Where is my Wallet? Modeling Object Proposal Sets for Egocentric Visual Query Localization (*Supplementary*)

Mengmeng Xu[1,2*] Yanghao Li[1] Cheng-Yang Fu[1] Bernard Ghanem[2] Tao Xiang[1]
Juan-Manuel Pérez-Rúa[1]

[1] Meta AI [2] KAUST, Saudi Arabia

{mengmeng.xu,bernard.ghanem}@kaust.edu.sa, {lyttonhao,chengyangfu,txiang,jmpr}@meta.com

A. Metrics and Implementation details

A.1. Metrics selection

In each task, we followed the metrics introduced in Ego4D [10].

Query Object Detection. We consider average precision (AP) as the main metric. It is the precision averaged over different recalls of the multiple predictions on the image. We also compare AP_{50}/AP_{75} to study the predicted bounding boxes on loose and tide criteria and the top-10 recall to study the missing detection problem.

VQ2D and VQ3D. Most of the metrics focus on the closeness of the prediction to the ground truth. tAP_{25} and $stAP_{25}$ in VQ2D evaluate how closely in the temporal and spatio-temporal extent the predicted response track matches the ground truth, respectively, where the intersection over the union threshold is 0.25 by default. $L2$ and $angle$ in VQ3D measure the difference between the predicted and ground-truth displacement vectors in the real-world coordinates. For a fair reference, we also report success ($Succ$) and recovery percentage ($rec\%$) to study how many predictions overlap the ground-truth, and how many ground truths are discovered by predictions.

A.2. Implementation details

Training details. Following the optimized VQ2D baseline [20], we implement our algorithms on Detectron2 [18]. The visual query detection is conducted on 4 8-V100 GPU nodes in a distributed machine learning cluster. Each experiment trains the detector for 125k iterations with an initial learning rate of 0.02, which decays at 50k and 100k iterations by 0.1. Our batch size is 64.

Frame Sampling. The training frames are sampled from video when a response track annotation is available. Our negative unlabeled frame sampling (N-UFS) is based on a *negative video* starting at the end of the response track until the query frame. We sample as many frames from this negative video as the number of positive frames. When applying positive unlabeled frame sampling (P-UFS), we run a COCO-pretrained Faster-RCNN [15] in on all training videos with FPS=1, and track [1] the predicted object with a confidence threshold of 0.5 on both forward and backward directions. We remove outliers of this object based on a pre-defined range of area and aspect ratio. In the optimal setting, we totally sample 1.7 million extra query-frame pairs to train the detector.

To achieve the visual query localization tasks, we apply our trained detector in the respective pipelines [10]. In VQ2D, we run a *Kys* [1] tracker from the detection peak to predict the response track. In VQ3D, we leverage our improved query detector for frames where camera pose information is available. Note that we do not further modify these stages to ensure a fair comparison.

B. Few-shot Object Detection

B.1. Experiment setup

Dataset Our few-shot object detection experiments are on the MS-COCO dataset [12]. The novel/base splits follow the setting of Kang *et al*. [11]. From the 80 object categories, we use the 20 classes that overlap with the PASCAL VOC [6] dataset as novel classes and the remaining 60 as base classes. Similarly, 5000 images from the validation set are used for evaluation, while the rest images in training and validation sets are used for training.

Training details Our few-shot object detection model follows the released Faster-RCNN design and training recipe in [13]. Its Hierarchical Attention Module encodes spatial information in the object proposals, then we vectorize the enriched proposal representation and feed them to our CocoFormer. We do base-training for 1-shot, 3-shot, and 5-shot without fine-tuning. Each base training is independent

*Work done during an internship at Meta AI.

Method	novel ft.	1-shot			3-shot			5-shot		
		nAP	AP50	AP75	nAP	AP50	AP75	nAP	AP50	AP75
TFA [16]	True	3.4	5.8	3.8	6.6	12.1	6.5	8.3	15.3	8.0
CoRPN [23]	True	4.1	7.2	4.4	-	-	-	-	-	-
Meta-DETR [21]	True	7.5	12.5	7.7	-	-	-	-	-	-
FADI [4]	True	5.7	10.4	6.0	-	-	-	-	-	-
Xiao *et al*. [19]	True	3.2	8.9	1.4	6.7	18.6	2.9	8.1	20.1	4.4
MPSR [17] †	True	2.3	4.1	2.3	5.2	9.5	5.1	6.7	12.6	6.4
Fan *et al*. [7] †	True	4.2	9.1	3.0	6.6	15.9	4.9	8.0	18.5	6.3
Zhang *et al*. [22]	True	4.4	7.5	4.9	7.2	13.3	7.4	-	-	-
QA-FewDet [8]	True	4.9	10.3	4.4	8.4	18.0	7.3	9.7	20.3	8.6
DeFRCN [14]	True	9.3	-	-	14.8	-	-	16.1	-	-
Fan *et al*. [7] †	False	4.0	8.5	3.5	5.9	12.5	5.0	6.9	14.3	6.0
Meta Faster-RCNN [9]	False	5.0	10.5	4.5	-	-	-	-	-	-
QA-FewDet [8]	False	5.1	10.5	4.5	8.6	17.7	7.5	9.5	19.3	8.5
FS-DETR [2]	False	7.0	13.6	7.5	9.8	18.5	9.8	10.7	20.5	10.8
DAnA [5]	False	11.9	**25.6**	10.4	14.0	**28.9**	12.3	14.4	**30.4**	13.0
hANMCL [13]	False	12.9	25.0	12.1	14.4	28.0	13.3	14.5	27.9	13.3
ours	False	**13.3**	**25.6**	**12.6**	**14.7**	<u>28.8</u>	**13.4**	**14.8**	<u>28.9</u>	**13.6**

Table 1. **Assessing model performance in Few-Shot Detection.** We show 1-shot, 3-shot, and 5-shot settings on the MS COCO dataset. nAP means the novel categories average precision. † means reproduced result by QA-FewDet [8].

and done on a single Tesla V100 machine for 12 epochs. The learning rate starts at 0.001 and increases by 0.1 times per 1000 steps. We used stochastic gradient descent to optimize the model with a momentum of 0.9 and a weight decay of 0.0001.

B.2. Full comparison with SOTA

Tab. 1 assesses model performance in Few-Shot Detection. 1-shot, 3-shot, and 5-shot settings are respectively applied on the MS COCO [3] dataset. We divide the methods into two groups. Methods in the first block require fine-tuning on the novel classes. Their models got further optimized on the support set, so the performance especially on higher shots is relatively higher. Our method belongs to the second group, where the model is directly evaluated after the base train. Comparing novel categories' average precision (nAP), our method can consistently improve the baseline [13], outperform state-of-the-art, and is competitive with the fine-tuning methods in the first block. Notably, our method achieves 13.3 nAP in 1-shot object detection, which shares a more similar problem setting as visual query object detection.

B.3. Visual query *vs*. few-shot detection

We would like to emphasize that although visual query and few-shot detection share similar configurations, but they are identical to each other.

First, visual query detection is based on *an instance-level dataset*, while few-shot detection is on the class level. This new task requires the system to localize exactly the same object registered by its visual crop. Therefore, more than one instance from the same classes can con-exist in the query video, but the metrics will penalize a wrong instance. For example, there are four bins in the blue bins video in the qualitative result, but we have to find the blue bin along the corner of the wall.

Second, the *episodic training strategy*, which is widely used in few-shot detection, is not the optimal solution in visual query detection. This is because we have only one visual crop of the query object and thousands of novel instances. Applying an episodic training strategy may slightly improve the model performance, but it will greatly increase the training time.

C. Supplementary experiment

Siam-RCNN *vs*. CocoFormer Our CocoFormer and P-UFS improve the framework in different aspects. CocoFormer is a novel transformer-based module that allows for object-proposal set context to be considered while incorporating query information, while the main motivation of positive unlabeled frame sampling (P-UFS) is to reduce the training domain gap between the overall possible object instance and the existing annotations.

In Tab. 2, we further validate this simple augmentation method on the baseline detector and our proposed CocoFormer. The comparisons in each block show our augmentation strategy P-UFS effectively extends the training set, bringing consistent performance gain in both settings. If we compare CocoFormer with Siam-RCNN with or without P-USF, we can find the AP score is improved, yet AR@10 becomes lower. This means CocoFormer is more strict about predicting positives, and the precision is greatly increased.

backbone	P-UFS	AP	AP_{50}	AP_{75}	$AR@10$
Siam-RCNN	✗	27.55	50.43	26.16	47.3
Siam-RCNN	✓	**28.74**	**52.25**	**27.35**	**50.1**
CocoFormer	✗	30.35	57.87	26.76	45.9
CocoFormer	✓	**31.26**	**57.96**	**28.88**	**47.1**

Table 2. **Our augmentation strategy effectively extends the training set.** We validate the augmentation on Siam-RCNN and CocoFormer, and it shows consistent performance gain in both settings.

D. Further discussion

Due to space limitations, we left some further discussion and insight in this section.

Performance mismatch between VQD and VQL. Most of the experiment tables show the model performances are not consistent when evaluated on VQ detection and VQ localization, which means a top-performing detection model can be sub-optimal for temporal localization. This is mainly because VQD is only evaluated on *individually annotated frames* of the dataset, while VQL is evaluated on the entire video. Positive frames are on average only *2% of all the frames* in the video. Also, VQD is heavily biased because annotated frames always contain the query object, while a randomly sampled video frame doesn't have this property. Thus, VQL is much more challenging than VQD. In this paper, we presented both VQD and VQL metrics to *prove* that a better detector doesn't always lead to a better localizer. This is precisely the main motivation for our work: to reduce training bias between VQD and VQL by introducing various sampling methods.

Concatenation and Conditional Projection in our proposed CocoFormer are both *possible settings*. Although Concatenation works better on VQD, Conditional Projection is generally better in VQL, showing that the tracking process in the localization model is more sensitive to AP75. It means a precise bounding box is necessary to produce a correct response track.

N-UFS and BPS for VQL follow our main idea to sample data close to the VQL *real distribution*. From the detection perspective, these simple methods are nontrivial or even counterintuitive, as clean images with the query object are preferred. However, the real-world data in VQL is noisy and long-tailed, so we have to use N-UFS and BPS to create necessary samples in this domain, and we find they are quite effective. Both methods are harmful when evaluated on VQD but helpful and essential in VQL to suppress false positives, as shown by similarity scores on background frames in Fig. 5.

References

[1] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Know your surroundings: Exploiting scene information for object tracking. In *European Conference on Computer Vision*, pages 205–221. Springer, 2020.

[2] Adrian Bulat, Ricardo Guerrero, Brais Martinez, and Georgios Tzimiropoulos. Fs-detr: Few-shot detection transformer with prompting and without re-training. *arXiv preprint arXiv:2210.04845*, 2022.

[3] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *CVPR*, 2018.

[4] Yuhang Cao, Jiaqi Wang, Ying Jin, Tong Wu, Kai Chen, Ziwei Liu, and Dahua Lin. Few-shot object detection via association and discrimination. *Advances in Neural Information Processing Systems*, 34:16570–16581, 2021.

[5] Tung-I Chen, Yueh-Cheng Liu, Hung-Ting Su, Yu-Cheng Chang, Yu-Hsiang Lin, Jia-Fong Yeh, Wen-Chin Chen, and Winston Hsu. Dual-awareness attention for few-shot object detection. *IEEE Transactions on Multimedia*, 2021.

[6] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[7] Qi Fan, Wei Zhuo, Chi-Keung Tang, and Yu-Wing Tai. Few-shot object detection with attention-rpn and multi-relation detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4013–4022, 2020.

[8] Guangxing Han, Yicheng He, Shiyuan Huang, Jiawei Ma, and Shih-Fu Chang. Query adaptive few-shot object detection with heterogeneous graph convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3263–3272, 2021.

[9] Guangxing Han, Shiyuan Huang, Jiawei Ma, Yicheng He, and Shih-Fu Chang. Meta faster r-cnn: Towards accurate few-shot object detection with attentive feature alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 780–789, 2022.

[10] Ego4D Consortium 2020. Ego4d: Around the World in 3,000 Hours of Egocentric Video. In *CVPR*, 2022.

[11] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. In *ICCV*, 2019.

[12] 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 *ECCV*. Springer, 2014.

[13] Dongwoo Park and Jongmin Lee. Hierarchical attention network for few-shot object detection via meta-contrastive learning. *arXiv preprint arXiv:2208.07039*, 2022.

[14] Limeng Qiao, Yuxuan Zhao, Zhiyuan Li, Xi Qiu, Jianan Wu, and Chi Zhang. Defrcn: Decoupled faster r-cnn for few-shot object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8681–8690, 2021.

[15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

[16] Xin Wang, Thomas E. Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu. Frustratingly simple few-shot object detection. *International Conference on Machine Learning (ICML)*, July 2020.

[17] Jiaxi Wu, Songtao Liu, Di Huang, and Yunhong Wang. Multi-scale positive sample refinement for few-shot object detection. In *European conference on computer vision*, pages 456–472. Springer, 2020.

[18] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

[19] Yang Xiao and Renaud Marlet. Few-shot object detection and viewpoint estimation for objects in the wild. In *ECCV*, 2020.

[20] Mengmeng Xu, Cheng-Yang Fu, Yanghao Li, Bernard Ghanem, Juan-Manuel Perez-Rua, and Tao Xiang. Negative frames matter in egocentric visual query 2d localization. *arXiv preprint arXiv:2208.01949*, 2022.

[21] Gongjie Zhang, Zhipeng Luo, Kaiwen Cui, and Shijian Lu. Meta-detr: Few-shot object detection via unified image-level meta-learning. *arXiv preprint arXiv:2103.11731*, 2(6), 2021.

[22] Weilin Zhang and Yu-Xiong Wang. Hallucination improves few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13008–13017, 2021.

[23] Weilin Zhang, Yu-Xiong Wang, and David A Forsyth. Cooperating rpn's improve few-shot object detection. *arXiv preprint arXiv:2011.10142*, 2020.