Open-world Instance Segmentation: Top-down Learning with Bottom-up Supervision

Tarun Kalluri[†*] Weiyao Wang[‡] Heng Wang[‡] Manmohan Chandraker[†]
Lorenzo Torresani[‡] Du Tran[‡]

[†]UC San Diego [‡] Meta AI
https://tarun005.github.io/UDOS

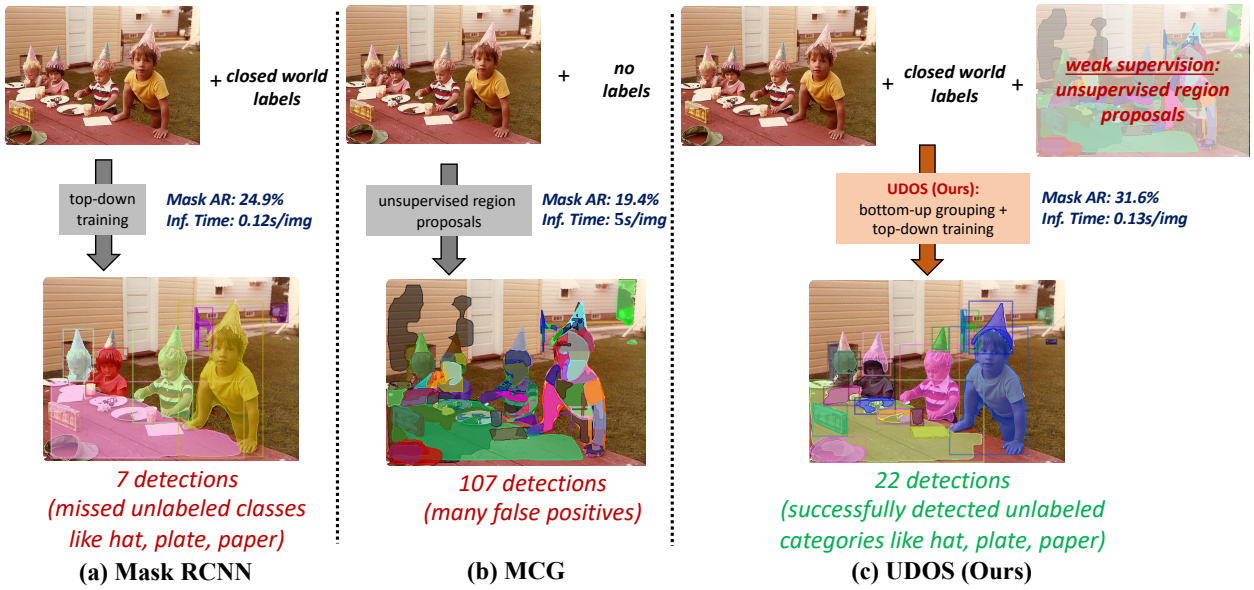


Figure 1: **Open world segmentation using UDOS (a)** Mask R-CNN [20], trained on VOC-categories from COCO, fails to detect many unseen categories due to seen-class bias; **(b)** MCG [46] provides diverse proposals for an open world, but predicts many false-positives which are over-segmentations with noisy boundaries; **(c)** combining the advantages of (a) and (b) into a joint framework, UDOS efficiently detects many unseen classes in open world when trained only using VOC-categories from COCO, while adding negligible inference time overhead.

Abstract

Many top-down architectures for instance segmentation achieve significant success when trained and tested on pre-defined closed-world taxonomy. However, when deployed in the open world, they exhibit notable bias towards seen classes and suffer from significant performance drop. In this work, we propose a novel approach for open world instance segmentation called bottom-**U**p and top-**D**own **O**pen-world **S**egmentation (UDOS) that combines classical bottom-up segmentation algorithms within a top-down learning framework. UDOS first predicts parts of objects using a top-down network trained with weak supervision from bottom-up segmentations. The bottom-up segmentations are class-agnostic and do not overfit to specific taxonomies. The part-masks are then fed into affinity-based grouping and refinement modules to predict robust instance-level segmentations. UDOS enjoys both the speed and efficiency from the top-down architectures and the generalization ability to unseen categories from bottom-up supervision. We validate the strengths of UDOS on multiple cross-category as well as cross-dataset transfer tasks from 5 challenging datasets including MS-COCO, LVIS, ADE20k, UVO and Open-Images, achieving significant improvements over state-of-the-art across the board. Our code and models are

[*]Work done during TK's internship at Meta.

available on our project page.

1. Introduction

Open world instance segmentation [55] is the problem of predicting class-agnostic instance masks for all objects in an image. The major challenge here is to segment novel instances: instances whose categories were out of the training taxonomy. This is an important step towards achieving robust and reliable real world deployment of instance segmentation models in various applications such as robotics [56], autonomous driving [11, 42], and embodied AI [49] where the model might regularly encounter novel objects. While scaling the size of the taxonomies during annotation is a possible counterpoint, on one hand, it requires significant human efforts to provide sufficient annotations for each category, and on the other hand, it is not possible to gather a comprehensive taxonomy of *all* categories; therefore, the ability to generalize to novel objects is preferable.

Unfortunately, instance segmentation frameworks such as Mask R-CNN [20] often couple recognition and segmentation [55] too closely to the extent that they are unable to segment out objects not labeled in the training data. This problem is exacerbated when these frameworks are trained with non-exhaustive annotations (*e.g.* MS-COCO [35]), where out-of-taxonomy objects are perceived as negatives (background). A prediction made on these objects are punished by the top-down supervision. We illustrate this in Fig. 1(a), where we show that a standard Mask R-CNN trained on the 20 VOC classes from COCO dataset effectively detects people, chair, table and car within the training taxonomy, but fails to detect out-of-taxonomy objects like hat, paper and plates.

On the other hand, classical bottom-up segmentation approaches [17, 46, 52] are by-design class-agnostic and unsupervised, making them suitable to the open world. They completely rely on low-level cues such as shape, size, color, texture and brightness between pixels to generate candidate object masks. However, these methods often produce over-segmentation of objects due to the lack of semantic notion of objectness as shown in Fig. 1(b), where MCG [46] generates over-segmentation of objects with noisy boundaries.

A natural question arises: how do we benefit from both these paradigms? We address this question in this work by designing a novel approach for open world instance segmentation called UDOS (bottom-**U**p and top-**D**own **O**pen-world **S**egmentation), that combines the advantages of aforementioned top-down and bottom-up methods into a single, jointly trainable framework. UDOS (Fig. 1c) reliably segments seen categories (person, table, chair) while generalizing well to unseen categories (party hats, paper, glass, plates) as well.

UDOS mainly stems from two key intuitions: Weak-supervision provided by class-agnostic segmentation from unsupervised bottom-up methods [17,46,52] should supplement the non-exhaustive human annotations. These class-agnostic segmentation complements the un-annotated area of an image which potentially contains out-of-taxonomy objects. This strategy forces the model to segment an image holistically, without leaving out a region as negative. However, only learning to predict part-masks from bottom-up grouping is insufficient, since they over-segment an object instance. To this end, our second insight lies in leveraging the seen-class supervision to bootstrap objectness, where we propose an affinity-based grouping module to merge parts into whole objects, and a refinement module to improve boundary qualities of the final predictions. To our best knowledge, UDOS is the first approach to effectively combine previously distinct top-down architecture and bottom-up supervision into a joint framework towards open-world instance segmentation. Through extensive set of empirical experiments, we demonstrate SOTA performance of UDOS in open world instance segmentation. In summary, our contribution towards open-world instance segmentation are three-folds:

1. We propose UDOS for open-world instance segmentation that effectively combines bottom-up unsupervised grouping with top-down learning in a single, jointly trainable framework (Sec. 3).
2. We propose an affinity based grouping strategy (Sec. 3.2) followed by a refinement module (Sec. 3.3) to convert noisy part-segmentations into coherent object segmentations. We show that such grouping generalizes well to unseen objects.
3. UDOS achieves significant improvements over competitive baselines as well as recent open-world instance segmentation methods OLN [29], LDET [48] and GGN [54] on cross-category generalization (VOC to NonVOC) as well as cross-dataset (COCO to UVO, ADE20K and OpenImagesV6) settings (Sec. 4).

2. Related Works

Object detection and instance segmentation. Object detection and instance segmentation were classically studied with hand-crafted low-level cues in a bottom-up fashion using graph-based grouping [16, 18], graph-based methods [10, 14, 15, 50], deformable parts [], hierarchical and combinatorial grouping [2, 46] or Selective Search [52]. With the introduction of deep learning, end-to-end top-down approaches demonstrated superior performance on a suite of detection and segmentation problems including object proposals [33, 45], object detection [47], semantic segmentation [40], instance segmentation [3, 9, 20] and panoptic segmentation [30, 53]. However, the problem setup in this paper is fundamentally different from previous works: instead of a closed-world assumption where training and testing share the same taxonomy, we focus on open-world

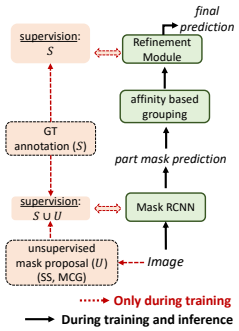


Figure 2: UDOS overview

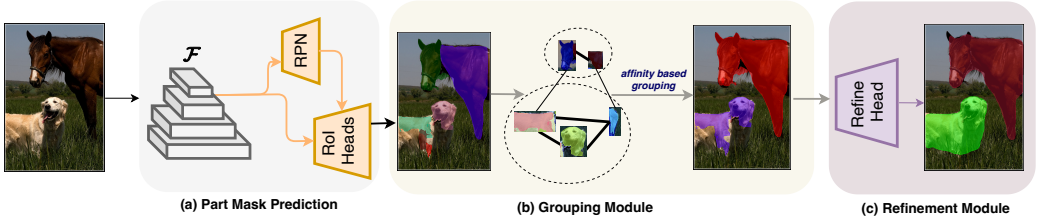


Figure 3: Proposed UDOS pipeline

(Fig. 2) UDOS overview: Overview of training and inference phases in UDOS. Note the the unsupervised proposal generation is only present during training and not needed during testing. **(Fig. 3) Proposed UDOS pipeline:** During training, we first augment the ground truth annotations on seen classes (S) with masks provided by the unsupervised segmentation algorithm (U) and use it to supervise the part mask prediction head in (a) (Sec. 3.1). As these predictions might only correspond to part-segments on unknown classes (*head* of the horse, *body* of the dog), we use an affinity based grouping strategy in (b) that merges part segments corresponding to the same underlying instance (Sec. 3.2 and Fig. 4). We then use a refinement head in (c) to output the final prediction that correspond to whole instances.

setup which requires to segment both in-taxonomy and out-of-taxonomy instances. As shown in Fig. 3 and Sec. 4.2, top-down methods suffer from seen-class bias and fail to detect novel objects in open world.

Open world instance segmentation Open-world vision requires generalization to objects not annotated in the training data [7, 45], and has regained interests in multiple computer vision problems [22, 28, 29, 38, 39, 55]. Our work focuses on open-world instance segmentation [55] which requires the model to correctly detect and segment instances whether their categories are in training taxonomy or not. This is different from [23, 32], where models are evaluated on object categories whose bounding box are available during training. Previous work [12, 13, 26, 44] often relies on additional cues such as video, depth, optical flow to solve this challenging problem, whereas UDOS requires no additional annotation and relies on unsupervised proposal generation to bootstrap training. UDOS is closely related to [29, 54, 55], while [55] proposes a new benchmark, our paper provides an approach. OLN [29] focuses on learning better objectness function to improve generalization but uses only seen class annotation during training. Our method, however, focuses on combining top-down training and bottom-up grouping towards novel object segmentation. GGN [54] carries similar philosophy as UDOS to leverage bottom-up grouping. While GGN performs grouping on learned pixel-level pairwise affinities, UDOS leverages bottom-up grouping with part-level pairwise affinities. We show that UDOS compares favorably to GGN on empirical evaluations while being potentially complementary.

Combining bottom-up and top-down. Bottom-up methods have recently been revisited in the context of representation learning to aid self-supervision [6, 21, 58]. In the relevant context of instance segmentation, bottom-up grouping is used to improve local segmentation quality through affinity maps [4, 37, 54, 57] or pixel grouping [25, 36, 51]. In contrast to these works designed for closed-world taxonomies, we propose to leverage both top-down training and bottom-up grouping to perceive and reason about open-world. Our work is also distinct from prior works employing grouping for segmentation [1, 31] as we group mid-level part-masks as opposed to low-level pixel features, and address the problem of open-world instance segmentation as opposed to 3D part-discovery or shape analysis [41].

3. Proposed Method

Problem definition. Given an image $I \in \mathbb{R}^{H \times W \times 3}$, the goal of open world instance segmentation is to segment all object instances in I regardless of their semantic categories, which includes objects both seen and unseen during training. Following prior works [29, 45, 54] for this problem, we adopt class-agnostic learning strategy, in which all annotated classes are mapped to a single foreground class during training and predictions are class-agnostic.

Method overview. An overview of UDOS during training and testing phases is shown in Fig. 2, and is explained in detail next. Our proposed approach is presented in Fig. 3 showing our part-mask prediction (Fig. 3a), affinity-based grouping (Fig. 3b) and refinement (Fig. 3c) modules. We build our backbone using a class-agnostic Mask R-CNN [20] with FPN [34], and we denote the FPN feature map computed by the backbone as $\mathcal{F}$.

3.1. Part-Mask Prediction

Generating candidate object regions. We first generate weak supervision in the form of approximate object regions for each image in the training set. We use off-the-shelf unsupervised segmentation algorithms (*e.g.* selective

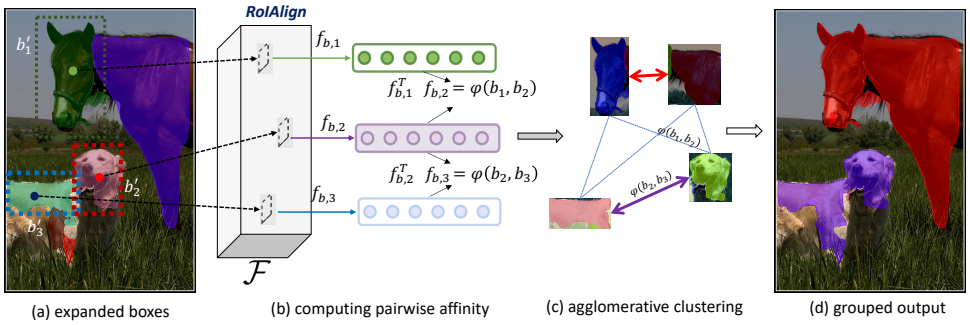


Figure 4: **Grouping module**. (a) the bounding boxes b_i of the predicted part-masks are expanded to incorporate local context. (b) The features $f_{b,i}$ are extracted using RoIAlign operator on the FPN features $\mathcal{F}$ with the expanded bounding boxes b'_i, and are used to compute pairwise affinity $\phi(b_i, b_j)$ using cosine similarity. (c) A clustering algorithm is used to group parts into whole object instances, as shown in (d). Note that the inaccuracies in the output from grouping module are later corrected by the refinement module.

search [52] or MCG [46]) that provide useful segmentation priors in images. By design, these segmentation masks are class-agnostic and cover entire span of the image without a distinction between in-taxonomy and out-taxonomy objects. Since we later group these masks, over-segmentation in proposal generation phase is desirable. So we tune hyper-parameters of the bottom-up segmentation algorithms (like scale and σ in sel. search) to prefer over-segmentation. We note that it is a *one time* effort to generate these pseudo-GT masks, before training, and they are *not needed* during inference or deployment (Fig. 3).

Augmenting labels using part-masks. Once we generate weak supervision for all images in the training set, we create a triplet (I, S, U) for each input training image I, in which $S = \{s_i\}_{i=1}^{N_s}$ is the set of ground truth box and mask labels covering only annotated categories, and $U = \{u_i\}_{i=1}^{N_u}$ is the set of masks generated by the unsupervised segmentation algorithm which has more exhaustive, albeit noisy, region proposals. We then use the set of augmented masks $A{=}S{\cup}U$ as supervision to train a top-down instance segmentation system, which we call part-mask prediction network as it might only predict parts of objects in line with the supervision provided (output Fig. 3a). To avoid duplicate labels, we filter out part masks from U that are overlapping with any ground truth mask in S with IoU greater than 0.9. Intuitively, while the masks in S provide information to detect within-taxonomy classes, the masks in U help to segment part masks for **all** objects, providing complementary training signal to the network. This strategy provides two advantages compared to top-down training with only ground truth masks in S. First, un-annotated regions of the image may contain valid objects that are out-of-taxonomy, which are covered by unsupervised region proposals U preventing the network from erroneously labeling these as background. Second, although masks in U might not correspond to complete objects, we still get useful training signal regarding detecting of out-of-taxonomy objects. For example, reliably detecting parts of a dog like head, body and ears in Fig. 3 will be useful in the final segmentation of the entire dog with our part-mask grouping strategy.

3.2. Grouping Module

To bridge the gap between mid-level part-mask predictions from Sec. 3.1 and complete object instances, we propose an efficient and lightweight grouping strategy to merge part predictions into whole objects. We compute pairwise affinities between features of the *expanded* part-masks, and cluster part-masks based on these affinities.

Pairwise affinity We denote the predictions made by the network in the first phase by $P = \{p_i\}_{i=1}^{n_p}$, where n_p is the number of predictions, and p_i contains mask (m_i) and box (b_i) predictions made on seen as well as unseen categories. For each bounding box $b_i{\in}p_i$, we first expand the width and height of the box by a factor $\delta(0{<}\delta{<}1)$ to compute a new, larger bounding box b'_i (Fig. 4a).

$$b_i : (x_i, y_i, h_i, w_i) \xrightarrow{\text{expand}} b'_i : (x_i, y_i, (1{+}\delta){*}h_i, (1{+}\delta){*}w_i)$$

where (x_i, y_i) is the center and (h_i, w_i) are the original height and width of box b_i. This inflation allows us to ingest useful context information around the part and the underlying object. Next, we compute the ROIAlign features for all the boxes $\{b'_i\}$ using the FPN feature map $\mathcal{F}$ resulting in a d-dim feature for each part-mask denoted using $\{f_{b,i}\}_{i=1}^{n_p}{\in}\mathbb{R}^d$. The pairwise affinity between two part-masks $(p_i, p_j) \in P$ is then computed using the cosine similarity between the corresponding feature maps (Fig. 4b).

$$\phi(p_i, p_j) = \frac{{f_{b,i}}^T \cdot f_{b,j}}{\|f_{b,i}\| \|f_{b,j}\|}; f_{b,i} = \text{RoIAlign}(\mathcal{F}, b'_i) \quad (1)$$

We visualize the parts retrieved using pairwise affinity for few examples in Fig. 6. While [54] has shown strong generalization of pixel pairwise affinities, we show that part pairwise affinities also generalize across object categories.

Affinity based grouping We use a clustering algorithm to merge parts based on the soft affinity scores given in Eq. (1).

Our clustering objective can be formulated as follows:

$$\max_{G} \sum_{k=1}^{|G|} \sum_{p_i, p_j \in g_k} \phi(p_i, p_j), \quad \text{s.t.} \sum_{k=1}^{|G|} |g_k| = n_p \qquad (2)$$

where G is a possible partition of the n_p predictions, $|G|$ denotes the total number of partitions and k^{th} partition in G is denoted by g_k ($1 \leq k \leq |G|$). In other words, given a set of elements along with their pairwise affinity scores, our clustering algorithm is employed to produce a partition of the elements that maximizes the average affinities within each resulting partition. We use an off-the-shelf agglomerative clustering algorithm [5] provided by `scikit-learn` [43]. It is parameter-free, lightweight, and fast, incurring minimum computation overhead in terms of time and memory even when clustering hundreds of part-masks in each iteration. In fact, as shown in Sec. 4.4 our final framework including the grouping and refinement modules adds negligible inference time overhead to the MaskRCNN backbone. We merge all the part masks (and boxes) within each partition group g_k to form more complete masks (and boxes), potentially corresponding to whole objects (output Fig. 3b). Since the original predictions in P might also correspond to whole objects on seen classes, we combine the originally detected masks as well as the grouped masks into our output at this stage.

3.3. Refinement Module

To handle the case where the resulting masks after grouping may not be sharp due to noisy initial segmentations, we employ a refinement module whose design follows the RoIHeads of Mask R-CNN, with inputs as the predictions obtained after the grouping stage (output Fig. 3c). The refinement head is trained only using annotated ground truth instances in S in order to induce the notion of object boundaries into the predictions, which is available only in the annotated masks. The backbone and refinement heads are then trained together in a single stage using the total losses obtained from the part-mask prediction and the refinement modules.

Objectness ranking Following [29], we add box and mask IoU branches to our RoIHead in part-mask predictions as well as refinement heads to compute the localization quality. IoU metrics are shown to improve objectness prediction [24] and avoid over-fitting to seen instances [29] when trained with non-exhaustive annotations. Our box and mask IoU heads follow the same architecture, with two fc-layers of 256-dim each followed by a linear layer to predict the IoU score, trained using an L1 loss for IoU regression.

Inference During inference (Fig. 2), we first predict part masks, followed by the affinity based grouping to hierarchically merge these into whole objects. We then pass these detections through the refinement layer to obtain the final predictions. We rank the predictions using the geometric mean of their predicted classification score c, box IoU b and mask IoU m from the refinement head as $s = \sqrt[3]{c * b * m}$.

Cross-category setting			
Train On	Test On	# Seen classes	# Unseen classes
VOC	Non-VOC	20	60
LVIS [19]	COCO [35]	1123	80
Cross-Dataset setting			
COCO [35]	UVO [55]	80	open
	ADE20k [59]		70
	OpenImagesV6 [8]		270

Table 1: **Evaluation settings**. Number of seen and unseen categories using in our cross-category and cross-dataset generalization evaluation settings.

4. Experiments

Datasets and evaluations. We demonstrate the effectiveness of UDOS for open-world instance segmentation under *cross-category* generalization within the same dataset, as well as *cross-dataset* generalization across datasets with different taxonomies. A summary of evaluation setups is presented in Tab. 1. For cross-category generalization we use the MS-COCO [35] dataset and train the model using categories from VOC and test on the remaining unseen nonVOC classes following prior work [29, 48, 54]. For cross-dataset generalization, we train on complete COCO dataset and directly test on validation splits of UVO [55], ADE20k [59] and OpenImagesV6 [8] datasets without any fine-tuning. Both UVO and ADE20k datasets provide exhaustive annotations in every frame, which is ideal to evaluate open world models, while OpenImagesV6 with 350 categories allows to test our open world segmentation approach on large scale datasets.

Implementation details. We use a standard Mask R-CNN model [20] with a ResNet-50-FPN [34] as our backbone network for top-down segmentation. We train UDOS using SGD for 10 epochs with an initial learning rate of 0.02 on 8 GPUs. To generate unsupervised masks for images in COCO dataset, we use selective search [52] for cross-category experiments and MCG [46] for cross-dataset experiments. Note that the mask proposals are required *only during training*, and *not required* during inference in cross-category or cross-dataset settings (Fig. 2). We follow prior works in open-world instance segmentation [29, 48, 54] and use average recall (AR) (between IoU thresholds of 0.5 to 1.0) as the evaluation metric. Since open world models generally detect many more objects in a scene than closed world models (see Fig. 5) and many datasets do not have exhaustive annotation, we use AR^{100} and AR^{300} as the evaluation metrics on both box (AR_B) and mask (AR_M) to avoid penalizing predictions of valid, yet unannotated, objects.

VOC→NonVOC	AR_B^{100}	AR_B^{300}	AR_M^{100}	AR_M^{300}
***Bottom-up**(No Training)*				
SS [52]	14.3	24.7	6.7	12.9
MCG [46]	23.6	30.8	19.4	25.2
***Top-down**(Class-agnostic Training)*				
MaskRCNN [20]	25.1	30.8	20.5	25.1
Mask R-CNN$_{SS}$	24.1	24.9	20.9	21.7
Mask R-CNN$_{SC}$	25.6	33.1	24.9	28
Open-World Methods				
OLN [29]	32.5	37.4	26.9	30.4
LDET [48]	30.9	38.0	26.7	32.5
GGN [54]	31.6	39.5	28.7	35.5
UDOS	**33.5**	**41.6**	**31.6**	**35.6**

Table 2: **Cross-category generalization evaluation on COCO dataset.** Training on 20 VOC categories of COCO and testing on 60 NonVOC categories. UDOS outperforms many competitive baselines as well as the current state-of-the-art GGN on the VOC→NonVOC setting.

4.1. Baselines

We use the following baselines for comparisons. (i) **Image-computable masks**: We directly use masks generated by **MCG** [46] and Selective Search [52] (**SS**), which are class-agnostic, learning-free proposal generation methods relying on low-level cues, for evaluation. (ii) **Mask-R-CNN** [20] denotes standard Mask R-CNN training in class-agnostic fashion only on the seen classes, (iii) **Mask R-CNN**$_{SS}$ indicates standard Mask R-CNN trained using selective search proposals as the supervision instead of the ground truth annotations, and (iv) **Mask R-CNN**$_{SC}$ denotes Mask R-CNN trained with BoxIoU and MaskIoU scoring to rank the proposals instead of the classification score.

We also compare with state of the art open-world instance segmentation algorithms OLN [29], LDET [48] and GGN [54]. For fair comparison with UDOS, we use the result from GGN [54] *without* the OLN backbone.

4.2. UDOS outperforms baselines on cross-category generalization

We show in Tab. 2 that both bottom-up grouping methods like SS or MCG that rely on low-level cues, and proposal based architectures [20] that train only on seen-class annotations are insufficient to reliably detect and segment unseen class instances. UDOS which jointly exploits complementary cues from ground truth annotation for seen classes as well as bottom-up unsupervised masks for unseen classes achieves Box AR of 33.5% and Mask AR of 31.6%, thus outperforming all the baselines by a significant margin, indicating the strength of our approach. UDOS also sets new state-of-the-art in cross-category open world instance segmentation, outperforming the current SOTA methods GGN [54] on both box and mask AR. Through numerous qualitative results for this setting in Fig. 5 and in the supplementary, we show how UDOS, unlike Mask R-CNN$_{SC}$, is able to efficiently detect many objects even when their annotations are not present in the training data.

LVIS→COCO	AR_B^{100}	AR_B^{300}	AR_M^{100}	AR_M^{300}
MaskRCNN	23.8	29.4	18.5	22.0
Mask R-CNN$_{SC}$	21.3	27.9	17.9	24.2
OLN [29]	28.5	38.1	23.4	27.9
UDOS	**33.2**	**42.2**	**26.3**	**32.2**

Table 3: **Cross-category generalization evaluation with large taxonomy**. All models are trained on 1123 categories from LVIS (excluding COCO categories), and evaluated on COCO 80 categories. UDOS outperforms OLN [29] by 4.7% and 2.9% on box and mask AR^{100}.

While a probable strategy to handle novel categories during test-time might be to build datasets with large taxonomies, we show in Tab. 3, by using LVIS [19] dataset for training, that this still doesn't achieve generalization to unseen categories. Specifically, we take masks for all the 1203 categories in LVIS, and we omit all annotations from training which have a IoU overlap of >0.5 with COCO masks to obtain 79.5k instance masks from LVIS. We then train on these annotations, and evaluate the transfer performance on the validation images from COCO dataset. As shown in Tab. 3, UDOS achieves 33.2% AR_B^{100} and 26.3% AR_M^{100} and significantly outperforms the baselines indicating the effectiveness of UDOS even on datasets with large vocabulary.

4.3. UDOS sets new SOTA on cross-dataset generalization

While the cross-category setting is useful to study the generalization ability from seen to unseen classes, a more realistic setting for understanding open world models would be to evaluate on both seen and unseen classes together, as the model needs to efficiently handle objects from any category during real world deployment. To reflect this, we present evaluations on in-the-wild target datasets like UVO, ADE20k and OpenImages which contain many objects not present in COCO categories. We reiterate that we neither use any fine-tuning on target dataset, nor use any target unlabeled data during training.

COCO to UVO The performance of UDOS on UVO dataset is presented in Tab. 4. Since UDOS is specifically designed to handle novel classes by leveraging low-level supervision in a top-down architecture, it achieves much better performance than other baselines even on a challenging dataset like UVO with exhaustive annotations for every object in the scene. Specifically, UDOS clearly outperforms baseline approaches like Mask R-CNN that overfits to the foreground instances (+5% AR_B^{100}). We also outperform OLN and LDET, and perform competitively with GGN, which is the current best method on the UVO benchmark under similar experimental setting.

COCO to ADE20K ADE20k [59] is a scene parsing benchmark consisting of annotations for both *stuff* (road, sky,

	COCO→UVO				COCO→ADE20K				COCO→OpenImages			
	AR_B^{100}	AR_B^{300}	AR_M^{100}	AR_M^{300}	AR_B^{100}	AR_B^{300}	AR_M^{100}	AR_M^{300}	AR_B^{100}	AR_B^{300}	AR_M^{100}	AR_M^{300}
MaskRCNN	47.7	50.7	41.1	43.6	18.6	24.2	15.5	20.0	57.1	59.1	55.6	57.7
Mask R-CNN$_{SS}$	26.8	31.5	25.1	31.1	18.2	25.0	17	21.6	34.0	42.7	33.1	38.8
Mask R-CNN$_{SC}$	42.0	50.8	40.7	44.1	19.1	25.6	18.0	22.0	54.1	59.1	54.2	57.4
OLN [29]	50.3	57.1	41.4	44.7	24.7	32.1	20.4	27.2	60.1	64.1	60.0	63.5
LDET [48]	52.8	58.7	43.1	47.2	22.9	29.8	19.0	24.1	59.6	63.0	58.4	61.4
GGN [54]	**52.8**	58.7	**43.4**	47.5	25.3	32.7	21.0	26.8	64.5	67.9	61.4	64.3
UDOS	52.7	**60.1**	43.1	**48.5**	**27.2**	**36.0**	**23.0**	**30.2**	**71.6**	**74.6**	**66.2**	**68.7**

Table 4: **Cross-dataset generalization evaluation for open world instance segmentation**. All models are trained on 80 COCO categories and evaluated on UVO (left), ADE20K (middle), OpenImages (right) as is without any fine-tuning.

Figure 5: **Visualization of segmentations for model trained only on VOC classes** from COCO dataset. The top row shows result using using Mask-RCNN$_{SC}$, second row shows output using UDOS and the third row shows some predictions made only by UDOS and missed by Mask-RCNN$_{SC}$. We also show the number of detections made by the network below each image. Starting from left most image, many classes like {jug, tissue papers, tie, eyeglasses}, {knife, cutting board, vegetables, glass}, {shoes, helmet, gloves}, {ostrich} and {dishwasher, faucet} among others which are not part of VOC-classes are missed by standard Mask-RCNN training, but detected using UDOS. More visualizations are provided in the supplementary.

floor etc.) and discrete *thing* classes. We regard each annotation mask as a separate semantic entity and compute the average recall (AR) on both in-taxonomy and out-of-taxonomy objects to evaluate the ability of trained models to detect thing classes and group stuff classes in images. From Tab. 4, we observe that UDOS achieves box AR100 of 27.2% and mask AR100 of 23.0%, which is higher than all baselines. We reiterate that this is truly in the wild evaluation with no fine-tuning on target ADE20k data.

COCO to OpenImagesV6 We consistently outperform all baselines as well as open-world methods like OLN and GGN by significant margins on the OpenImagesV6 dataset [8], as shown in Tab. 4. We achieve AR_B^{100} of 71.6%, which is better than the strongest baseline Mask R-CNN by 14.5% and current state-of-the-art GGN by 7.1%. Likewise, AR_M^{100} of 66.2% obtained by UDOS is 4.8% higher than GGN, setting new state of the art. Several visualizations of UDOS predictions on UVO, ADE20k and OpenImages datasets have been provided in the supplementary material.

4.4. Ablations

We use the VOC to NonVOC cross-category generalization on COCO dataset in the following ablations.

Refinement and grouping modules We show in Tab. 5a that without the proposed grouping and refinement modules, maskAR drops to 11.8% from 31.6%, as the masks are noisy and only correspond to parts of instances. Using a refinement module after grouping leads to more refined

Group	Refine	AR_B^{100}	AR_M^{100}
✗	✗	25.4	11.8
✓	✗	32.6	30.7
✓	✓	33.5	31.6

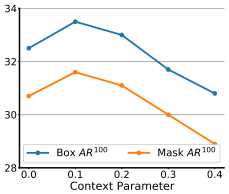
(a)

BoxIoU	MaskIoU	AR_B^{100}	AR_M^{100}
✗	✗	29.0	24.3
✓	✗	32.7	28.9
✗	✓	32.9	29.2
✓	✓	**33.5**	**31.6**

(b)

Segmentation	AR_B^{100}	AR_M^{100}
Uniform Grid	9.9	9.2
SSN [27]	19.4	18.7
Sel. Search [52]	**33.5**	**31.6**
MCG [46]	32.4	29.4

(c)



(d)

Table 5: **Ablation results**. Effect of (a) grouping and refinement modules, (b), boxIoU and maskIou losses during training, (c) segmentation algorithm and (d) context dilation parameter δ on the VOC→NonVOC performance.

(a) part prediction **(b) query mask** **(c) retrieved part masks with highest affinity** **(d) grouped output**

Figure 6: **Visualization of pairwise affinity maps and grouped predictions.** Given a part mask as a query, we show the 3 nearest part masks of the query using our pairwise affinity. The images are taken from UVO dataset, and the affinity is computed using UDOS model trained on COCO. Our affinity-based grouping module correctly groups parts into whole instances even with unseen objects. The last row visualizes a failure case where the model retrieves a part mask from a neighboring instance.

masks further improving the performance.

Choice of proposal ranking We show the importance of using BoxIoU and MaskIoU scoring functions in Tab. 5b, where significant drops in AR100 are observed without the use of both the scoring functions, validating the observations in prior works [29] that scoring module prevents overfitting and improves open world learning.

Influence of δ Intuitively, a small value of delta would not capture sufficient context around the region for extracting similarity while a very high value of δ would induce noisy features from different neighboring objects. In Tab. 5d, we show that a value of 0.1 achieves an optimum trade-off, so we use the same value of $\delta = 0.1$ in all our experiments.

Choice of proposal generation methods From Tab. 5c, we show that a naive segmentation of image using uniform grids by extracting 64×64 patches from the image expectedly performs worse, as the part masks generated using uniform grids do not semantically correspond to object parts. We also use super-pixels generated from SSN [27], but found that bottom-up supervision generated from image-based segmentation algorithms like SS or MCG lead to much better accuracies.

Visualizations of affinity maps In Fig. 6, we present 3-nearest part masks retrieved for a given query mask using their affinity (Eq. (1)) and the grouped outputs. We observe that different part masks of the same entity are often retrieved with high affinity, using our grouping module.

Inference time comparison Our grouping module is lightweight and adds negligible run-time overhead. Specifically, at 100 output proposals, MaskRCNN [20] and GGN [54] take 0.09s/im, MaskRCNN$_{SC}$ and OLN [29] take 0.12s/im while UDOS takes 0.13s/im (+0.01s/im) with stronger performance. Generating part-masks using selective search for the complete COCO [35] dataset takes around 1.5 days on a 16-core CPU, but we reiterate that the part-masks only need to be generated once before training and are not needed during testing/deployment (Fig. 2).

5. Discussion

While several prior works independently explored advances in top-down architectures and bottom-up proposal generation towards closed-world instance segmentation, to our best knowledge, the proposed UDOS is the first to integrate these ideas into a unified framework for open-world instance segmentation with SOTA results. Our grouping and refinement modules efficiently convert part mask predictions into complete instance masks for both seen and unseen objects, making UDOS distinct from prior closed-world as well as open-world segmentation methods. Furthermore, extensive experiments demonstrate the significant improvements achieved by UDOS over these prior works, setting new state-of-the-arts across 5 different chal-

lenging datasets including COCO, LVIS, ADE20k, UVO, and OpenImages. In terms of limitations, although UDOS showcases excellent generalization capability in handling open world categories, it is difficult to aggregate masks effectively for complex scenarios such as different objects with similar appearance, which we plan to address in a future work by incorporating learnable grouping module into our framework.

References

[1] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4981–4990, 2018. 3

[2] Pablo Arbelaez. Boundary extraction in natural images using ultrametric contour maps. In *CVPR Workshops*, 2006. 2

[3] Anurag Arnab and Philip HS Torr. Pixelwise instance segmentation with a dynamically instantiated network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 441–450, 2017. 2

[4] Alberto Bailoni, Constantin Pape, Steffen Wolf, Thorsten Beier, Anna Kreshuk, and Fred A Hamprecht. A generalized framework for agglomerative clustering of signed graphs applied to instance segmentation. *arXiv preprint arXiv:1906.11713*, 2019. 3

[5] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine learning*, 56(1):89–113, 2004. 5

[6] Amir Bar, Xin Wang, Vadim Kantorov, Colorado J Reed, Roei Herzig, Gal Chechik, Anna Rohrbach, Trevor Darrell, and Amir Globerson. Detreg: Unsupervised pretraining with region priors for object detection. *arXiv preprint arXiv:2106.04550*, 2021. 3

[7] 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

[8] Rodrigo Benenson, Stefan Popov, and Vittorio Ferrari. Large-scale interactive object segmentation with human annotators. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11700–11709, 2019. 5, 7, 14

[9] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9157–9166, 2019. 2

[10] 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

[11] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 2

[12] Achal Dave, Pavel Tokmakov, and Deva Ramanan. Towards segmenting anything that moves. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 3

[13] Yuming Du, Yang Xiao, and Vincent Lepetit. Learning to better segment objects from unseen classes with unlabeled videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3375–3384, 2021. 3

[14] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. Ieee, 2008. 2

[15] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2009. 2

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

[17] Matthias Grundmann, Vivek Kwatra, Mei Han, and Irfan Essa. Efficient hierarchical graph-based video segmentation. In *2010 ieee computer society conference on computer vision and pattern recognition*, pages 2141–2148. IEEE, 2010. 2

[18] 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

[19] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5356–5364, 2019. 5, 6

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE*

international conference on computer vision, pages 2961–2969, 2017. 1, 2, 3, 5, 6, 8

[21] Olivier J Hénaff, Skanda Koppula, Jean-Baptiste Alayrac, Aaron van den Oord, Oriol Vinyals, and João Carreira. Efficient visual pretraining with contrastive detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10086–10096, 2021. 3

[22] Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10951–10960, 2020. 3

[23] Ronghang Hu, Piotr Dollár, Kaiming He, Trevor Darrell, and Ross Girshick. Learning to segment every thing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4233–4241, 2018. 3

[24] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring r-cnn. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6409–6418, 2019. 5

[25] Jyh-Jing Hwang, Stella X Yu, Jianbo Shi, Maxwell D Collins, Tien-Ju Yang, Xiao Zhang, and Liang-Chieh Chen. Segsort: Segmentation by discriminative sorting of segments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7334–7344, 2019. 3

[26] Suyog Dutt Jain, Bo Xiong, and Kristen Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pages 2117–2126. IEEE, 2017. 3

[27] Varun Jampani, Deqing Sun, Ming-Yu Liu, Ming-Hsuan Yang, and Jan Kautz. Superpixel sampling networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 352–368, 2018. 8

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

[29] Dahun Kim, Tsung-Yi Lin, Anelia Angelova, In So Kweon, and Weicheng Kuo. Learning open-world object proposals without learning to classify. *IEEE Robotics and Automation Letters*, 2022. 2, 3, 5, 6, 7, 8

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

[31] Shu Kong and Charless C Fowlkes. Recurrent pixel embedding for instance grouping. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9018–9028, 2018. 3

[32] Weicheng Kuo, Anelia Angelova, Jitendra Malik, and Tsung-Yi Lin. Shapemask: Learning to segment novel objects by refining shape priors. In *Proceedings of the ieee/cvf international conference on computer vision*, pages 9207–9216, 2019. 3

[33] W. Kuo, B. Hariharan, and J. Malik. Deepbox: Learning objectness with convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2479–2487, Los Alamitos, CA, USA, dec 2015. IEEE Computer Society. 2

[34] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 3, 5

[35] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2, 5, 8

[36] Shu Liu, Jiaya Jia, Sanja Fidler, and Raquel Urtasun. Sgn: Sequential grouping networks for instance segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3496–3504, 2017. 3

[37] Yiding Liu, Siyu Yang, Bin Li, Wengang Zhou, Jizheng Xu, Houqiang Li, and Yan Lu. Affinity derivation and graph merge for instance segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 686–703, 2018. 3

[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] 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

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

[41] Tiange Luo, Kaichun Mo, Zhiao Huang, Jiarui Xu, Siyu Hu, Liwei Wang, and Hao Su. Learning to group: A bottom-up framework for 3d part discovery in unseen categories. *arXiv preprint arXiv:2002.06478*, 2020. 3

[42] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE international conference on computer vision*, pages 4990–4999, 2017. 2

[43] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 5

[44] Trung Pham, Thanh-Toan Do, Gustavo Carneiro, Ian Reid, et al. Bayesian semantic instance segmentation in open set world. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–18, 2018. 3

[45] Pedro O Pinheiro, Ronan Collobert, and Piotr Dollár. Learning to segment object candidates. In *Advances in neural information processing systems*, volume 28, 2015. 2, 3

[46] Jordi Pont-Tuset, Pablo Arbelaez, Jonathan T Barron, Ferran Marques, and Jitendra Malik. Multiscale combinatorial grouping for image segmentation and object proposal generation. *IEEE transactions on pattern analysis and machine intelligence*, 39(1):128–140, 2016. 1, 2, 4, 5, 6, 8

[47] 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. 2

[48] Kuniaki Saito, Ping Hu, Trevor Darrell, and Kate Saenko. Learning to detect every thing in an open world. *arXiv preprint arXiv:2112.01698*, 2021. 2, 5, 6, 7

[49] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9339–9347, 2019. 2

[50] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000. 2

[51] Mennatullah Siam, Alex Kendall, and Martin Jagersand. Video class agnostic segmentation benchmark for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2825–2834, 2021. 3

[52] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013. 2, 4, 5, 6, 8

[53] 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

[54] Weiyao Wang, Matt Feiszli, Heng Wang, Jitendra Malik, and Du Tran. Open-world instance segmentation: Exploiting pseudo ground truth from learned pairwise affinity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4422–4432, 2022. 2, 3, 4, 5, 6, 7, 8

[55] Weiyao Wang, Matt Feiszli, Heng Wang, and Du Tran. Unidentified video objects: A benchmark for dense, open-world segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10776–10785, 2021. 2, 3, 5, 12, 14

[56] Christopher Xie, Yu Xiang, Arsalan Mousavian, and Dieter Fox. Unseen object instance segmentation for robotic environments. *IEEE Transactions on Robotics*, 37(5):1343–1359, 2021. 2

[57] Xingqian Xu, Mang Tik Chiu, Thomas S Huang, and Honghui Shi. Deep affinity net: Instance segmentation via affinity. *arXiv preprint arXiv:2003.06849*, 2020. 3

[58] Xiao Zhang and Michael Maire. Self-supervised visual representation learning from hierarchical grouping. *Advances in Neural Information Processing Systems*, 33:16579–16590, 2020. 3

[59] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019. 5, 6, 14

A. Weight sharing on refinement module

In UDOS, our design of the refinement module follows the RoI-heads of Mask R-CNN architecture. Specifically, our box and mask prediction heads in the refinement head have the same architecture as the box and mask prediction heads of the Mask R-CNN. Therefore, one option is to share the weights between the RoIHeads in the first stage Mask R-CNN and the refinement heads during training. However, we note from Tab. 6 that such an approach of weight-sharing between the two stages of UDOS hurts the mask

Table 6: **Effect of weight sharing between RoI head and refinement head.** Comparison of results with and without sharing parameter weights between RoI module of part mask prediction head and the refinement head. Using separate heads for the RoI head of part-mask prediction module and refinement module improves AR_M^{100} by 3.3%.

	AR_B^{100}	AR_B^{300}	AR_B^s	AR_B^m	AR_B^l	AR_M^{100}	AR_M^{300}	AR_M^s	AR_M^m	AR_M^l
Shared weights	33.5	41.3	13.2	43.6	60.6	28.3	33.6	11.4	37.7	48.4
Non shared weights	33.5	41.6	16.2	43.9	53.5	31.6	35.6	15.2	42.7	48.3

accuracy on the cross-category detection on COCO. This is because the goals of prediction in the two stages are different. While the RoIHeads in the first stage are trained to predict part-masks and trained on weak supervision from bottom-up segmentation algorithms, the refinement head is trained only using ground truth annotations and is used to predict the final object boxes and masks. However, this improvement also comes with additional increase in model parameters from 57.4M to 86.5M. Also, we observed that using individual weights only benefits cross-category setting, while cross-dataset benefits from shared weights between the part-mask MaskRCNN and refinement head.

B. Visualizing outputs at each stage of UDOS

We visualize the outputs after each stage of UDOS for cross-category VOC to NonVOC setting in Fig. 7 and for cross-dataset setting in Fig. 8. We illustrate the effect of our part-mask prediction module in in generating the segmentation masks for parts of objects, rather than the whole objects. This enables us to detect a much larger taxonomy of objects than what are present in the annotated concepts. For example, in Fig. 7 for the case of cross-category transfer setting from VOC $\rightarrow$ Non-VOC, *tie* and *shoe* are not one of the annotated classes. Yet, our model effectively retrieves these from the image, instead of considering it a background or combining it with the boy. Our grouping module, powered by the context aggregation, then effectively groups the various part masks predicted on the *pot*, *boy* and *tie*. Note that the accuracy of predictions obtained by directly merging the part masks might be limited due to noisy part mask supervision, which are further corrected by our refinement layer. Similar observations for the cross-dataset setting are presented in Fig. 8.

C. Qualitative comparisons

In addition to the comparisons provided in the main paper, we provide more comparisons of predictions made by UDOS and Mask R-CNN$_{SC}$ in Fig. 9 for the setting where we train only using VOC categories from COCO. We also show the predictions made on the cross-dataset setting, by using a model trained on all COCO categories and testing on images from UVO [55] in Fig. 10. In each case, we also show the predictions made *only* by UDOS and missed by Mask R-CNN$_{SC}$ (highlighted in yellow), indicating the utility of our approach on open world instance segmentation.

For instance, in the second column in Fig. 9, the predictions made by Mask R-CNN$_{SC}$ do not include objects like *keyboard*, *joystick*, *glass* and *speaker* which are efficiently retrieved by UDOS. Also note that the number of predictions made by UDOS is always higher than Mask R-CNN$_{SC}$ for both cases of cross-category transfer in Fig. 9 as well as cross-dataset transfer in Fig. 10.

(a) Input Image (b) part-mask output (c) Grouped prediction (d) Refined prediction

Figure 7: **Visualizing outputs after each stage of UDOS for cross-category training**. All images belong to the COCO dataset, and outputs are generated using a model trained only on VOC categories. (a) shows the input image, followed by (b) Part-mask prediction, (c) grouped outputs using our affinity based grouping and (d) refined prediction. The masks in last two columns correspond to true-positives with respect to the ground truth. Note that classes such as *pot, van, elephant*, and *auto-rickshaw* do not belong to any of the training VOC categories. Also note that the merged outputs might be noisy due to the imperfection in the initial part-mask supervision used, which are corrected by our refinement layer.

(a) Input Image (b) part-mask output (c) Grouped prediction (d) Refined prediction

Figure 8: **Visualizing outputs after each stage of UDOS for cross-dataset training**. The images in the three rows belong to OpenImages [8], UVO [55] and ADE20K [59] datasets respectively. All outputs are generated by model trained on complete COCO dataset. (a) shows the input image, followed by (b) Part-mask prediction, (c) grouped outputs using our affinity based grouping and (d) refined prediction. The masks in last two columns correspond to true-positives with respect to the ground truth.

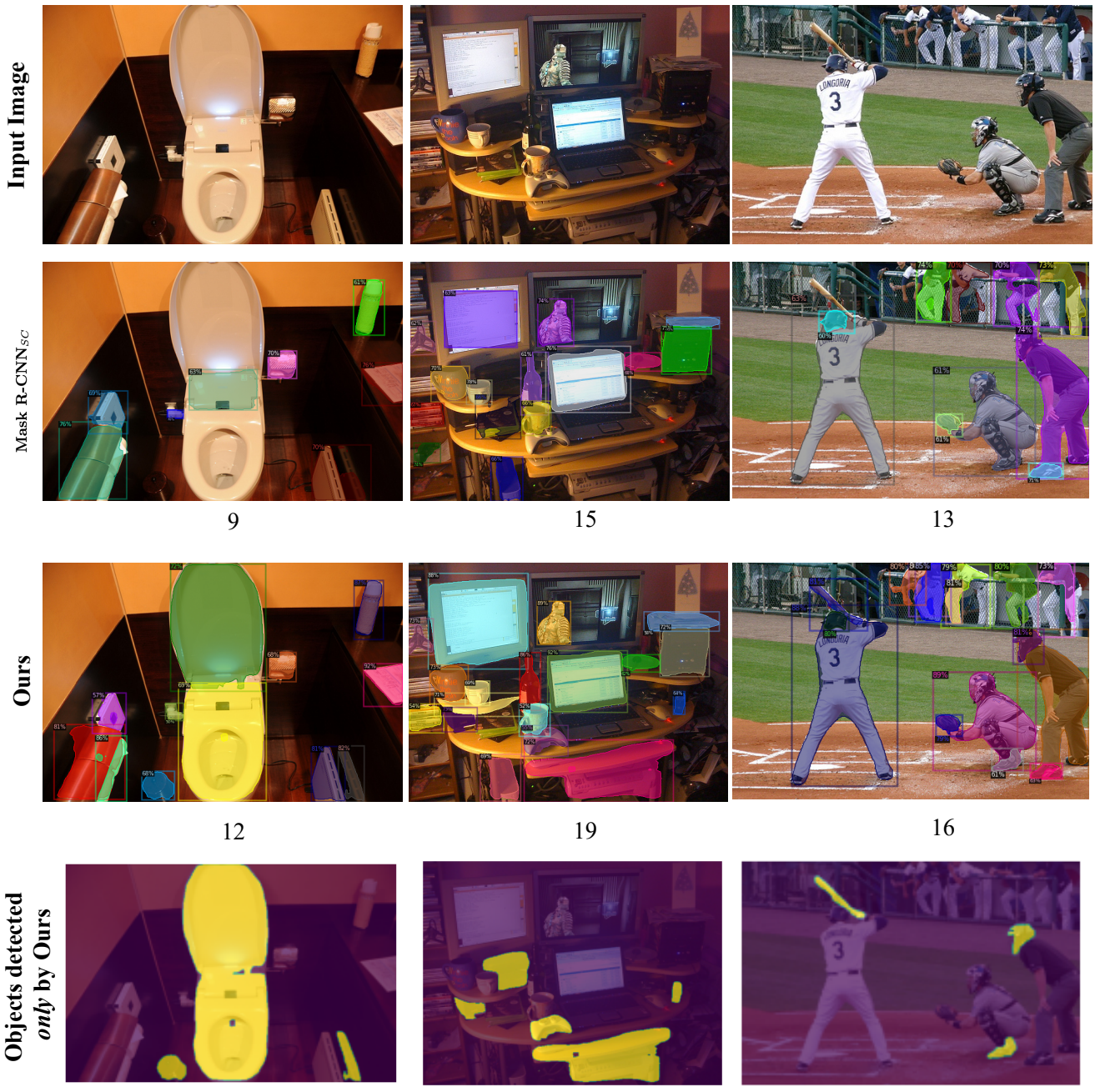


Figure 9: **Visualization of segmentations for model trained only on VOC classes** from COCO dataset. For various input images given in the first row, the second row shows result using Mask-RCNN$_{SC}$, third row shows output using UDOS and the fourth row shows some predictions made only by UDOS and missed by Mask-RCNN$_{SC}$ on these images. We also show the number of detections made by the network below each image. All images belong to COCO dataset.

Figure 10: **Visualization of segmentations for model trained on all COCO classes**. For various input images given in the first row, the second row shows result using Mask-RCNN$_{SC}$, third row shows output using UDOS and the fourth row shows some predictions made only by UDOS and missed by Mask-RCNN$_{SC}$ on these images. We also show the number of detections made by the network below each image. All images belong to UVO dataset.