


SOS: Segment Object System for Open-World Instance Segmentation With Object Priors

Supplementary Material

Christian Wilms¹, Tim Rolff^{1,2}, Maris Hillemann¹, Robert Johanson¹, and
Simone Frintrop¹

¹ Computer Vision Group, University of Hamburg, Germany

² Human-Computer Interaction Group, University of Hamburg, Germany
`{firstname.lastname}@uni-hamburg.de`

1 Details of Object Priors

In the following, we provide implementation details of all object priors evaluated in this paper.

Grid The object prior *Grid* utilizes the baseline implemented in SAM³, however, using 64 points per side leading to a 64×64 grid as in the zero-shot object proposals experiment described in [11]. Given an image of size $h \times w$, the grid points have a spacing of $\frac{h}{64}$ in the vertical dimension and a spacing of $\frac{w}{64}$ in the horizontal dimension. All grid points are translated by $\frac{h}{64} \cdot \frac{1}{2}$ in the horizontal dimension and $\frac{w}{64} \cdot \frac{1}{2}$ in the vertical dimension, to center the grid on the image. Note that we directly use the grid points as prompts, without the sampling utilized for most other priors.

Dist For the *Dist* object prior, we first extract the centroid of each annotated VOC object from the COCO train split. As the centroid, we take the average x - and y -coordinates of all pixels belonging to the object. Subsequently, we normalize the coordinates by the image height and width, leading to relative coordinates. Given these relative coordinates, we aggregate all centroids across the dataset in an array of size 64×64 . Finally, each entry in the aggregator represents the absolute number of objects with a centroid at this location, given a resolution of 64×64 . To generate the object prior per image, we resize the aggregator to the respective image’s size, ignoring the image content.

GT To create the object prior *GT*, we extract the centroid of each *non-VOC* object from the COCO train split and directly take these centroids as prompt.

³ https://github.com/facebookresearch/segment-anything/blob/main/segment_anything/automatic_mask_generator.py

As the centroid, we take the average x - and y -coordinates of all pixels belonging to the object. Note that *GT* is the only object prior that accesses information from the unknown non-VOC classes in this study. Hence, it serves as an upper bound rather than a practical object prior.

Spx We create the *Spx* object prior by first applying the superpixel segmentation method FH [3] to the image. In its original formulation, FH has only one parameter (k) to indirectly control the number of superpixels. We use $k = 10000$ in our experiments, leading to an average of 33.4 superpixels per image. Since we use the implementation in *scikit-image*⁴, additional pre- and post-processing steps apply, including Gaussian smoothing and the removal of tiny superpixels. Note that we use default parameters for all these steps. After the superpixels are generated, we extract the centroid per superpixel as point prompts. Similar to *GT* and *Dist*, we take the average x - and y -coordinates of all pixels belonging to the superpixel as centroid.

Contour For the *Contour* object prior, we follow [18]. First, we extract an edge map from the image using the SE edge detector [2]. Subsequently, we apply strong Gaussian smoothing ($\sigma = 20$) to the edge map. This yields a weighted edge strength per pixel and represents the density of edges around each pixel. Hence, we take this result as object prior.

VOCUS2 To generate the *VOCUS2* object prior, we apply the VOCUS2 system by [4] to each image. Since VOCUS2 has several parameters steering the center-surround contrast calculation, we use the default parameters⁵ for the Coffee Machine Sequence dataset [7], focusing on small objects. The resulting saliency map of VOCUS2 is the object prior.

DeepGaze The object prior *DeepGaze* utilizes the pre-trained system DeepGaze IIE [13]. We use the official⁶ model pre-trained on the datasets SALI-CON [8] and MIT1003 [9] for eye fixation prediction, including a center bias. The output of DeepGaze IIE is the object prior.

CAM We create the object prior *CAM*, using the CAMs [19] of a pre-trained classifier. As our classifier, we choose a ResNet-50 [6] pre-trained on ImageNet [15]. Hence, no training on the COCO dataset is conducted. Given an input image, we select the class with the highest probability returned by the classifier and produce a CAM for this class, if the probability is greater than 0.2 to remove uncertain classifier results. The generated CAM is the object prior.

⁴ <https://scikit-image.org/docs/stable/api/skimage.segmentation.html>

⁵ https://github.com/GeeeG/VOCUS2/blob/master/cfg/coffee_cfg.xml

⁶ <https://github.com/matthias-k/DeepGaze>

DINO For the object prior *DINO* we follow [1] and utilize a ViT-S backbone pre-trained on ImageNet [15] with a patch size of 8 in the self-supervised DINO framework. From the six attention heads of the network’s last transformer layer, we extract the self-attention of the CLS token. We combine these six self-attention maps by taking the per-pixel maximum to create the final object prior.

U-Net To generate the *U-Net* object prior, we train a U-Net [14] on the COCO training images with a joined mask covering all VOC objects as target. The U-Net encoder consists of four stages with two 3×3 convolution layers each using 64, 128, 256, and 512 filters, respectively. The bottleneck consists of another two 3×3 convolution layers with 1024 filters each. Finally, the decoder’s structure is set up to match the encoder also including skip connections between the respective stages. The final layer of the network is a 1×1 convolution with one filter to produce the output.

The network is trained for up to 20 epochs with early stopping using binary cross entropy loss and Adam optimizer. All input images are rescaled to 512×512 and the batch size is 4. The training data is generated from the original annotations of the VOC objects in the COCO training set. For each image, we join all masks of VOC objects resulting in a binary segmentation. These binary segmentations serve as the training target for the U-Net.

To create the object prior, the U-Net processes each image of the COCO training set. The resulting output map with per-pixel logits is the object prior.

2 Additional Object Prior Results

This section presents additional qualitative results of the proposed object priors with resulting pseudo annotations. Moreover, we present detailed quantitative results of SOS using all proposed object priors w.r.t. object sizes and classes.

Qualitative Results of Object Priors Figure 1 depicts examples of all object priors, except *GT*, on the COCO training set. While most object priors highlight similar areas, some differences are clearly visible. The baseline object prior *Dist* only resizes the spatial distribution of VOC objects in the COCO training set to the image size, ignoring the image content. Similarly, *Grid* rescales the 64×64 grid to the image size. In contrast, *Spx* and *Contour* focus more on the objects but also on highly textured background areas like the trees in the first example. *VOCUS2* highlights objects, but also salient background regions as visible in the fourth example. The second saliency-based object prior, *DeepGaze*, has a much stronger focus on objects, but mainly focuses on faces and has difficulties detecting secondary objects like the elephant in the second example. Similarly, *CAM* focuses on the objects, but at a very coarse resolution. The *DINO* object prior only highlights a few locations, however, covering most objects in the images. This leads to pseudo annotations for a variety of objects. In contrast, the *U-Net* object prior highlights entire objects.

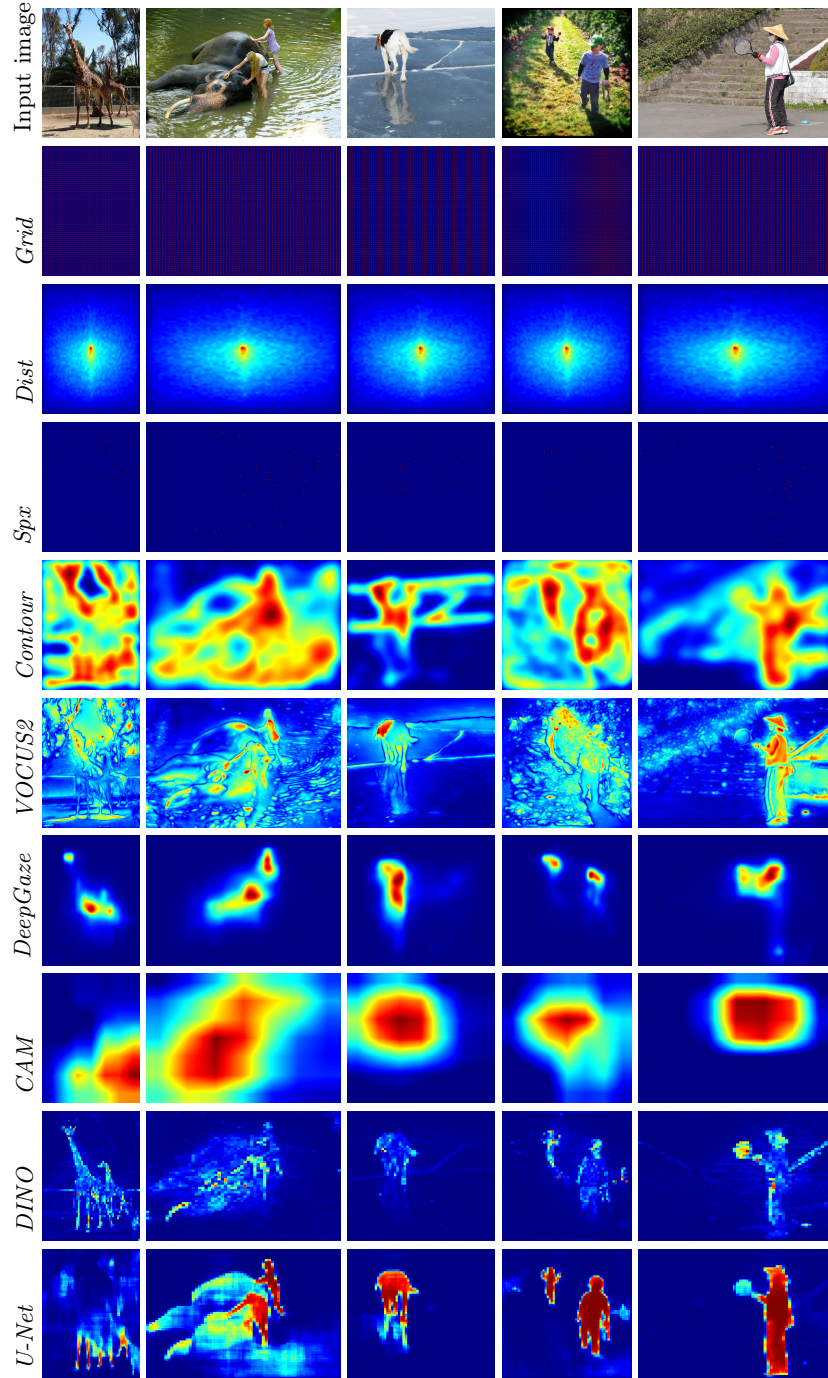


Fig. 1: Examples of all object priors evaluated in this paper as heatmaps.



Fig. 2: Example pseudo annotations generated in SOS based on all object priors evaluated in this paper.

Qualitative Results of Pseudo Annotations In Fig. 2, pseudo annotations for all object priors and example images presented in Fig. 1 are given. The pseudo annotations reflect the findings on the level of the object priors. Most notably, all priors, except for *DINO*, consistently lead to several background pseudo annotations, while missing foreground objects.

Size-specific Results of SOS with Various Object Priors To better assess the strengths and weaknesses of the object priors, Tab. 1 presents the size-specific results of SOS using each object prior in the COCO (VOC) \rightarrow COCO (non-VOC) setting. This is the same setting used in the object prior study in the main paper. The size-specific results assign each annotated object to the class S (small object, $\text{area} < 32^2$), M (medium object, $32^2 \leq \text{area} < 96^2$), or L (large object, $96^2 \leq \text{area}$), as defined by [12]. For the subsequent analysis, we ignore the *GT* object prior results as they use the original annotations’ centroids of the unknown classes, resulting in an upper bound.

The results show that SOS with *DINO* object prior outperforms all other variations on all but one measure (AR_{100}^M). For small objects, *DINO* and *Spx* lead to the highest recall, while *DINO* also generates the best results in precision. Medium and large objects are recalled by most methods on a similar level. However, *DINO* produces the best precision results. Hence, *DINO*’s success is driven by strong results on small objects and high precision across all object sizes.

Table 1: Class-specific results of SOS using all object priors presented in this paper in the COCO (VOC) \rightarrow COCO (non-VOC) setting *: Uses ground truth of unknown classes.

Object Prior	Small objects			Medium objects			Large objects		
	AP^S	AR_{100}^S	F_1^S	AP^M	AR_{100}^M	F_1^M	AP^L	AR_{100}^L	F_1^L
<i>Grid</i>	0.8	20.6	1.5	5.3	46.9	9.5	7.7	53.3	13.5
<i>Dist</i>	1.0	15.2	1.9	3.5	31.0	6.3	8.3	47.3	14.1
<i>GT</i>	9.2*	29.9*	14.1*	22.2*	50.7*	30.9*	29.8*	55.8*	38.9*
<i>Spx</i>	1.3	34.8	2.5	6.2	44.9	10.9	12.9	51.0	20.6
<i>Contour</i>	1.3	20.8	2.4	6.9	48.3	12.1	11.8	50.9	19.2
<i>VOCUS2</i>	1.1	21.7	2.1	6.2	48.5	11.0	15.0	54.1	23.5
<i>DeepGaze</i>	0.7	20.4	1.4	6.0	45.6	10.6	13.9	53.1	22.0
<i>CAM</i>	1.3	20.3	2.4	7.1	47.6	12.4	11.2	53.7	18.5
<i>DINO</i>	2.2	38.1	4.2	8.7	48.1	14.7	22.6	55.5	32.1
<i>U-Net</i>	1.4	21.3	2.6	7.3	49.4	12.7	18.5	51.9	27.3

Class-specific Results of SOS with Various Object Priors Table 2 presents the class-specific AR_{100} results of SOS using all introduced object priors

in the COCO (VOC) \rightarrow COCO (non-VOC) setting. This is the same setting as used in the object prior study in the main paper. Note that measuring precision in this experiment is not useful, as no classification of the detections is done in OWIS. For the subsequent analysis, we ignore the *GT* object prior results as *GT* uses the original annotations' centroids of the unknown classes, resulting in an upper bound.

The results show that SOS using the *DINO* object prior performs best on most classes. Overall, all object priors behave similarly on different levels of AR across most classes. However, on classes like **Sports balls** or **Toaster**, some object priors perform much worse compared to the others. Most notably, SOS using *VOCUS2* only produces an AR_{100} of 40.4 for class **Toaster**, which is much lower than SOS with *DINO* object prior (70.0). Another deviation from the general per-class trend is visible in the results of *Dist*. Several classes including **Traffic light**, **Snowboard**, or **Mouse** are not covered well by SOS with *Dist* object prior, due to their mostly off-center location.

Across all object priors, animal classes and traffic-related classes lead to strong results. Conversely, elongated objects like **Skis**, **Fork**, or **Knife** result in low AR_{100} scores. Generally, more research w.r.t. object properties is necessary to better understand the per-class differences.

Table 2: Class-specific AR_{100} results of SOS using all object priors presented in this paper in the COCO (VOC) \rightarrow COCO (non-VOC) setting. *: Uses ground truth of unknown classes. *V2* and *DG* denote *VOCUS2* and *DeepGaze*.

Class	<i>Grid</i>	<i>Dist</i>	<i>GT</i>	<i>Spx</i>	<i>Contour</i>	<i>V2</i>	<i>DG</i>	<i>CAM</i>	<i>DINO</i>	<i>U-Net</i>
Truck	52.7	50.5	55.5*	52.6	52.9	51.9	51.0	53.5	53.5	52.9
Traffic light	25.3	13.3	35.4*	24.7	23.6	27.0	22.3	20.5	27.1	19.5
Fire hydrant	64.7	56.7	66.0*	62.7	63.0	63.6	63.2	62.4	63.9	64.2
Stop sign	68.3	51.9	72.3*	68.5	66.1	67.7	66.7	67.1	69.5	65.9
Parking m.	56.8	52.8	59.7*	59.8	57.7	58.8	56.7	62.0	61.5	60.2
Bench	20.7	17.1	21.1*	20.0	21.1	19.8	19.2	19.6	19.3	21.2
Elephant	59.6	57.9	61.7*	58.4	58.9	59.3	59.4	60.0	59.4	62.2
Bear	74.5	72.8	74.8*	71.7	70.8	72.5	73.0	74.1	73.1	73.2
Zebra	52.8	47.9	56.6*	54.8	53.0	54.4	53.7	53.8	55.5	55.0
Giraffe	47.7	43.8	50.8*	49.3	47.6	49.6	48.3	49.2	50.0	48.3
Backpack	25.1	21.6	31.9*	24.3	26.4	26.8	26.5	24.8	25.3	29.8
Umbrella	48.8	31.8	50.2*	45.7	49.9	49.6	47.0	46.5	46.6	49.1
Handbag	26.4	17.4	32.3*	24.9	28.0	27.7	23.7	27.2	25.2	30.6
Tie	24.6	19.3	35.2*	23.5	26.3	24.4	25.1	22.8	25.3	27.2
Suitcase	45.5	41.5	50.6*	41.9	49.3	49.6	49.3	49.4	47.9	53.6
Frisbee	67.4	51.7	68.1*	67.0	63.5	69.4	66.4	64.7	68.2	67.7
Skis	2.7	1.4	3.4*	3.1	3.1	3.2	2.5	3.2	2.8	3.0
Snowboard	21.0	10.6	22.8*	22.0	23.6	23.6	20.9	21.3	23.5	23.3
Sports ball	46.7	22.7	51.7*	45.7	35.2	45.0	37.3	39.8	47.2	37.5
Kite	42.6	30.1	45.5*	44.0	38.4	42.4	42.8	39.2	44.2	40.6

Baseball bat	20.8	15.7	29.1*	21.5	26.5	24.6	23.4	23.5	28.1	24.6
Baseball gl.	40.8	35.4	46.8*	39.4	44.7	42.5	45.3	42.3	44.7	48.9
Skateboard	17.0	16.4	24.9*	19.3	20.2	18.6	19.6	18.3	22.2	21.3
Surfboard	31.6	26.6	34.5*	30.5	33.4	34.2	33.8	33.1	33.6	34.1
Tennis racket	43.5	31.8	52.8*	44.4	44.7	42.3	44.7	46.5	48.3	45.0
Wine glass	26.7	20.5	35.7*	26.3	25.9	21.9	24.1	25.5	25.6	25.6
Cup	49.6	36.2	57.0*	47.4	49.0	47.7	45.6	49.9	48.3	49.7
Fork	8.2	5.5	16.4*	10.6	11.3	10.3	8.8	8.8	12.4	11.6
Knife	12.9	6.8	19.1*	14.0	15.4	14.4	11.2	14.5	14.1	15.8
Spoon	17.4	7.1	23.7*	18.9	18.7	19.0	15.2	17.3	19.9	18.9
Bowl	44.2	29.7	46.8*	38.5	40.9	47.7	41.6	47.8	45.2	42.3
Banana	32.1	19.9	34.6*	24.1	30.9	32.1	31.8	31.2	34.0	32.4
Apple	39.3	23.9	44.0*	33.6	36.7	40.6	36.9	38.6	39.5	41.7
Sandwich	40.5	35.1	50.1*	35.3	36.2	41.2	40.2	41.9	48.2	39.4
Orange	46.6	27.6	52.1*	39.1	46.3	49.8	46.6	46.7	47.6	48.9
Broccoli	26.1	22.4	38.5*	28.0	32.8	33.4	33.1	31.7	37.0	33.0
Carrot	34.8	22.5	39.5*	22.8	35.1	37.3	35.3	33.6	32.8	36.9
Hot dog	26.2	24.7	37.3*	20.6	24.6	27.1	30.0	26.2	31.2	27.5
Pizza	49.0	37.4	56.1*	38.0	41.7	51.2	50.4	47.2	54.9	42.4
Donut	51.8	32.6	60.7*	41.0	51.8	56.7	51.2	54.0	53.8	56.0
Cake	43.5	31.5	51.5*	41.1	41.9	44.3	42.3	45.2	45.8	43.9
Bed	30.9	29.3	29.8*	30.6	28.8	32.5	28.9	32.2	29.3	27.7
Toilet	47.8	42.4	58.7*	48.6	46.4	48.8	46.0	49.0	59.3	46.3
Laptop	52.5	48.9	55.2*	52.5	53.6	52.8	49.8	54.9	55.8	52.5
Mouse	60.2	25.6	68.5*	61.8	58.3	61.7	57.4	62.3	60.5	61.6
Remote	29.1	21.3	43.5*	34.1	31.4	33.3	29.5	31.6	35.5	35.1
Keyboard	51.6	41.4	54.6*	48.2	52.8	49.7	55.0	50.8	54.7	54.8
Cell phone	38.6	31.3	46.4*	39.5	40.2	40.4	40.0	40.2	44.1	41.4
Microwave	62.0	51.8	61.5*	58.4	65.5	62.7	61.6	65.8	62.0	57.6
Oven	29.0	26.4	33.1*	30.4	30.5	31.7	31.0	32.2	34.1	30.9
Toaster	65.6	58.9	70.0*	74.4	68.9	40.4	72.2	76.7	70.0	68.9
Sink	42.2	29.8	47.9*	39.0	42.7	44.5	42.7	44.6	43.6	45.0
Refrigerator	50.2	37.9	51.7*	49.0	50.8	53.7	49.3	48.4	53.7	49.7
Book	17.0	10.2	26.8*	16.3	19.4	18.6	16.3	16.2	15.3	16.5
Clock	59.7	38.0	64.6*	59.1	60.6	59.7	60.3	57.1	63.7	57.8
Vase	47.8	41.6	54.6*	47.7	47.2	47.7	49.3	49.2	49.5	51.0
Scissors	16.1	11.4	21.1*	19.2	19.4	21.9	17.2	12.5	15.6	18.3
Teddy bear	42.7	39.7	48.8*	39.9	41.3	45.5	43.6	42.6	47.4	43.2
Hair drier	32.7	12.7	16.4*	31.8	36.4	36.4	25.5	28.2	36.4	29.1
Toothbrush	15.8	14.6	19.5*	17.4	19.5	18.6	19.8	16.8	22.5	20.9

3 Additional COCO (VOC) \rightarrow COCO (non-VOC) Qualitative Results

Figure 3 depicts the qualitative results of baseline Mask R-CNN [5], LDET [16], GGN [17], UDOS [10], and our SOS on various test images in the cross-category COCO (VOC) \rightarrow COCO (non-VOC) setting. The results clearly show that SOS detects more objects across a variety of scenes, including cluttered indoor and outdoor scenes. Moreover, the detected objects cover several classes, sizes, and other object properties (e.g., elongation).

4 Pseudo Annotation Quality

This section presents detailed size- and class-specific results of the pseudo annotations from GGN [17] and SOS against the non-VOC objects of the COCO training set as described in Sec. 5.4 in the main paper.

Size-specific Results of Pseudo Annotation Quality Table 3 presents the size-specific recall results of the GGN₃ [17], SOS₃, and SOS₁₀ pseudo annotations against the non-VOC original annotations of the COCO training set. The size-specific results assign each annotated object to the class S (small object, area < 32²), M (medium object, 32² ≤ area < 96²), or L (large object, 96² ≤ area), as defined by [12]. The results show that SOS₃ outperforms GGN₃ in all size categories. Moreover, all object sizes profit from enlarging the number of pseudo annotations from 3 to 10. Hence, SOS₁₀ outperforms GGN₃ and SOS₃ across all object sizes.

Table 3: Size-specific recall results of the pseudo annotations GGN₃, SOS₃, and SOS₁₀ against the original non-VOC object annotations of the COCO training set.

Annotations	Rec ^S	Rec ^M	Rec ^L
GGN ₃	0.4	6.2	21.5
SOS ₃	4.4	15.6	34.4
SOS ₁₀	9.2	25.1	40.9

Class-specific Results of Pseudo Annotation Quality The class-specific recall results of the GGN₃ [17], SOS₃, and SOS₁₀ pseudo annotations against the original non-VOC object annotations on the COCO training set are given in Tab. 4. The results show that SOS₃ outperforms GGN₃ on most object classes, while SOS₁₀ outperforms SOS₃ on almost every class. In total, SOS₁₀ yields better results than GGN₃ on every object class.

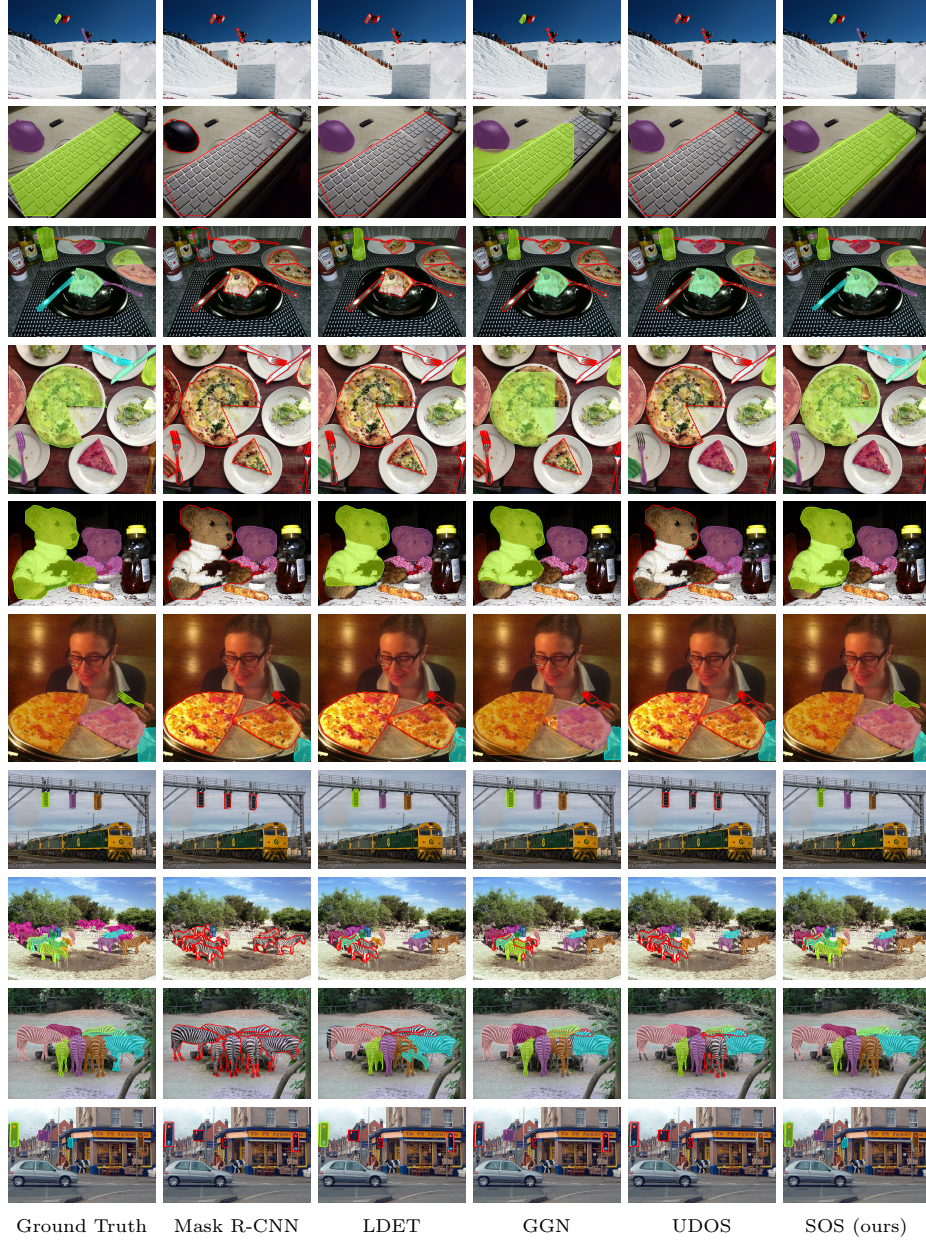


Fig. 3: Qualitative results of OWIS methods and baseline Mask R-CNN in the cross-category COCO (VOC) \rightarrow COCO (non-VOC) setting. Filled masks denote detected objects, while red frames indicate missed objects.

The characteristics between GGN_3 and SOS_3 differ. SOS_3 produces a high recall on animal-related classes, similar to the results in Tab. 2. GGN_3 has a notably low recall for these classes. A similar behavior is visible for **Sports ball**, **Broccoli**, and **Clock**, among others. The opposite effect, strong recall by GGN_3 and low recall for SOS_3 , is only visible for the class **Parking meter**. Comparing SOS_3 and SOS_{10} reveals similar per-class results on different levels of recall, with few exceptions (e.g., **Bear** and **Frisbee**).

Overall, SOS_3 and SOS_{10} recall many objects from animal classes, several sports-related classes, and most food classes. On the contrary, elongated objects like **Skis**, **Fork**, and **Knife** lead to low recalls, similar to the results in Tab. 2. Moreover, objects that are usually coupled with other objects like **Handbag**, **Backpack**, or **Bed** that typically appear with a human wearing it (**Handbag** and **Backpack**) or an item being located on it (**Bed**) lead to low recalls for both SOS_3 and SOS_{10} . We attribute this to the ambiguity of point prompts in these cases. However, more research is necessary to define clear patterns of simple and difficult object properties or alignments.

5 Additional Ablation Studies

Finally, we present several ablation studies showing the influence of the parameters in SOS, the use of multiple segments per prompt, the effect of the random sampling from the object priors, and the use of both pseudo annotations and original annotations from known classes. All experiments follow the setup described in Sec. 4.2 in the main paper.

Influence of Parameter S First, we analyze the influence of the parameter S , the number of sampled coordinates from the object priors. The results of SOS utilizing 20, 50, and 100 samples, presented in Tab. 5, show that initially sampling more coordinates leading to more prompts is beneficial, however, there is no difference between $S = 50$ and $S = 100$. Hence, more sampled coordinates do not lead to more or better pseudo annotations, as the number of pseudo annotations is limited by parameter N and other filtering steps (see Sec. 3.3 in the main paper). Overall, SOS is robust to the exact choice of S .

Influence of Parameter N We also investigate the influence of N , the size of the pruning region inside an object prior around an extracted coordinate during sampling. The results of SOS with different values for N in Tab. 6 indicate that the pruning of substantial areas is beneficial since $N = 20$ outperforms $N = 5$. Pruning larger areas ($N = 30$) does not lead to better results, and with even larger values the results are expected to decrease again. Overall, the pruning is important for strong results of SOS, however, SOS is robust to the exact choice of the value for N .

Table 4: Class-specific recall results of the pseudo annotations GGN₃, SOS₃, and SOS₁₀ against the original non-VOC object annotations of the COCO training set.

Class	GGN ₃	SOS ₃	SOS ₁₀	Class	GGN ₃	SOS ₃	SOS ₁₀
Truck	22.9	15.3	23.2	Bowl	8.8	10.4	18.2
Traffic light	2.9	6.7	12.6	Banana	3.4	7.3	12.2
Fire hydrant	27.4	40.5	50.1	Apple	2.4	11.6	12.1
Stop sign	17.2	39.2	41.4	Sandwich	12.0	23.9	36.0
Parking meter	21.2	15.6	24.1	Orange	2.5	9.8	11.5
Bench	4.7	7.0	12.0	Broccoli	1.5	16.5	29.0
Elephant	17.8	44.8	48.6	Carrot	1.0	4.4	7.3
Bear	8.0	79.5	69.5	Hot dog	8.4	18.7	35.8
Zebra	5.0	47.8	57.7	Pizza	14.4	39.8	43.8
Giraffe	10.0	62.7	68.6	Donut	4.7	22.4	27.6
Backpack	3.7	3.9	9.0	Cake	10.3	20.5	32.6
Umbrella	14.2	14.7	19.4	Bed	13.5	9.0	13.7
Handbag	2.5	4.2	9.1	Toilet	11.5	24.0	40.1
Tie	1.6	12.3	17.3	Laptop	25.3	25.5	35.4
Suitcase	12.0	14.3	20.2	Mouse	3.6	16.6	20.3
Frisbee	14.1	48.4	46.6	Remote	3.1	10.3	19.7
Skis	0.3	1.7	5.6	Keyboard	3.5	15.6	25.1
Snowboard	7.9	15.3	23.4	Cell phone	7.0	16.8	24.6
Sports ball	2.5	31.5	42.0	Microwave	12.2	19.0	25.8
Kite	9.8	14.9	27.0	Oven	8.8	8.8	19.5
Baseball bat	1.9	14.9	40.5	Toaster	15.1	20.0	24.9
Baseball glove	2.7	7.9	21.3	Sink	5.0	14.5	21.3
Skateboard	5.0	12.6	27.4	Refrigerator	12.4	13.9	19.0
Surfboard	13.2	27.7	41.1	Book	0.9	1.9	3.8
Tennis racket	10.5	21.3	44.4	Clock	3.3	33.0	38.0
Wine glass	4.5	5.6	10.8	Vase	10.6	19.1	23.5
Cup	7.7	8.2	13.2	Scissors	1.6	6.8	19.7
Fork	1.0	2.6	7.5	Teddy bear	13.9	20.1	28.3
Knife	1.5	3.2	6.5	Hair drier	11.6	19.7	33.3
Spoon	1.3	3.6	7.5	Toothbrush	2.1	12.6	24.5

Table 5: SOS results with various values for S in the COCO (VOC) \rightarrow COCO (non-VOC) setting.

S	AP	AR ₁₀₀	F ₁
20	8.6	37.9	14.0
50	8.9	38.1	14.4
100	8.9	38.1	14.4

Table 6: SOS results with various values for N in the COCO (VOC) \rightarrow COCO (non-VOC) setting.

N	AP	AR ₁₀₀	F ₁
5	8.1	37.9	13.3
10	8.6	38.1	14.0
20	8.9	38.1	14.4
30	8.9	38.0	14.4

Influence of Parameter τ_{conf} The parameter τ_{conf} controls the filtering of SAM segments based on SAM’s confidence score. The results of SOS with various values for τ_{conf} in Tab. 7 indicate that restricting the pseudo annotation generation to high-confidence SAM segments is beneficial ($\tau_{\text{conf}} = 70$ vs. $\tau_{\text{conf}} = 90$), with $\tau_{\text{conf}} = 90$ leading to the best results. This is similar to the findings in [11], where a threshold of 88 is used. Overall and similar to the previous parameters, SOS is robust to the exact choice of τ_{conf} .

Table 7: SOS results with various values for τ_{conf} in the COCO (VOC) \rightarrow COCO (non-VOC) setting.

τ_{conf}	AP	AR ₁₀₀	F ₁
70	8.6	38.0	14.0
80	8.8	38.2	14.3
90	8.9	38.1	14.4
95	8.9	36.9	14.3

Influence of Parameter τ_{NMS} To remove duplicate pseudo annotations and pseudo annotations strongly overlapping with original annotations, we apply NMS with τ_{NMS} as the IoU threshold. The results of SOS utilizing various values for τ_{NMS} and deactivating the NMS, visible in Tab. 8, indicate that the filtering is important for high-quality results of SOS. Similar to previous parameters, the exact choice of τ_{NMS} is not important, implying a robustness of SOS against the exact value of τ_{NMS} .

Number of Segments per Prompt in SAM To resolve ambiguous point prompts, we follow [11] and allow SAM to produce three segments per prompt. Table 9 shows the results of SOS, with SAM producing one or three segments per prompt. It is clearly visible that three segments per prompt improve the results. This is also in line with the image data that features several classes that regularly lead to ambiguous point prompts, including **Bench** and **Bed**.

Table 8: SOS results with various values for τ_{NMS} in the COCO (VOC) \rightarrow COCO (non-VOC) setting.

τ_{NMS}	AP	AR ₁₀₀	F ₁
off	8.0	36.5	13.1
70	8.6	38.1	14.0
90	8.8	38.1	14.3
95	8.9	38.1	14.4
98	8.9	37.8	14.4

Table 9: SOS results with 1 or 3 segments generated per prompt inside SAM in the COCO (VOC) \rightarrow COCO (non-VOC) setting.

Segments per prompt	AP	AR ₁₀₀	F ₁
1	5.7	33.8	9.8
3	8.9	38.1	14.4

Random Sampling from Object Prior Since coordinates for point prompts are randomly sampled from the object prior, we investigate the influence of this randomness on the overall results. To this end, we run the entire training pipeline of SOS including the random sampling from the object priors ten times and investigate the variability in AP, AR₁₀₀, and F₁. Overall, the standard deviation is very low with 0.09 for AP, 0.07 for AR₁₀₀, and 0.12 for F₁ on an interval of $[0, 100]$. For instance, the largest difference in AP between two runs is 0.2 (8.9 vs. 9.1). Therefore, the results of SOS are stable.

Pseudo Annotations Only Finally, we investigate the performance of SOS using only pseudo annotations, ignoring the original annotations of the known classes. The results in Tab. 10 clearly show that a mixture of original and pseudo annotations drastically improves the results of SOS. Hence, despite the high quality of the pseudo annotations, a foundation of original annotations is crucial for a strong performance of SOS.

Table 10: SOS results with the instance segmentation system trained on pseudo annotations only or a mixture of pseudo annotations and original COCO annotations of VOC classes.

Annotations	AP	AR ₁₀₀	F ₁
Pseudo	1.4	10.6	2.5
Pseudo + COCO (VOC)	8.9	38.1	14.4

References

1. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: International Conference on Computer Vision (2021) [3](#)
2. Dollár, P., Zitnick, C.L.: Structured forests for fast edge detection. In: International Conference on Computer Vision (2013) [2](#)
3. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. *International Journal of Computer Vision* **59** (2004) [2](#)
4. Frintrop, S., Werner, T., Martin Garcia, G.: Traditional saliency reloaded: A good old model in new shape. In: Conference on Computer Vision and Pattern Recognition (2015) [2](#)
5. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: International Conference on Computer Vision (2017) [9](#)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Conference on Computer Vision and Pattern Recognition (2016) [2](#)
7. Horbert, E., Martín García, G., Frintrop, S., Leibe, B.: Sequence-level object candidates based on saliency for generic object recognition on mobile systems. In: International Conference on Robotics and Automation (2015) [2](#)
8. Jiang, M., Huang, S., Duan, J., Zhao, Q.: SALICON: Saliency in context. In: Computer Vision and Pattern Recognition (2015) [2](#)
9. Judd, T., Ehinger, K., Durand, F., Torralba, A.: Learning to predict where humans look. In: 2009 IEEE 12th international Conference on Computer Vision (2009) [2](#)
10. Kalluri, T., Wang, W., Wang, H., Chandraker, M., Torresani, L., Tran, D.: Open-world instance segmentation: Top-down learning with bottom-up supervision. *arXiv preprint arXiv:2303.05503* (2023) [9](#)
11. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. In: International Conference on Computer Vision (2023) [1](#), [13](#)
12. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: European Conference on Computer Vision (2014) [6](#), [9](#)
13. Linardos, A., Kümmeler, M., Press, O., Bethge, M.: DeepGaze IIE: Calibrated prediction in and out-of-domain for state-of-the-art saliency modeling. In: International Conference on Computer Vision (2021) [2](#)
14. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention (2015) [3](#)
15. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. *International Journal of Computer Vision* **115** (2015) [2](#), [3](#)
16. Saito, K., Hu, P., Darrell, T., Saenko, K.: Learning to detect every thing in an open world. In: European Conference on Computer Vision (2022) [9](#)
17. Wang, W., Feiszli, M., Wang, H., Malik, J., Tran, D.: Open-world instance segmentation: Exploiting pseudo ground truth from learned pairwise affinity. In: Conference on Computer Vision and Pattern Recognition (2022) [9](#)
18. Wilms, C., Frintrop, S.: Edge adaptive seeding for superpixel segmentation. In: German Conference on Pattern Recognition (2017) [2](#)
19. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: Conference on Computer Vision and Pattern Recognition (2016) [2](#)