Appendix

Learning to Find Common Objects Across Few Image Collections

A. Co-Localization: COCO Dataset Creation and Faster-RCNN Training

COCO dataset has 80 classes in total. We take the same 17 unseen classes which is used in zero-shot object detection paper [Ref1] and keep remaining 63 classes for training. The training set is constructed using the images in COCO 2017 train set which contain at least one object from the seen classes. The COCO test set, is built by combining the unused images of the train set and images in COCO validation set which contain at least one object from the unseen classes. Similar to [Ref1], to avoid training the network to classify unseen objects as background, we remove objects from unseen classes from the training images using their ground-truth segmentation masks.

We use Tensorflow object-detection API for pre-training the Faster-RCNN feature extraction module [Ref4]. To speed up pre-training, training images are resized down to 336×336 pixels and ResNet-50 [Ref3] is used as the backbone feature extractor. All layer weights are initialized with variance scaling initialization [Ref2] and biases are set to zero initially. An additional linear layer which maps the 2048 dimensional output of second stage feature extractor to a $d = 640$ dimensional feature vector is added to the network. We did this to have the dimension of the feature space the same as few-shot common object recognition experiment. We pre-train the feature extractor on four GPUs with batch size of 12 for $600k$ iterations. The $d = 640$ dimensional features are used as input to all of the methods in our experiments.

B. Hyperparameter Tuning

In the few-shot common object recognition task, we use grid search on the validation set to tune the hyperparameters of all the methods. To ensure that the structured inference methods optimize the same objective function, we find η for the TRWS method and use the same value in AStar and greedy energy functions. For the few-shot common object recognition task value of η is shown in Table 5 for each setting.

In the Co-Localization experiments, the results of the best performing hyperparameters is reported for all the methods. $\eta = 0.5$ and $\eta = 0.7$ is used in COCO and ImageNet experiments respectively.

C. Structured Inference Methods Comparison

The numerical results which are used to generate Figure 2 of the paper are shown in Table 5. The success rate of the greedy method is on par with the other inference algorithms. From the optimization point of view it is also important to see the mean energy value for the top selection of each method. These results are shown in Table 6 and Table 7 for few-shot common object recognition and co-localization experiments respectively. While AStar and TRWS achieve lower energy values for this problems, the success rate of the methods are comparable. This suggests that finding an approximate solution for the minimization problem is sufficient for achieving high success rate.

	N	4			8			16		
	$\bar{B}$	0	10	20	0	10	20	0	10	20
B = 5	TRWS	$54.55 \pm 1.54(0.0)$	$63.78 \pm 1.49(0.5)$	$65.43 \pm 1.47(0.6)$	$64.55 \pm 1.05(0.0)$	$72.60 \pm 0.98(0.8)$	$73.80 \pm 0.96(1.2)$	$70.29 \pm 0.71(0.0)$	$78.71 \pm 0.63(1.6)$	$80.08 \pm 0.62(1.9)$
	ASTAR	$54.55 \pm 1.54(0.0)$	$63.82 \pm 1.49(0.5)$	$65.48 \pm 1.47(0.6)$	$64.48 \pm 1.05(0.0)$	$72.49 \pm 0.98(0.8)$	$73.99 \pm 0.96(1.2)$	$69.91 \pm 0.71(0.0)$	$78.49 \pm 0.64(1.6)$	$80.03 \pm 0.62(1.9)$
	Greedy(Ours)	$54.55 \pm 1.54(0.0)$	$63.83 \pm 1.49(0.5)$	$65.48 \pm 1.47(0.6)$	$64.48 \pm 1.05(0.0)$	$72.49 \pm 0.98(0.8)$	$73.99 \pm 0.96(1.2)$	$69.67 \pm 0.71(0.0)$	$78.60 \pm 0.64(1.6)$	$79.93 \pm 0.62(1.9)$
B = 10	TRWS	$29.40 \pm 1.41(0.0)$	$37.15 \pm 1.50(0.5)$	$38.50 \pm 1.51(0.7)$	$36.14 \pm 1.05(0.0)$	$42.61 \pm 1.08(0.9)$	$47.59 \pm 1.09(1.1)$	$41.45 \pm 0.76(0.0)$	$50.88 \pm 0.77(1.5)$	$53.71 \pm 0.77(2.3)$
	ASTAR	$29.20 \pm 1.41(0.0)$	$37.43 \pm 1.50(0.5)$	$38.50 \pm 1.51(0.7)$	$35.96 \pm 1.05(0.0)$	$42.83 \pm 1.08(0.9)$	$47.46 \pm 1.09(1.1)$	$41.41 \pm 0.76(0.0)$	$51.32 \pm 0.77(1.5)$	$53.57 \pm 0.77(2.3)$
	Greedy(Ours)	$29.20 \pm 1.41(0.0)$	$37.42 \pm 1.50(0.5)$	$38.50 \pm 1.51(0.7)$	$35.98 \pm 1.05(0.0)$	$42.85 \pm 1.08(0.9)$	$47.63 \pm 1.09(1.1)$	$41.54 \pm 0.76(0.0)$	$51.70 \pm 0.77(1.5)$	$53.63 \pm 0.77(2.3)$

Table 5. *Success rate of different energy minimization algorithms on miniImageNet. These numbers were used to generate Figure 2 in the paper. Value of the parameter η is shown in the parenthesis for each experiment. See section 5.2 and Table 1 for the detailed problem setup.*

D. Sharing Parameters of Unary and Pairwise Relation Modules

As it is discussed in section 4, both unary and pairwise potential functions use the relation module with an identical architecture. However, since the input class distribution is different for these functions, we choose not to share their parameters. We conduct an experiment to see the effect of parameter sharing in few-shot common object recognition task with $B = 5$, $N = 8$, and $\bar{B} = 10$. As Table 1 shows, the success rate for this setting is $72.49 \pm 0.98\%$ without parameter sharing. However, when the unary and pairwise are trained with shared relation module parameters, the performance degrades to $69.35 \pm 1.01\%$.

	N	4			8			16		
	$\bar{B}$	0	10	20	0	10	20	0	10	20
B = 5	TRWS	2.929179	−4.416873	−4.842334	18.300657	−4.425953	−12.602217	86.034355	−6.873013	−10.020649
	ASTAR	2.908970	−4.429455	−4.851543	18.192284	−4.529052	−12.666497	85.560267	−7.277377	−10.398633
	Greedy	2.908970	−4.429455	−4.851543	18.192282	−4.529052	−12.666499	86.692482	−6.909996	−10.002609
B = 10	TRWS	0.515563	−6.576048	−8.300273	8.749933	−15.959289	−17.238385	53.324193	−28.602048	−59.609459
	ASTAR	0.502832	−6.597286	−8.315386	8.675015	−16.079914	−17.404502	52.819455	−29.388606	−60.499036
	Greedy	0.502832	−6.597286	−8.315387	8.707342	−16.048676	−17.384832	57.168652	−25.869081	−57.885948

Table 6. *Expected energy for different inference methods. Lower energy is better.*

Method	COCO	ImageNet
TRWS	−28.485636	−28.630786
AStar	−28.487422	−28.631678
Greedy	−27.246355	−25.496649

Table 7. *Mean energy on COCO and ImageNet with* 8 *positive and* 8 *negative images. Lower energy is better.*

E. More Qualitative Results

Qualitative results on ImageNet dataset are illustrated in Figure 5. Figure 6 shows the complete qualitative results presented in the paper with the negative images on COCO dataset.

References

[Ref1] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. Zero-shot object detection. In *ECCV*, pages 384–400, 2018. http://ankan.umiacs.io/zsd.html.

[Ref2] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, pages 249–256, 2010.

[Ref3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[Ref4] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *CVPR*, pages 7310–7311, 2017.

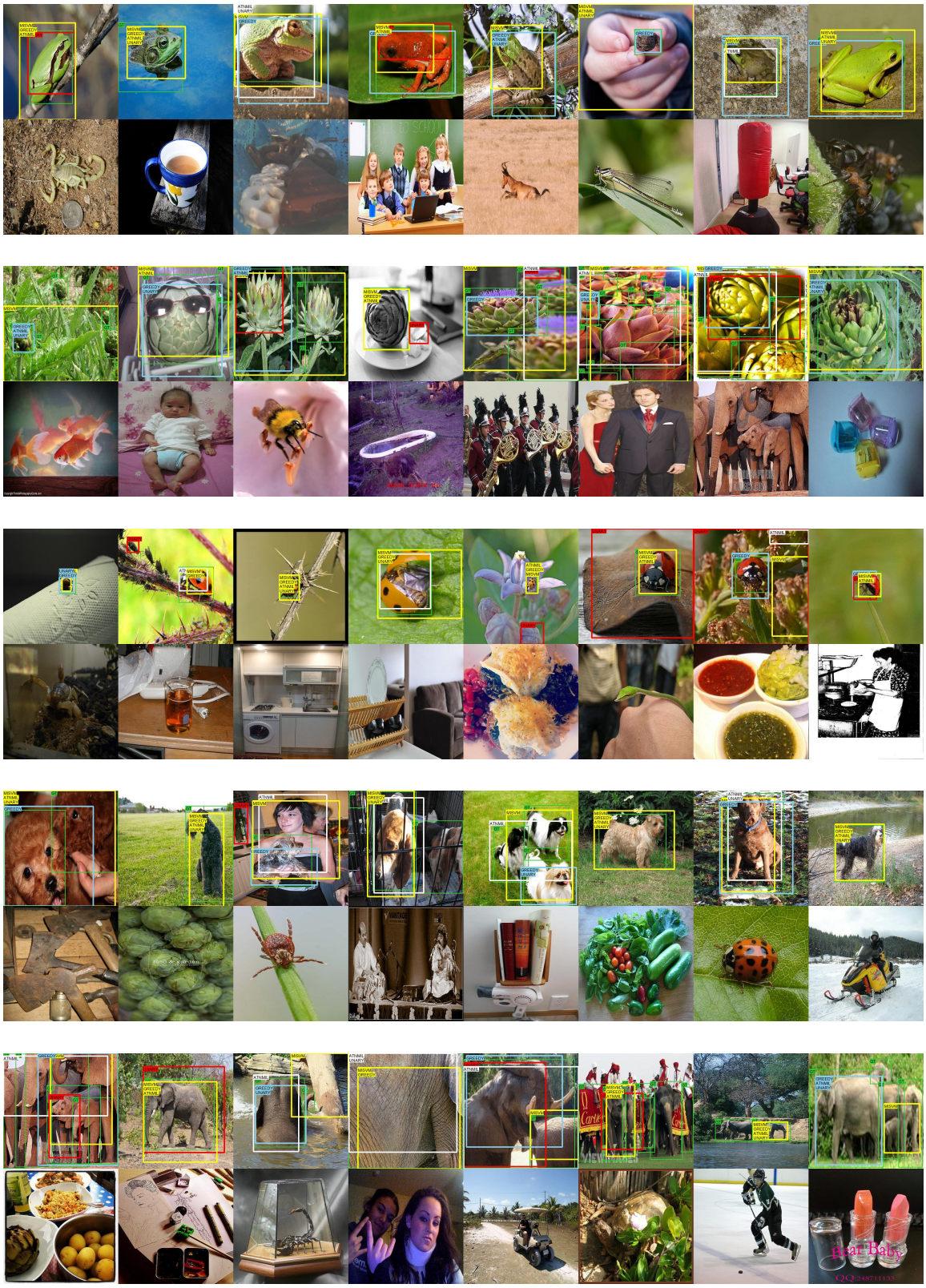

Figure 5. *Qualitative results on ImageNet dataset. In each problem, the first row and the second row show positive and negative images respectively. While different methods work as good in easier images with one object, the greedy method performs better in harder examples with multiple objects in each image. Selected regions are tagged with method names. Ground-truth target bounding box is shown in green with tag "GT".*

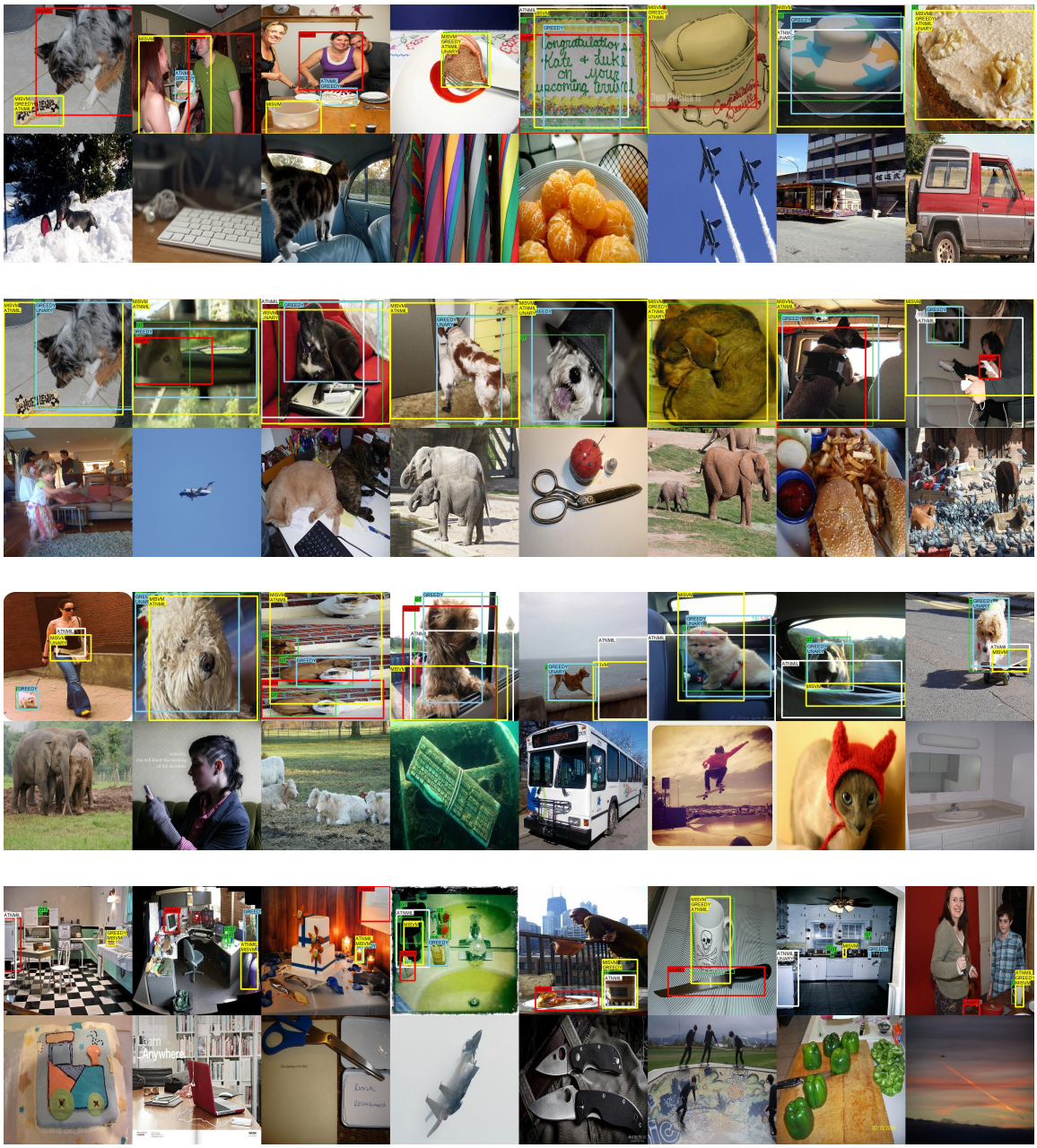

Figure 6. *Qualitative results on COCO. Complete version of the results shown in Figure 3 of the paper with negative images. In the first problem, class "Person" does not appear in the negative images. This could explain why "Unary Only" method detects people in the first problem.*