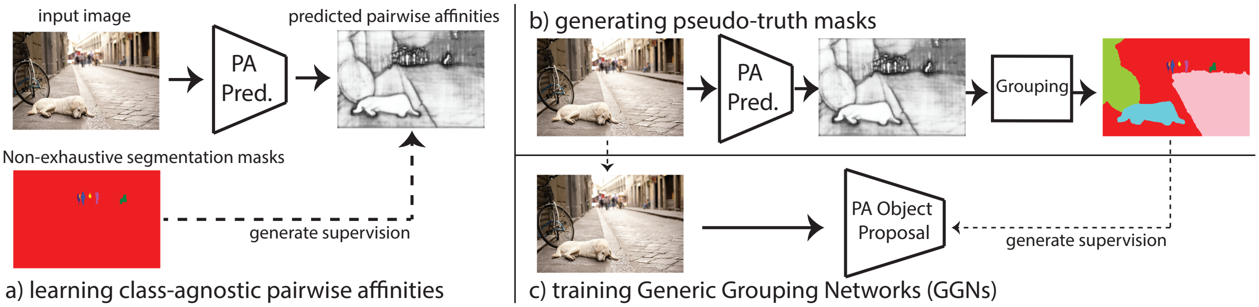

input image
PA Pred.
predicted pairwise affinities
Non-exhaustive segmentation masks
generate supervision
a) learning class-agnostic pairwise affinities
b) generating pseudo-truth masks
PA Pred.
Grouping
PA Object Proposal
generate supervision
c) training Generic Grouping Networks (GGNs)

Figure 2. **Overview of our approach**. (a) First, a Pairwise-Affinity Predictor (*PA Pred.*) is trained to predict pairwise affinities using non-exhaustive segmentation masks as supervision. (b) Once trained, *PA Pred.* is used to predict pairwise affinities of the images. A grouping module is then applied on the predicted pairwise affinity maps to generate pseudo-ground-truth masks. (c) A class-agnostic generic object proposal network (e.g., class-agnostic Mask R-CNN) is trained end-to-end using a combination of GT and generated pseudo-GT masks.

annotations. Different from previous approaches, instead of directly using the learned pairwise affinity for segmentation, we use it an intermediate representation for pseudo-ground-truth generation which is later used to train our generic grouping model. Another difference between our approach and previous learned pairwise affinity comes from the open-world setting of the problem.

Open-world benchmarks and approaches. Open-world setup [6, 45] has been recently (re)-introduced into various problems in computer vision such as recognition [31, 40], tracking [38], detection [26, 28, 57], and segmentation [58]. Among these, our work is mostly related to UVO [58] and OLN [28]. We share the same problem of interest of open-world instance segmentation with UVO [58]. However, [58] provides a new benchmark for the problem while our work provides an approach. Compared with the concurrent OLN work [28], which uses an objectness-based loss for generalization to unseen classes, our approach addresses generalization by combining pixel-level pairwise affinity with local grouping. Our work and OLN are orthogonal and complementary; as shown later, our approach alone is on par with OLN, and produces $4.5 - 5.7\%$ improvements when combined with OLN (see Table 6 in section 5).

3. Learning pairwise pixel affinities

Grouping can be locally represented by pixel pairwise relationship: whether two neighboring pixels should be grouped together or not. Given a 3-channel RGB input image $I \in \mathbb{R}^{3\times H\times W}$, we consider a pixel's pairwise relationship in a 3×3 neighborhood. This gives a pairwise affinities map of $P \in \{0,1\}^{8\times H\times W}$ and $P_{i,j} \in \{0,1\}^8$ encodes the local pixel-level pairwise affinities of the pixel (i, j) with its 8-neighboring pixels in the image I. Figure 4 (c) illustrates the pairwise affinity encodings of the two pixels at the centers of the two image patches marked with the pink and yellow squares in Figure 4 (a) and (b). We use pixel-level-prediction convolutional neural networks to predict P (*PA Pred.*, Figure 2 a), such as FCN [41] and UPerNet [60]. We remark that this is a dual problem to binary object boundary detection ([43, 61]), and techniques used for training *PA Pred.* can also be adopted to binary object boundary detectors to serve as the local representation for our framework.

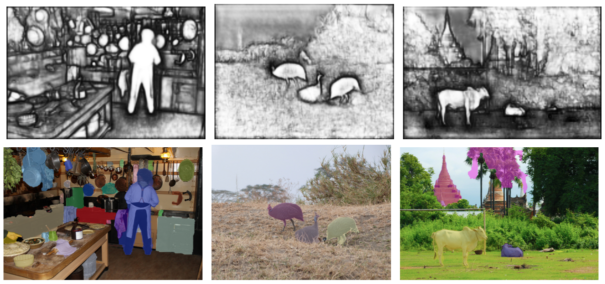


Figure 3. **Visualization of predicted pairwise affinities and generated pseudo masks, trained only on Person class in COCO.** Despite only seeing masks of Person during training, the PA predictor correctly captures pairwise relationships of other types of objects (top-row). By grouping pixels based on predicted PA, we can generate pseudo masks of other categories (bottom-row).

Training from non-exhaustive segmentation masks. Ideally, if all pixels in the image are exhaustively annotated with instance segmentation masks, e.g., all object boundaries are labelled, then all pairs of neighboring pixels can provide good supervision signal for learning pairwise affinity. However, exhaustive annotations for instance segmentation are expensive and time-consuming to obtain, so most datasets come with non-exhaustive segmentation masks (eg. COCO [34]). In particular, non-annotated out-of-taxonomy objects cannot be distinguished from background pixels. To address this, we only use neighboring pixels with object–object or object–background relations for training pairwise affinities; we ignore the unreliable background-background pairs. In addition, training of pairwise affinities is unbalanced: only pixels at an object's boundary has zero-valued affinities; all other pixels have affinity 1 to all their neighbors. We weight the positive affinities by computing the ratio between the positive affinities and negative affinities on a subset of training data (e.g., 0.05 for positive).

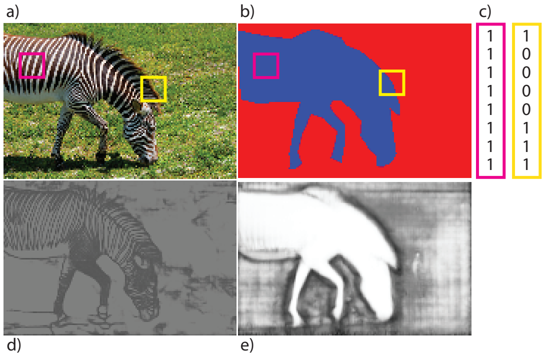


Figure 4. **Pairwise affinity encoding and prediction**. Visualization of an example input image (a) and its corresponding ground truth mask annotation (b) and two pairwise-affinity encoding vectors (c) of the two center pixels of the two image patches marked with pink and yellow squares. The pink-patch's center pixel belongs to the same instance with all of its 8-neighbor pixels, thus it is encoded with a vector of all ones. The yellow-patch's center pixel lies at the object boundary and has 4 neighbor pixels belonging to background, thus it is encoded with a binary vector with four 0s and four 1s. (d) Edge prediction of the image in (a) using off-the-shelf edge detector [53]. (e) Pairwise affinity prediction of the image in (a) using our Pairwise Affinity predictor. Our predictor is trained using only person category masks from COCO. Best viewed in color.

What does a Pairwise Affinity predictor learn? Intuitively, an ideal PA measure should discriminate between instance boundaries and instance interiors: i.e. whether two neighboring pixels cross an object boundary or not. Ideally, our PA predictor should be robust to boundaries of all objects, not just the categories seen during training; this is a key requirement for success in the open-world setting. Indeed, this is the case: our PA predictor learns instance boundaries and generalizes well to unseen classes. Figure 3 top-row and Figure 4 (e) visualize pairwise affinity predictions from our PA predictor, trained using ground truth masks of **only** the person category from COCO. Our PA predictor learns to generalize to unseen classes such as zebra, bird, temple, and cooking pan. We note that 'person' masks are particularly diverse owing to clothing and accessories. This is helpful for models to learn and generalize the notion of semantic boundaries. We show quantitative results on PA's generalization in section 5.2 and exploit this generalization behavior to improve segmentation in section 4. Finally, we point out that pairwise affinity should capture the semantics of instance boundaries; this is quite different from visual edge maps because many visual edges are not instance boundaries. Figure 4 (d) presents the edge prediction by an off-the-shelf edge detector [53]. Many visual edges on the back of the zebra are clearly not object boundaries, but are still detected by an edge detector.

4. Augmenting with pseudo ground truth

Existing state-of-the-art detectors and instance segmentation models, such as Mask R-CNN, often fail to detect and segment novel objects unseen during training. This can be caused by difficulties to group pixels into unknown entities due to lack of supervision signals during training. In addition, even a novel region is grouped and proposed, a generic concept of objectness is missing and such out-taxonomy proposals are suppressed. We kill two birds with one stone, by using pseudo-GT masks generated from PA to train these detectors. The pseudo-masks benefit from pixel diversity to provide novel segments not seen during training, and therefore enhance supervision signals for both novel grouping and a more inclusive concept of objectness.

Grouping pixels into regions. Based on predicted pairwise affinities, we leverage class-agnostic local grouping algorithms to group pixels into instances. One may use the *Connected Component* (CC) algorithm for grouping. CC treats all affinities independently using a hard cut-off threshold to decide pixel connections which may be a sensitive parameter to tune. Alternatively one may use *graph-based hierarchical* grouping (GBH) [16] which is a variant of agglomerative clustering. In segmentation literature [4, 64], Oriented Watershed Transform (OWT), globalized contour through Normalized Cut (gPb) [51] and Ultrametric Contour Map (UCM) [3] are also used for grouping from the edge map of an image. Following [3, 5], we first aggregate the image pairwise affinity map into a semantic edge map using pooling along the channel dimension, e.g., reducing from 8-channels to 1-channel. This semantic edge map is passed to OWT to generate initial segments, whose edges are then globalized through normalized cut. We take the average of semantic edge map and its globalized version as input to UCM for grouping. We acknowledge that a different linear combination of these two might work better upon further study. We provide the ablation comparing CC, GBH, and different components of OWT+gPb+UCM in section 5.

Computing objectness. Objectness [14] measures grouping qualities; in our framework, it is critical to decide which pseudo-GT masks to select for training detectors. An ideal objectness score should reveal over-segmentation and under-segmentation. In previous literature, objectness can be modeled by low-level features such as shape and contours such as MCG [5] or learned directly from annotated data as classification (Region Proposal Network [48]) or regression (Object Localization Network [28]). We consider both types of objectness. For low-level features, we use predicted pairwise affinities to define objectness score of each region R by total affinities $\mathcal{O}_{PA}(R)$:

$$\mathcal{O}_{PA}(R) = \frac{Inner(R)}{R_{inner}} - \frac{Outer(R)}{R_{boundary}} \quad (1)$$

where $Inner(R)$ and $Outer(R)$ are the inner and outer affinities of R defined by the sum of pairwise affinities of pixels inside or crossing-boundary of R, respectively. R_{inner} and $R_{boundary}$ denote the number of pixels inside R and on the boundary of R. Intuitively, we want to rank high for the region with strong inner pairwise affinities and weak affinities at the boundary (a.k.a strong cut). For learned objectness, we consider scoring from OLN [28] $\mathcal{O}_{OLN}(R)$:

$$\mathcal{O}_{OLN}(R) = \sqrt{centerness(R) * IoUness(R)} \qquad (2)$$

where $centerness(R)$ and $IoUness(R)$ are the centerness and IoU predictions of the bounding box of R. We can optionally combine $\mathcal{O}_{PA}$ and $\mathcal{O}_{OLN}$ by taking average.

Generic Grouping Networks (GGNs). We generate class-agnostic masks from PA predictor and grouping module and use the objectness score to rank regions provided by grouping methods (Figure 2 b). We then select top ranked regions from each image as pseudo ground-truth (GT) masks for training our generic object proposal network (Figure 2 c). Since the whole approach to generate pseudo-GT masks are designed in a class-agnostic grouping fashion, we expect the pseudo-GT masks to cover a diverse set of objects and parts, and more importantly most of them are from unseen categories, as shown in Figure 3 bottom-row. Since our GGN is trained on a large and diverse set of masks, it is expected to generalize to unseen classes, thus providing a good solution for open-world instance segmentation. GGN is generic in the sense of both pixels and models: it can work on different domains of images, labeled or unlabeled, and it can work on any architecture for object detection or segmentation, such as Faster R-CNN [48], Mask R-CNN [23], YOLO [47], or Swin Transformer [39]. The adoption is as simple as making the multi-class classification prediction head into a binary foreground vs. background classification head.

5. Experiments

5.1. Implementation Details

Datasets. We conduct experiments on COCO17 [34], LVIS [20], ADE20K [65], and UVO [58]. **COCO** is a standard benchmark for instance segmentation with 80 object categories annotated on 164k images. **LVIS** is an instance segmentation dataset with 1203 classes in a long-tail distribution. It is labeled as a federated dataset and does not include exhaustive label for its categories. We adopt LVIS to study cross-category generalization when a large taxonomy is provided. **ADE20k** is a semantic segmentation dataset with all pixels exhaustively annotated by object instances or stuff. **UVO** is a video instance segmentation dataset of YouTube videos (Kinetics400 [27]) with object masks exhaustively labeled. We use validation set of ADE20K (2000 images) and UVO sparse (7356 frames) to evaluate open-world segmentation in the wild in section 5.4. In all setups, we use only mask annotation (without class labels) for training and evaluation in the open-world, class-agnostic. We note that PA predictor, baseline methods (e.g., Mask R-CNN), and GGN has access to the **same** labeled masks.

Backbone architectures and loss function. We adopt UperNet [60] for our PA predictor to learn pairwise affinities. For training, our generic grouping networks, we use Mask R-CNN [23] with a ResNet-50 backbone as a default setup. Unless specified otherwise, models are initialized by ImageNet [13] pre-training. We use Binary Cross Entropy loss to train pairwise affinities. We ignore background-background affinity as in section 3. We note that back-propagating losses that include background-background affinities leads to very poor cross-category generalization (e.g., -15% Average Recall).

Ranking and selecting pseudo-GT masks. Unless otherwise specified, we use $\mathcal{O}_{PA}$ (Eq. 1) to rank pseudo-GT masks. We pick top-k pseudo-GT masks per image (k$\in$ $[1, 3]$), where k is selected to improve unseen categories performance while minimally impact seen performance.

Training and evaluation. We build model training and inference on MMDet [10] platform; all training are done with the default 1x schedule. Following previous object proposal literature [45,55], we use **average recall** (AR) over multiple IoU thresholds (0.5:0.95) to evaluate model performance.

Cross-category evaluation. Cross-category generalization is a major challenge for open-world: how do we detect and segment objects whose categories are outside training data. We split existing datasets by their categories to construct controlled environment for ablations (Table 1). In each setup, we train PA and baseline methods with the same splits of categories (no additional supervision for PA).

On COCO dataset, we follow common practice [28, 45] to split COCO into 20 classes overlapped with Pascal VOC [15] for training (seen) and use the rest of 60 COCO-exclusive classes for evaluation (unseen). We further include an extreme case by using only *person* class for training and the rest 79 classes for evaluation.

On LVIS dataset, some categories are highly overlapped: for example, clothes ("jacket", "wet suit") are highly overlapped with "person" when the person is wearing the clothes. In a class-agnostic setup, a detector trained with person masks can detect clothes as person and vice versa. Other examples of high-overlapped category pairs are "ball" with "tennis ball", "alcohol" with "beer bottle", or "computer monitor" with "television set". COCO and LVIS share the same set of images, but with different annotations. LVIS covers 1203 categories which include all 80 categories from COCO. COCO also exhaustively annotates all masks of objects that belong to its 80 categories while LVIS is annotated so as to maintain a similar number of masks across categories. This means that some object instances, even if they belong to LVIS categories, are not an-

Dataset	Train	Eval	Image	Mask
COCO	Person(1)	non-Person	64k	161k
	VOC(20)	non-VOC	95k	493k
LVIS	COCO(80)	non-COCO	100k	455k
	non-COCO(1122)	COCO	85k	749k
	+Person(1123)	non-Person	86k	775k

Table 1. **Cross-category generalization evaluation setups.** We split categories in COCO and LVIS to evaluate.

Grouping	CC	GBH	WT+UCM	+OWT	+gPb
Recall@all	14.4	17.1	23.6	23.8	**24.2**

Table 2. **Comparing different grouping methods**. Methods are applied on the same affinity maps and output roughly a similar number of proposals. OWT+gPb+UCM gives the best recall.

Aggregate	Min	Max	Mean
8-channel	22.8/**19.3**	18.1/16.7	22.1/18.9
1-channel	19.9/18.4	NA/NA	**23.1**/18.5

Table 3. **The effect of different PA aggregation.** Evaluated by AR@100 on both seen and unseen categories (VOC/non-VOC) and results are separated by /. 1-channel prediction is not applicable with max-pooling since the prediction target is all 1s (all pixels have at least one neighbor connected). 8-channel PA prediction with min pooling provides the best AR.

notated. As COCO masks are more exhaustively annotated, we use COCO masks and validate cross-category overlap with LVIS masks. We find that there are 67k LVIS masks outside COCO taxonomy having >0.5 IoU with COCO masks. To ensure a clear distinction between seen and unseen categories, we remove those masks in both training and validation for cross-category generalization evaluation. We study transfer performance by training on COCO categories and evaluate on LVIS non-COCO categories and vice-versa.

5.2. Learning Pairwise Affinities: Ablation Study

Grouping mechanisms. We revisit different grouping methods to construct segment masks from pairwise affinities: Connected Component (CC), Graph-Based Hierarchical [19] (GBH) and methods based on Ultrametric Contour Map [3] (UCM). In UCM, we ablate the effect of orientation in watershed transform (OWT vs. WT), and the effect of including globalized edge (gPb). To evaluate, we generate mask outputs from each of the method and directly evaluate their AR. We tune parameters for each method so that each has roughly the same number of output segments on average. Since CC gives a single non-overlapping output instead of a hierarchical structure like GBH or UCM, we use multiple thresholds and use all segments from each threshold. We found that UCM-based methods significantly outperforms other two methods (Table 2): whereas CC and GBH make decision on a merge based on a single pairwise relationship, UCM uses all relations between two segments, and is more robust. In addition, adding orientation and gPb further improves grouping results.

PA aggregation. In UCM, PA (8 neighbors) need to be aggregated into one value to feed to WT. The aggregation can be implemented by a pooling operation and can be applied before or after PA prediction. Specifically, we can either: (i) train a PA predictor to predict a 8-channel PA map then apply aggregation on the prediction output of the PA predictor; or (ii) train a PA predictor to predict a 1-channel PA map which is the aggregated version of ground truth. We compare different methods for aggregating pairwise affinity values (Table 4) and found min aggregation works the best. Alternatively, we can directly predict the aggregated pairwise affinities. We found that a single-channel prediction of mean of pairwise affinities works comparably with 8-neighbor predictions (Table 4). We remark that "1-channel, min" is equivalent to adopting a binary boundary detector trainer in our framework (e.g., HED [61]), which offers weaker supervision signals than PA.

5.3. Cross-category evaluation of GGN

We use the pseudo-GT masks from PA+Grouping to train detectors for open-world segmentation. We optionally use additional ground truth masks when they are available. Since PA generalize well in the open-world, the pseudo masks offer more diversities to the training data and therefore improve generalization of downstream detectors (GGNs). We begin by comparing PA with other candidate representations for open-world segmentation.

Pairwise affinity is a strong representation for open-world. Besides pairwise affinities, we consider a few other types of mid-level representations to encode grouping and generalize in the open-world:

- **Edge maps** are strong alternative to PA to encode grouping. We take SOTA edge detector DexiNed [53].
- **Depth maps** by pretrained Mannequin network [33].
- **Feature affinities** computed on semantic features, self-supervised trained on ImageNet (MoCoV2 [12]).

Most of the features here, except edge map, are not proper to run UCM to construct grouping. Therefore, we consider replacing the RGB input with the proposed representation (e.g., depth map or PA) to understand how well the representation can generalize in the open-world compared to RGB in cross-category evaluation.

Surprisingly, all representations have regularizing effects compared to RGB to improve generalization to unseen classes when only training on person (Table 4). Pairwise affinities outperform all other types of representations regardless of application methodologies (replacing input or adding pseudo masks). In particular, using UCM to generate pseudo-GT masks for edge does not benefit much, since without semantics, edge map can over-segment entities.

GGN significantly outperforms baselines on cross-

Method	replace RGB					UCM mask	
	RGB	depth	edge	MoCo	PA	edge	PA
nonPerson	4.9	10.9	10.5	10.7	14.1	7.9	**20.9**
nonVOC	19.9	17.8	21.3	21.8	26.5	19.7	**28.7**

Table 4. **Compare Pairwise Affinity with other types of mid-level representations**. All mid-level representations can serve to help generalizing to unseen categories in certain scenario to certain degrees. The significant improvement of pairwise affinities over edge map shows the importance for boundaries to contain semantics. Methods are evaluated by AR@100. Pairwise Affinities provide strongest generalization signals for open-world grouping.

Train (# classes)	Mask R-CNN	PA+ Grouping	GGN	Upper Bound
COCO-Dataset				
Person (1)	4.9	14.6	**20.9**	49.2
VOC (20)	19.9	22.0	**28.7**	49.6
LVIS-Dataset				
COCO (80)	16.5	17.1	**20.4**	36.1
non-COCO (1123)	21.7	16.2	**23.6**	35.1
+Person (1124)	27.3	18.4	**29.1**	44.2

Table 5. **GGN generalizes to out-taxonomy categories significantly better than baselines.** GGN also outperforms the pseudo-GT masks generated by pairwise affinities (denoted as PA+Grouping), proving the benefit of the instance-level training with additional pseudo-GT supervision. Upper bound indicates AR@100 achieved by training on entire taxonomy (all classes).

category generalization. We take top-scoring pseudo-GT masks and use them in addition with the ground truth masks; we remove pseudo-GT masks whose IoU overlap with in-taxonomy GT masks are greater than 0.5.

GGN has significantly stronger cross-category generalization compared to baseline Mask R-CNN (Table 5). On low to medium-sized taxonomy on COCO dataset, GGN achieves **+16%** AR@100 gain and **+8.8%** AR@100 gain when training on Person-only and training on 20 VOC classes, respectively. In the large-taxonomy setup, GGN achieves 1.8% to 3.9% gain on AR@100 in different setups. The gain is smaller when training on non-COCO categories; we believe that this is caused by the fine-grained taxonomy of LVIS: many classes in LVIS are objects parts or parts of other classes. Training on non-COCO categories on LVIS makes pairwise affinities closer to edge maps.

Additionally, we evaluate pseudo-GT masks generated by pairwise affinities with UCM. This is equivalent to our GGN but without using top-down instance-level training (as in Figure 2 b), denoted as *PA+Grouping*. Comparing with Mask R-CNN baseline, local grouping using learned pairwise affinities offer stronger performance in low to medium sized taxonomy. Finally, GGN significantly outperforms the *PA+Grouping* baseline which suggests the benefits of instance-level end-to-end training on pseudo-GT masks.

Backbone	Base	OLN	GGN	GGN+ $\mathcal{O}_{OLN}$	GGN+ OLN
Faster R-CNN	24.9	33.0	31.5	34.7	**37.2**
Mask R-CNN	19.9	26.9	28.7	30.9	**33.7**

Table 6. **GGN is competitive yet complementary to OLN.** Trained on VOC and evaluated on non-VOC using AR@100. Adopting $\mathcal{O}_{OLN}$ improves ranking of pseudo-masks and thus improves GGN; adopting OLN backbone further improves.

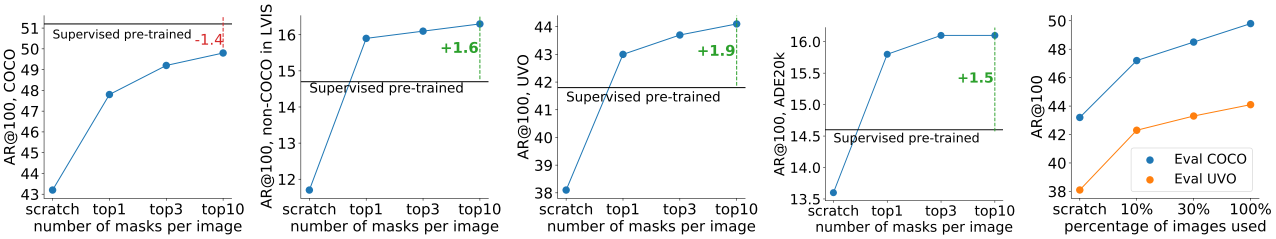
Method	Ranking	ADE20K		UVO	
		AR	AP	AR	AP
Selective Search		3.8	-	4.7	-
Mask R-CNN		14.7	6.4	40.1	18.5
GGN	$\mathcal{O}_{PA}$	18.3	7.9	42.6	19.4
	$\mathcal{O}_{PA} + \mathcal{O}_{OLN}$	21.0	**9.7**	43.4	20.3
GGN, pseudo-GT pre-training		**21.5**	9.3	**45.3**	**21.0**

Table 7. **Open-world segmentation in the wild on ADE20K and UVO.** GGN significantly outperforms the baseline Mask R-CNN when using the same amount of training data and annotation. In addition, having stronger objectness by combining $\mathcal{O}_{PA}$ and $\mathcal{O}_{OLN}$ further improves model performance. Finally, replacing ImageNet label pre-training with ImageNet pseudo-GT masks pre-training (sec 5.5) offers additional improvement.

GGN is comparable and complementary to state-of-the-arts object proposal method. Object Localization Network (OLN) [28] is a concurrent work to tackle open-world object proposal. OLN proposes to replace classification with localization quality prediction to avoid overfitting to annotated objects, which is similar to how we train pairwise affinities by not backpropagating the loss in un-annotated relationships. Different from OLN, pseudo masks generated bring more diversity to training data, and therefore help to generalize better. We compare GGN with OLN in Table 6 and find that GGN achieves similar performance as OLN (-1.5% on box AR@100, $+1.8\%$ on mask AR@100). When adding $\mathcal{O}_{OLN}$ (Eq. 2) to rank and select pseudo-GT masks, GGN improves by 2.2%. When adopting OLN as backbone, GGN sets new state-of-the-arts for cross-category generalization of VOC to COCO.

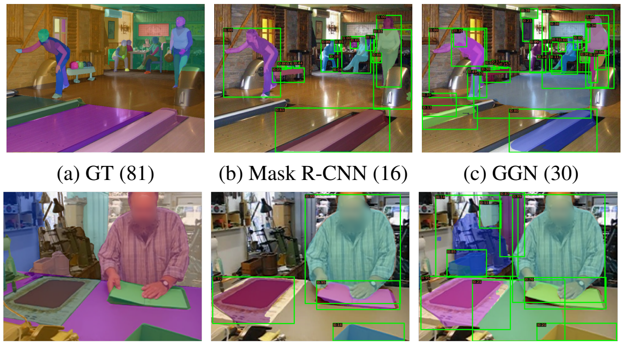
5.4. Evaluate open-world segmentation in the wild

Ablations in section 5.3 focus on cross-category generalization in a controlled version of open-world. A more practical question is: how well detectors can generalize across datasets in the wild? It is difficult to evaluate since common datasets, e.g., COCO and LVIS, are only *partially-annotated*. Evaluating open-world segmentation on such datasets may fail to capture performance difference across methods due to punishing precision and not rewarding recall [9]. To handle this, we adopt ADE20K [65] and UVO [58] for evaluating generic proposals in the wild open-world. Specifically, we treat each segmentation mask in ADE20K or UVO as a ground-truth semantic entity and



(a) COCO (b) LVIS, non-COCO (c) UVO (d) ADE20K (e) Scaling pixels

Figure 5. **GGN outperforms ImageNet supervised pre-training for open-world instance segmentation and demonstrates promising scaling behaviors**. We compare pseudo-GT mask pre-training by GGN with ImageNet label pre-training on closed-world (COCO, a), cross-category (non-COCO in LVIS, b) and open-world (UVO, ADE20K, c,d). Except closed-world setting, pseudo-GT masks provide stronger pre-training signals than ImageNet annotated labels (b-d). In addition, performance improves when more pseudo-GT masks selected per image (a-d) or more pixels (unlabeled images) (e) are used.



(a) GT (81) (b) Mask R-CNN (16) (c) GGN (30)

(d) GT (13) (e) Mask R-CNN (4) (f) GGN (9)

Figure 6. **Visualization of predictions of GGN and Mask R-CNN on ADE20K and UVO**. GGN retrieves more instances correctly (numbers denoted in parentheses) and covers a more diverse set of object categories.

evaluate Average Recall (AR) and Average Precision (AP). This setup evaluates both in-taxonomy and out-taxonomy segments. While UVO contains only objects, ADE20K also includes stuff masks. We emphasize that this is truly **in the wild test** as no fine-tuning is done on ADE20K or UVO.

We compare GGN with Selective Search [55] and Mask R-CNN baselines trained with the GT masks of all 80 COCO classes from COCO dataset (Table 7). GGN (enhanced by pseudo masks) significantly outperforms the baseline on both ADE20K and UVO dataset, both AR and AP. In addition, better ranking by combining $\mathcal{O}_{PA}$ and $\mathcal{O}_{OLN}$ further improves model performance. Qualitative results comparing Mask R-CNN and GGN on UVO and ADE20K are showed in Figure 6 (more in supplementary).

5.5. Pre-training on unlabeled images with GGN

Since PA-based bottom-up grouping can generate masks for any unlabeled image, we hypothesize that training with pseudo-GT masks from additional pixels may help open-world segmentation. Masks from PA generalizes well to new categories on new pixels, and thus benefit from pixel diversity. We study the effect of training GGN on pseudo-GT masks from unlabeled images from ImageNet [13].

Specifically, we use PA trained on 80 COCO categories from COCO with random initialization. We generate pseudo-GT masks on ImageNet images and pre-train a randomly initialized Mask R-CNN on pseudo-GT masks (18 epochs). We then finetune the model on COCO annotated masks (80 categories) for standard 1x schedule. Similar to previous training from random initialization setup [22], we use GroupNorm [59] for long training with small batch size.

Results are summarized in Figure 5. When evaluated on COCO categories (same as training, closed-world), pre-training by pseudo-GT masks performs slightly worse than supervised label pre-training (-1.4%). On open-world setup, however, pseudo-GT pre-training consistently outperforms supervised training. We note that different from closed-world [22], ImageNet supervised pre-training is a strong initialization for open-world (see supplementary). In addition, we observe two promising scaling behaviors of pseudo-GT pretraining: a. using more masks per image, despite some being noisy, improves performance; using more images/ pixels improves performance for both closed-world and open-world instance segmentation. We show similar results of Pre-training on images from OpenImages [32] in supplementary materials. Finetuning on both COCO annotated masks and PA generated Pseudo-masks on COCO images provides additional gain (last row in Table 7).

6. Conclusion

We have presented GGN, a novel approach for open-world instance segmentation which combines learned semantic boundaries with grouping to generate additional pseudo ground truth for instance-level training. GGNs significantly outperform baselines on various benchmarks. GGN is on par with state-of-the-art approaches, e.g., OLN [28], and when combined with OLN, GGN obtains an additional 6.8%, establishing new state-of-the-art results for open-world instance segmentation. Finally, we showed that GGN is robust when evaluated "in the wild" and benefits from training on additional unlabeled data.

Acknowledgement. We thank Ross Girshick for the discussion on baselines and grouping methods and Abhijit Ogale for the discussion about open-world setting.

References

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels, 2010. 2

[2] David Acuna, Amlan Kar, and Sanja Fidler. Devil is in the edges: Learning semantic boundaries from noisy annotations. In *CVPR*, 2019. 15

[3] Pablo Arbelaez. Boundary extraction in natural images using ultrametric contour maps. In *CVPR Workshops*, 2006. 4, 6, 15

[4] Pablo Arbelaez, Michael Maire, Charless C. Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916, 2011. 4

[5] Pablo Andrés Arbeláez, Jordi Pont-Tuset, Jonathan T. Barron, Ferran Marqués, and Jitendra Malik. Multiscale combinatorial grouping. In *CVPR*, 2014. 2, 4, 15

[6] Abhijit Bendale and Terrance Boult. Towards open world recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1893–1902, 2015. 3

[7] M. V. D. Bergh, G. Roig, X. Boix, S. Manen, and L. V. Gool. Online video seeds for temporal window objectness. In *2013 IEEE International Conference on Computer Vision*, pages 377–384, 2013. 2

[8] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, 2001. 2

[9] Neelima Chavali, Harsh Agrawal, Aroma Mahendru, and Dhruv Batra. Object-proposal evaluation protocol is 'gameable'. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 835–844, 2016. 7

[10] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 5

[11] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 2, 15

[12] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 2, 6

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 5, 8, 12

[14] Ian Endres and Derek Hoiem. Category independent object proposals. In *ECCV*, 2010. 4

[15] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, Jan. 2015. 5

[16] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2):167–181, 2004. 2, 4

[17] Charless C. Fowlkes, David R. Martin, and Jitendra Malik. Learning affinity functions for image segmentation: Combining patch-based and gradient-based approaches. In *CVPR*, 2003. 2

[18] Naiyu Gao, Yanhu Shan, Yupei Wang, Xin Zhao, Yinan Yu, Ming Yang, and Kaiqi Huang. Ssap: Single-shot instance segmentation with affinity pyramid. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 642–651, 2019. 2

[19] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2141–2148, 2010. 2, 6

[20] A. Gupta, P. Dollár, and R. Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5351–5359, 2019. 5, 12

[21] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9726–9735, 2020. 2, 15

[22] Kaiming He, Ross B. Girshick, and Piotr Dollár. Rethinking imagenet pre-training. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4917–4926, 2019. 8, 12

[23] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, pages 2980–2988, 2017. 1, 2, 5, 14

[24] Xuming He, Richard S. Zemel, and Miguel Á. Carreira-Perpiñán. Multiscale conditional random fields for image labeling. In *CVPR*, 2004. 2

[25] Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. 2

[26] K J Joseph, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Towards open world object detection. In *CVPR*, 2021. 3

[27] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *CoRR*, abs/1705.06950, 2017. 5

[28] Dahun Kim, Tsung-Yi Lin, Anelia Angelova, In So Kweon, and Weicheng Kuo. Learning open-world object proposals without learning to classify. *CoRR*, abs/2108.06753, 2021. 2, 3, 4, 5, 7, 8, 15

[29] Tae Hoon Kim, Kyoung Mu Lee, and Sang Uk Lee. Learning full pairwise affinities for spectral segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(7):1690–1703, 2013. 2

[30] Alexander Kirillov, Kaiming He, Ross B. Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *CVPR*, 2019. 2

[31] Shu Kong and Deva Ramanan. Opengan: Open-set recognition via open data generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 813–822, October 2021. 3

[32] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020. 8, 12

[33] Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T Freeman. Learning the depths of moving people by watching frozen people. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2019. 6

[34] Tsung-Yi Lin, M. Maire, Serge J. Belongie, James Hays, P. Perona, D. Ramanan, Piotr Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 2, 3, 5, 12

[35] Sifei Liu, Shalini De Mello, Jinwei Gu, Guangyu Zhong, Ming-Hsuan Yang, and Jan Kautz. Learning affinity via spatial propagation networks. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 2017. 2

[36] Yun Liu, Ming-Ming Cheng, Jiawang Bian, Le Zhang, Peng-Tao Jiang, and Yang Cao. Semantic edge detection with diverse deep supervision. *ArXiv*, abs/1804.02864, 2018. 15

[37] Yiding Liu, Si Cheng. Yang, Bin Li, Wen gang Zhou, Jizheng Xu, Houqiang Li, and Yan Lu. Affinity derivation and graph merge for instance segmentation. In *ECCV*, 2018. 2

[38] Yang Liu, Idil Esen Zulfikar, Jonathon Luiten, Achal Dave, Aljosa Osep, Deva Ramanan, Bastian Leibe, and Laura Leal-Taixé. Opening up open-world tracking. *CoRR*, abs/2104.11221, 2021. 3

[39] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 2, 5

[40] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2537–2546, 2019. 3

[41] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 2, 3

[42] Michael Maire, Takuya Narihira, and Stella X. Yu. Affinity cnn: Learning pixel-centric pairwise relations for figure/ground embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[43] Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Pablo Arbeláez, and Luc Van Gool. Convolutional oriented boundaries. In *European Conference on Computer Vision (ECCV)*, 2016. 2, 3

[44] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2536–2544, 2016. 2

[45] Pedro O. Pinheiro, Ronan Collobert, and Piotr Dollár. Learning to segment object candidates. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, page 1990–1998, Cambridge, MA, USA, 2015. MIT Press. 3, 5

[46] Stephen M. Pizer, E. Philip Amburn, John D. Austin, Robert Cromartie, Ari Geselowitz, Trey Greer, Bart ter Haar Romeny, John B. Zimmerman, and Karel Zuiderveld. Adaptive histogram equalization and its variations. *Computer Vision, Graphics, and Image Processing*, 39(3):355–368, 1987. 15

[47] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016. 5

[48] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28, pages 91–99. Curran Associates, Inc., 2015. 4, 5

[49] Ellen Riloff. Automatically generating extraction patterns from untagged text. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 2*, AAAI'96, page 1044–1049. AAAI Press, 1996. 2

[50] H. Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371, 1965. 2

[51] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000. 4

[52] Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection. In *arXiv:2005.04757*, 2020. 15

[53] Xavier Soria, Edgar Riba, and Angel Sappa. Dense extreme inception network: Towards a robust cnn model for edge detection. In *The IEEE Winter Conference on Applications of Computer Vision (WACV '20)*, 2020. 4, 6

[54] Srinivas C. Turaga, Kevin L. Briggman, Moritz Helmstaedter, Winfried Denk, and H. Sebastian Seung. Maximin affinity learning of image segmentation. In *Neural Information Processing Systems*, 2009. 2

[55] K. E. A. van de Sande, J. R. R. Uijlings, T. Gevers, and A. W. M. Smeulders. Segmentation as selective search for object recognition. In *ICCV*, pages 1879–1886, 2011. 1, 2, 5, 8

[56] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan L. Yuille, and Liang-Chieh Chen. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, 2021. 2

[57] Rui Wang, Dhruv Kumar Mahajan, and Vignesh Ramanathan. What leads to generalization of object proposals? In *ECCV Workshops*, 2020. 3

[58] Weiyao Wang, Matt Feiszli, Heng Wang, and Du Tran. Unidentified video objects: A benchmark for dense, open-world segmentation. In *ICCV*, 2021. 2, 3, 5, 7, 12

[59] Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, 2018. 8

[60] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018. 3, 5

[61] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1395–1403, 2015. 2, 3, 6

[62] 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

[63] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics*, ACL '95, page 189–196, USA, 1995. Association for Computational Linguistics. 2

[64] Xiao Zhang and Michael Maire. Self-supervised visual representation learning from hierarchical grouping. In *NeurIPS*, 2020. 4

[65] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5122–5130, 2017. 1, 5, 7, 12

[66] Barret Zoph, Ekin Dogus Cubuk, Golnaz Ghiasi, Tsung-Yi Lin, Jonathon Shlens, and Quoc V. Le. Learning data augmentation strategies for object detection. In *ECCV*, 2020. 15

[67] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc V. Le. Rethinking pre-training and self-training. *ArXiv*, abs/2006.06882, 2020. 2

In this supplementary material, we include:

1. Experiments on OpenImages [32] demonstrating that scaling number of unlabeled training images further improves GGN (section A).
2. Effects of ImageNet [13] pre-training on open-world instance segmentation (section B)
3. Additional qualitative results of open-world segmentation in the wild on ADE20K and UVO (section C).
4. A proof of concept experiment of using GGN for closed-world class-aware instance segmentation (section D).
5. Our discussion on the limitations and future directions of the proposed method (section E).
6. Ablations on data augmentation techniques for training PA (section F)

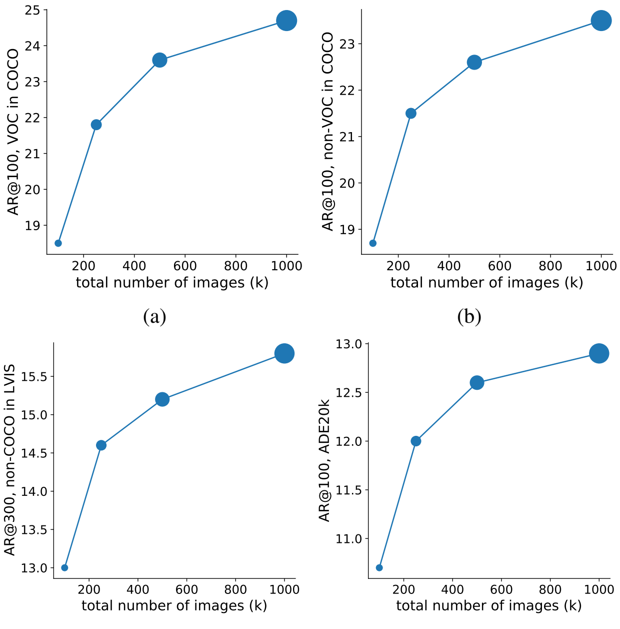
A. Improve GGN by scaling unlabeled pixels

In section 5 of main paper, we showed how our proposed method generates pseudo-GT masks on unlabeled images, and how GGN benefited from training on unlabeled images (Table 7). Here, we further show how GGN can be further improved by scaling the number of unlabeled training images.

We increase the size of unlabeled images (e.g., 100k, 250k, 500k, 1M) sampled from OpenImagesV4 [32] and take top-3 scoring pseudo-masks per image and use them as pseudo-GT masks for training. As shown in Figure 7, increasing the number of unlabelled training images continuously improves model performances in various setups. This further demonstrates the potential of GGN in both open-world (non-VOC, non-COCO [34], ADE20k [65]) and closed-world (VOC) instance segmentation.

B. ImageNet pre-training for open-world instance segmentation

In closed-world setup, ImageNet label pre-training offers limited values [22]: when training from scratch at 6x standard schedule, detectors perform on-par with 1x schedule finetuning from ImageNet label pre-training. This questioned if ImageNet label pre-training is a strong baseline to compare to (as we did in section 5.5). We argue that it is indeed a strong baseline, and that ImageNet label pre-training outperforms 6x schedule training from scratch (Table 8). This validates the value of ImageNet label pre-training for open-world instance segmentation, making it a proper baseline to compare with.



(a)

(b)

(c)

(d)

Figure 7. **The effect of scaling the number of images in training GGN.** We increase the size of subset of OpenImages [32] to 100k, 250k, 500k and 1M and train GGNs with pseudo-masks generated by pairwise affinities trained on VOC masks. In all setups, scaling images keeps improving model performance.

Training strategy	LVIS	UVO	ADE20K
6x schedule from scratch	13.1	41.5	13.7
ImageNet pre-training	**14.7**	**41.8**	**14.6**

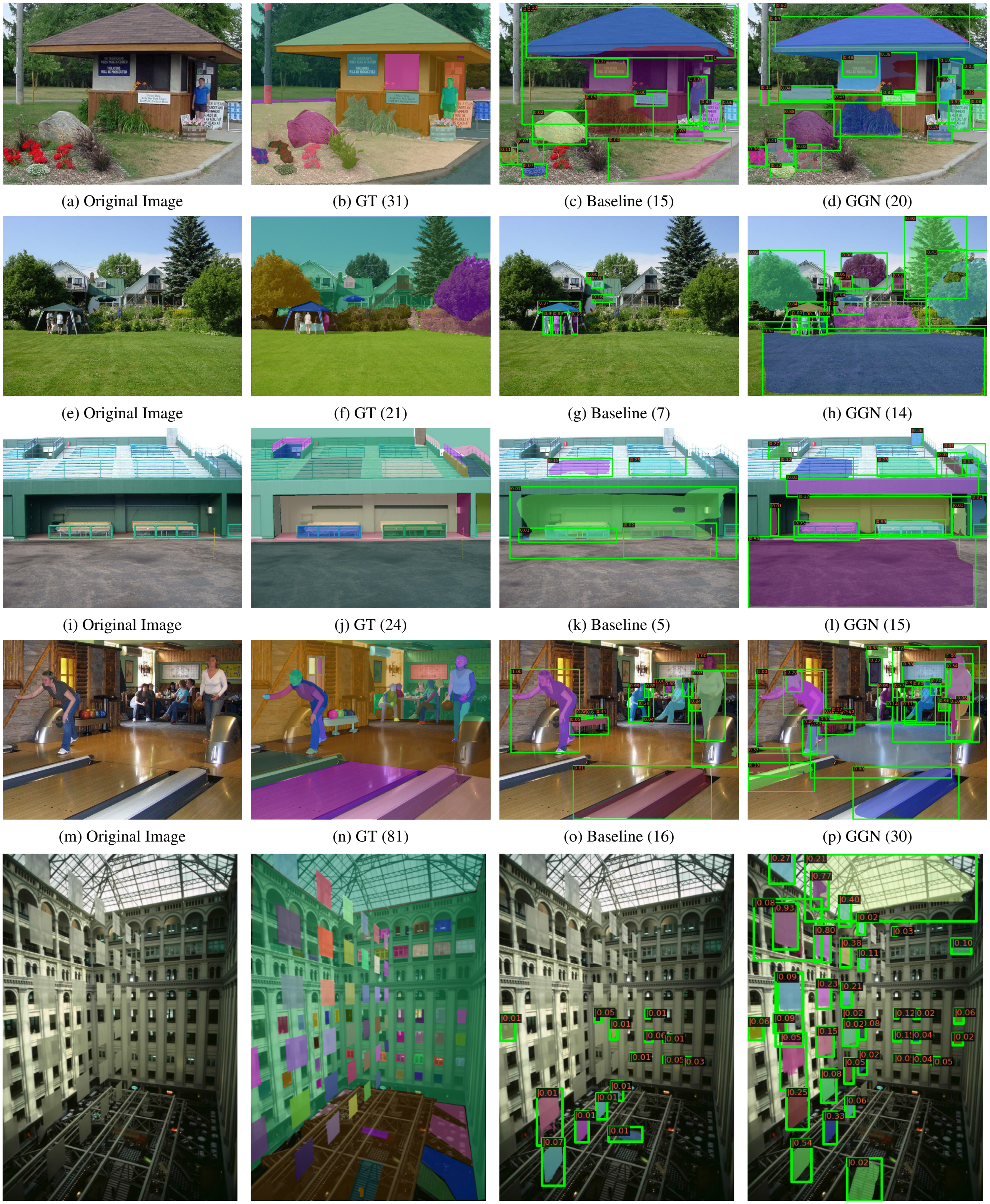
Table 8. **Different from common wisdom in closed-world instance segmentation, ImageNet pre-training outperforms long training schedule from random initialization in open-world**. We verify this with Mask R-CNN trained/ finetuned on COCO and evaluate on Non-COCO categories in LVIS [20], UVO [58] and ADE20K [65]

C. Qualitative results in the wild

We provide additional visualizations to compare GGN and baseline Mask R-CNN on ADE20k and UVO [58] (Figure 8 and Figure 9). Both models are trained with masks from 80 COCO categories, with GGN enhanced by pseudo-masks on COCO images. We show that GGN can recall more true positive segments than baseline, including novel objects, severely occluded objects and stuff.

D. Does generic grouping help closed-world segmentation?

In the previous experiments, we showed that GGN is useful for instance segmentation in the open-world (a.k.a class-aware instance segmentation). One may wonder if GGN is also useful for closed-world segmentation. In order to answer the question, we conduct the following proof-



(a) Original Image (b) GT (31) (c) Baseline (15) (d) GGN (20)

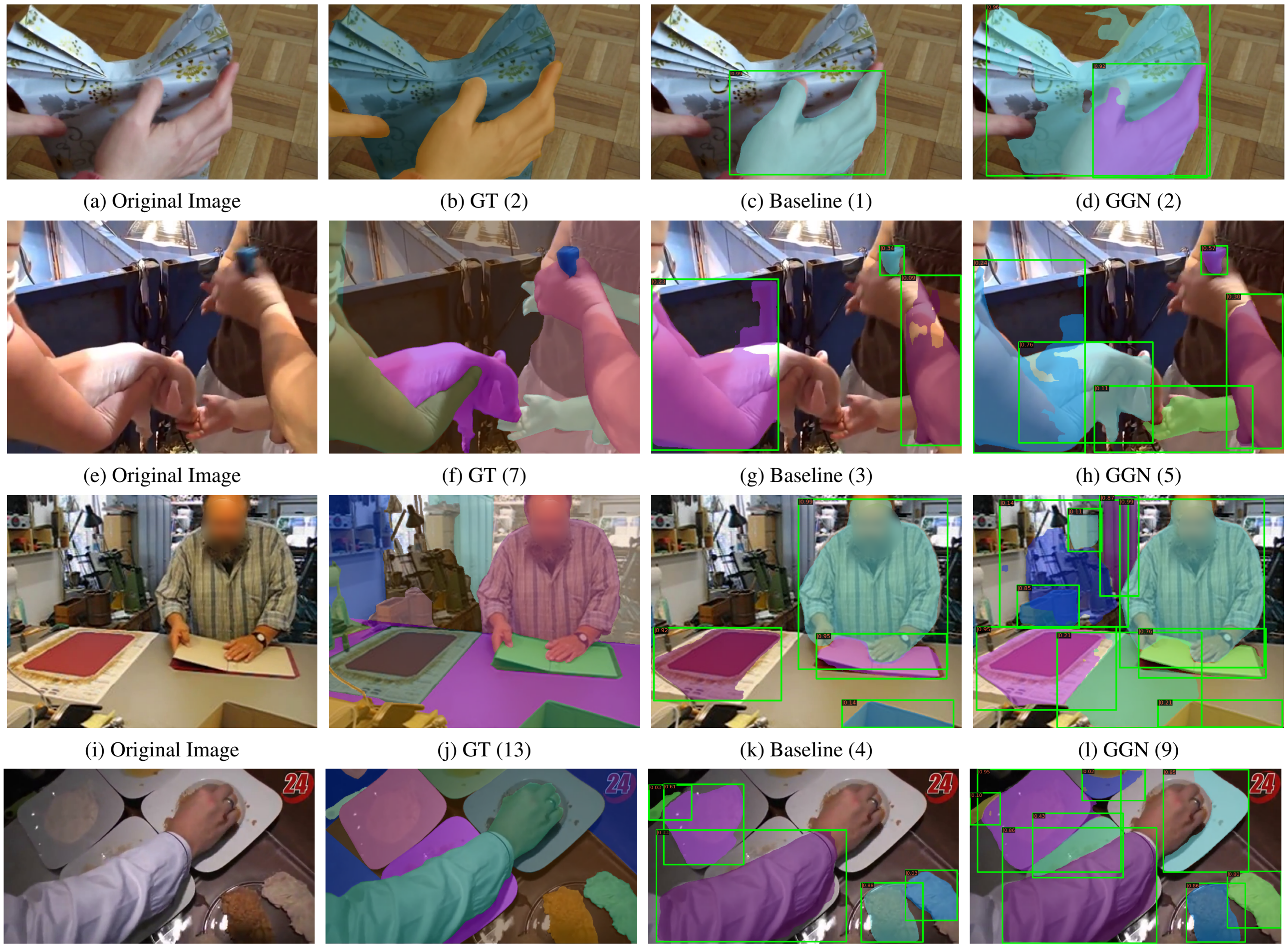
(e) Original Image (f) GT (21) (g) Baseline (7) (h) GGN (14)

(i) Original Image (j) GT (24) (k) Baseline (5) (l) GGN (15)

(m) Original Image (n) GT (81) (o) Baseline (16) (p) GGN (30)

(q) Original Image (r) GT (97) (s) Baseline (15) (t) GGN (40)

Figure 8. **Visualization of GGN compared to baseline on ADE20k.** We take top-100 scoring predictions for each of the methods. GGN detects significantly more true positive segments compared to baseline, including novel objects and stuff. Number in bracket represents number of retrieved segments.



(a) Original Image (b) GT (2) (c) Baseline (1) (d) GGN (2)

(e) Original Image (f) GT (7) (g) Baseline (3) (h) GGN (5)

(i) Original Image (j) GT (13) (k) Baseline (4) (l) GGN (9)

(m) Original Image (n) GT (11) (o) Baseline (5) (p) GGN (8)

Figure 9. **Visualization of GGN compared to baseline on UVO.** We take top-100 scoring predictions for each of the methods. GGN detects significantly more true positive segments compared to baseline, including novel objects and stuff. Number in bracket represents number of retrieved segments.

of-concept experiment. We adopt the standard Mask R-CNN [23] by replacing its RPN branch with our GGN. We note that our GGN also outputs bounding boxes and masks thus can completely replace RPN. In our experiment, GGN is pretrained with pseudo-GT in a class-agnostic and fixed (no fine-tuning or refinement) during class-aware training and evaluation. This means, during class-aware training and evaluation, only the recognition head is trained. We hypothesis the GGN can be competitive with RPN, even with a closed-world, class-aware setup. We name this modified architecture as *Two-Tower* to reflex the recognition and grouping branches. The grouping branch, GGN, is trained on only VOC-category masks. We compare this Two-Tower architecture with Mask R-CNN which is trained end-to-end in limited data domain: using only 10% of COCO images on all classes. Whereas Mask R-CNN is trained on grouping from all categories, the two-tower grouping module only leverages VOC masks and generated pseudo masks. Results are presented in Table 9.

Training length	Method	mAP	mAR
short	Mask R-CNN	10.6	36.2
	Two-Tower	**12.7**	**36.5**
normal	Mask R-CNN	**15.4**	**40.0**
	Two-Tower	13.5	36.5

Table 9. **Proof of concept on Two-Tower model for grouping and recognition** Mask R-CNN is trained on all 80 COCO categories. GGN, as the grouping module, is only trained on 20 VOC classes. The recognition module does not alter the mask predictions of the grouping module, and is trained with 80 COCO categories for classification. Two-tower is competitive in both short and normal training schedules.

E. Limitations and future directions

We present GGN that combines bottom-up grouping and top-down training for open-world instance segmentation.

The framework has shown significant gains and achieves the new state-of-the-art results on multiple benchmarks. In this section, we discuss the limitations of the approach, which also inform future directions to tackle.

Objectness. In GGN, we used WT+UCM [3] to group pixels into segments leveraging learned pixel pairwise affinities. However, WT+UCM has certain limitations: it has no notion of objectness, and therefore constructs pixel groups of "part" of an object. It is important to find novel methods to select good masks from all proposed pseudo-masks leveraging certain objectness prior, which can be learned [28] or hand-crafted [5].

Hierarchy of groups. When we select pseudo-GT masks generated from pairwise affinities, we ignore the natural hierarchical structure of the groups generated by UCM. It is worth understanding if enforcing grouping hierarchies can further improve the supervision signals.

Grouping as pretext task. Existing frameworks, such as Mask R-CNN, leverage recognition as pre-training for grouping (e.g., by pre-training on ImageNet). In this paper, we have demonstrated the value of training on unlabeled data to form grouping. A extension of this work should study how learning to group can potentially benefit recognition ability.

F. Data augmentation for learning PA

While data augmentation is well-explored in learning object proposals or masks [52, 66], it is not well-studied in the context of pairwise affinities or similar representation such as semantic edges [2, 36]. Different from bounding boxes or masks, pairwise affinities are local features and can be very sensitive to both pixel-level and spatial-level transforms.

Many data augmentation has a positive effect on pairwise affinities: multi-scaling is the strongest augmentation among all. Besides, CLAHE [46] and Hue-Saturation value jittering provide strong pixel-level augmentation. Not all augmentation helps to learn pairwise affinities: among 20 types of augmentation experimented, more than half hurts the performance of pairwise affinities (Fig. 10). For instance, different from the findings in contrastive learning [11, 21], all kinds of blurring hurts the performance of pairwise affinities. Pairwise affinities predict local relationship, which becomes uncertain with blurred images. In addition, orientation matters. While horizontal flipping and shearing contribute positively to learning pairwise affinities, vertical operators of the same kinds hurt the performance. We visualize a few augmentation in Figure 11.

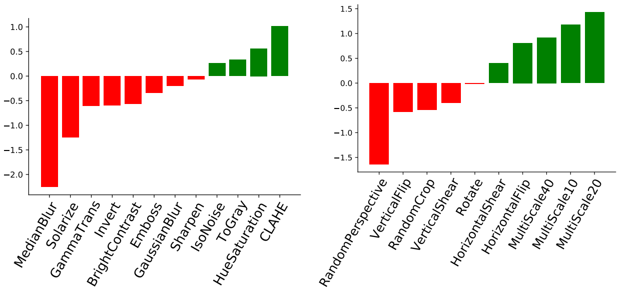
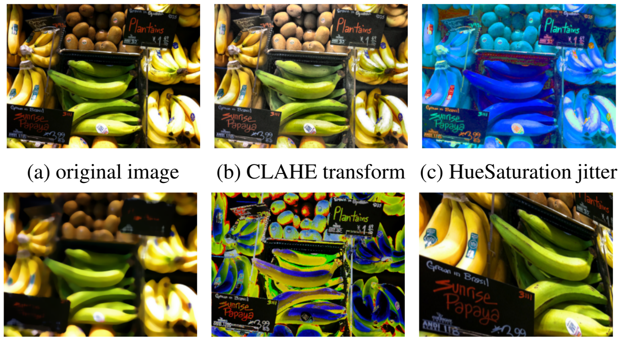


Figure 10. **The effects of different data augmentations on learning pairwise affinities.** The performance is evaluated by UCM masks generated by the pairwise affinities trained under different augmentations. Performance is represented as gain (loss) in AR100 compared to without augmentation.


(a) original image (b) CLAHE transform (c) HueSaturation jitter

(d) Median blur (e) Solarize (f) Random perspective

Figure 11. **Visualization of data augmentation.** Top row includes original image and two strong pixel-level augmentation. Bottom row contains three augmentation types that hurt the performance.