OW-DETR: Open-world Detection Transformer Supplementary Material

Akshita Gupta[*1] Sanath Narayan[*1] K J Joseph[2,4]
Salman Khan[4,3] Fahad Shahbaz Khan[4,5] Mubarak Shah[6]
[1]Inception Institute of Artificial Intelligence [2]IIT Hyderabad [3]Australian National University
[4]Mohamed Bin Zayed University of Artificial Intelligence [5]CVL, Linköping University [6]University of Central Florida

In this supplementary, we present additional quantitative and qualitative results along with societal impact and limitations of our proposed open-world object detection framework, OW-DETR. The quantitative results are discussed in Sec. A1 followed by qualitative analysis in Sec. A2. The limitations and societal impact are covered in Sec. A3.

A1. Additional Quantitative Results

A1.1. Evaluation using WI and A-OSE Metrics

Tab. A1 shows a state-of-the-art comparison the for open-world object detection (OWOD) setting on the MS-COCO dataset in terms of wilderness impact (WI) and absolute open-set error (A-OSE). The WI metric [3,4] measures the model's confusion in predicting an unknown instance as known class, given by

$$\text{WI} = \frac{P_{\mathcal{K}}}{P_{\mathcal{K}\cup\mathcal{U}}} - 1,$$

where $P_{\mathcal{K}}$ is the model precision for known classes when evaluated on known class instances alone and $P_{\mathcal{K}\cup\mathcal{U}}$ denotes the same when evaluated with unknown class instances included. On the other hand, the A-OSE metric measures the total number of unknown instances detected as one of the known classes. Both these two (WI and A-OSE) indicate the degree of confusion in predicting the known classes in the presence of unknown instances. Furthermore, we also show the comparison in terms of U-Recall for ease of comparison. It is worth mentioning that U-Recall directly relates to the unknown class and measures the model's ability to retrieve the unknown instances.

The standard object detectors (Faster R-CNN and DDETR) in the top part of Tab. A1 are inherently not suited for the OWOD setting since they cannot detect any unknown object. Thereby, for these frameworks, only WI and A-OSE can be computed but not U-Recall. Since the energy-based unknown identifier (EBUI) in the recently introduced ORE [4] is learned using a held-out validation set with weak unknown supervision, for a fair comparison in the OWOD setting, we compare with ORE not employing EBUI. We observe that the standard single-stage DDETR wrongly predicts unknown instances as known classes and performs poorly in terms of A-OSE, compared to the two-stage Faster R-CNN. However, by adapting DDETR to OWOD setting through the proposed introduction of attention-driven pseudo-labeling, novelty classification and objectness branch, our OW-DETR obtains improved performance in terms of all three metrics across tasks over the Faster R-CNN based ORE. These results emphasize the importance of the proposed contributions towards a more accurate OWOD.

A1.2. Proposed MS-COCO Split for Open-world

Open-world object detection (OWOD) is a challenging setting due to its open-taxonomy nature. However, the dataset split proposed in ORE [4] for OWOD allows data leakage across tasks since different classes from a super-categories are introduced in different tasks, *e.g.*, most classes from *vehicle* and *animal* super-categories are introduced in Task 1, while related classes like *truck*, *elephant*, *bear*, *zebra* and *giraffe* are introduced in Task 2. Here, we conduct an experiment by constructing a stricter MS-COCO split, where classes are added across super-categories, as shown in Tab. A3. Such a split mitigates possible data leakage across tasks since all the classes of a super-category are introduced at a time in a task and not spread across tasks. Thereby, the proposed split is more challenging for OWOD setting. The new split is divided by super-categories with nearly 20 classes in each task: *Animals*, *Person*, *Vehicles* in Task 1; *Appliances*, *Accessories*, *Outdoor*, *Furniture* in Task 2; *Food*, *Sport* in Task 3; *Electronic*, *Indoor*, *Kitchen* in Task 4. Here, Tab. A2 shows that our OW-DETR achieves improved performance even on this stricter OWOD split, compared to the recently introduced ORE. We note that the proposed bottom-up attention driven pseudo-labeling scheme aids our OWOD framework to better generalize to unseen super-categories.

[*]Equal contribution

Table A1. **State-of-the-art comparison for open-world object detection (OWOD) on MS-COCO.** The comparison is shown in terms of wilderness impact (WI), absolute open set error (A-OSE) and unknown class recall (U-Recall). The unknown recall (U-Recall) metric quantifies a model's ability to retrieve the unknown object instances. The standard object detectors (Faster R-CNN and DDETR) in the top part of table *are inherently not suited for the OWOD setting since they cannot detect any unknown object* and thereby U-Recall cannot be computed for them. For a fair comparison in the OWOD setting, we compare with the recently introduced ORE [4] not employing EBUI. Our OW-DETR achieves improved WI and A-OSE over ORE across tasks, thereby indicating lesser confusion in detecting unknown instances as known classes. Furthermore, our OW-DETR achieves improved U-Recall over ORE across tasks, indicating our model's ability to better detect the unknown instances. Note that WI, A-OSE and U-Recall cannot be computed in Task 4 (and hence not shown) since all 80 classes are known. See Sec. A1.1 for additional details.

Task IDs (→)	**Task 1**			**Task 2**			**Task 3**		
	U-Recall (↑)	WI (↓)	A-OSE (↓)	U-Recall (↑)	WI (↓)	A-OSE (↓)	U-Recall (↑)	WI (↓)	A-OSE (↓)
Faster-RCNN [6]	-	0.0699	13396	-	0.0371	12291	-	0.0213	9174
Faster-RCNN + Finetuning	Not applicable in Task 1			-	0.0375	12497	-	0.0279	9622
DDETR [7]	-	0.0608	33270	-	0.0368	18115	-	0.0197	9392
DDETR + Finetuning	Not applicable in Task 1			-	0.0337	17834	-	0.0195	10095
ORE − EBUI [4]	4.9	0.0621	10459	2.9	0.0282	10445	3.9	0.0211	7990
Ours: OW-DETR	**7.5**	**0.0571**	**10240**	**6.2**	**0.0278**	**8441**	**5.7**	**0.0156**	**6803**

Table A2. **State-of-the-art comparison for OWOD on the proposed MS-COCO split.** The comparison is shown in terms of known class mAP and unknown class recall (U-Recall). For a fair comparison in the OWOD setting, we compare with the recently introduced ORE [4] not employing EBUI. The proposed split mitigates data leakage across tasks and is more challenging than the original OWOD split of [4]. Even on this harder data split, our OW-DETR achieves improved U-Recall over ORE across tasks, indicating our model's ability to better detect the unknown instances. Furthermore, our OW-DETR also achieves significant gains in mAP for the known classes across the four tasks. Note that since all 80 classes are known in Task 4, U-Recall is not computed. See Sec. A1.2 for more details.

Task IDs (→)	**Task 1**		**Task 2**				**Task 3**				**Task 4**			
	U-Recall (↑)	mAP (↑) Current known	U-Recall (↑)	mAP (↑) Previously known	mAP (↑) Current known	mAP (↑) Both	U-Recall (↑)	mAP (↑) Previously known	mAP (↑) Current known	mAP (↑) Both	mAP (↑) Previously known	mAP (↑) Current known	mAP (↑) Both
ORE − EBUI [4]	1.5	61.4	3.9	56.5	26.1	40.6	3.6	38.7	23.7	33.7	33.6	26.3	31.8
Ours: OW-DETR	**5.7**	**71.5**	**6.2**	**62.8**	**27.5**	**43.8**	**6.9**	**45.2**	**24.9**	**38.5**	**38.2**	**28.1**	**33.1**

Table A3. **Task composition in the proposed MS-COCO split for Open-world evaluation protocol**. The semantics of each task and the number of images and instances (objects) across splits are shown. The proposed task split mitigates the data leakage across tasks that was present in the split of ORE [4]. *E.g.*, all *vehicles* including *truck*, which was in Task 2 earlier are now in Task 1. Similarly all *animals* are now in Task 1, while other Pascal VOC classes like *sofa*, *bottle*, *etc*. are moved out of Task 1.

	Task 1	**Task 2**	**Task 3**	**Task 4**
Semantic split	Animals, Person, Vehicles	Appliances, Accessories, Outdoor, Furniture	Sports, Food	Electronic, Indoor, Kitchen
# training images	89490	55870	39402	38903
# test images	3793	2351	1642	1691
# train instances	421243	163512	114452	160794
# test instances	17786	7159	4826	7010

A1.3. Fully- *vs*. Self-supervised Pretraining

As discussed in the implementation details, our OW-DETR framework employs a ResNet-50 backbone that is pretrained on ImageNet1K [2] in a self-supervised manner [1] (DINO) without labels. Such a pretraining mitigates a likely open-world setting violation, which could occur in fully-supervised (FS) pretraining, with class labels, due to possible overlap with the novel classes. Here, we additionally evaluate the performance of employing the ResNet-50 backbone, which is pretrained in an FS manner. Tab. A4 shows the performance comparison between FS and DINO pretraining of ResNet-50. We observe that DINO pretraining enables a stronger backbone and achieves improved performance over FS pretraining for OWOD while additionally mitigating the violation in open-world setting.

A2. Additional Qualitative Results

OWOD comparison: Figs. A1 and A2 show qualitative comparisons between ORE [4] and our proposed OW-DETR on example images in MS-COCO test-set. For each example image, detections of ORE are shown on the left, while the predictions of our OW-DETR are shown on the

Table A4. Comparison of OW-DETR when using ImageNet1K pretrained ResNet-50 trained in (i) fully-supervised (FS) setting using class labels and (ii) self-supervised (DINO) setting without class labels. Note that the FS backbone violates OWOD settings due to overlap between pretraining (*annotated*) classes and unknowns. Hence, we utilize DINO ResNet50 for a fair OWOD evaluation.

Backbone	**Task 1**		**Task 2**		**Task 3**		**Task 4**
	U-Recall	mAP	U-Recall	mAP	U-Recall	mAP	mAP
FS	6.2	57.6	5.6	40.2	4.1	30.0	27.2
DINO	**7.5**	**59.2**	**6.2**	**42.9**	**5.7**	**30.8**	**27.8**

right. In general, we observe that the proposed OW-DETR obtains improved detections for the unknown objects, in comparison to ORE. *E.g.*, in top row of Fig. A1, while ORE fails to detect the *refrigerator* (unknown in Task 1) in the left part of the image as unknown, our OW-DETR correctly predicts it as unknown. Similarly, in Fig. A2 (top row), ORE wrongly predicts *traffic light* on a road sign that is a true unknown, whereas our OW-DETR correctly detects it as an unknown object. These results show that the proposed contributions (attention-driven pseudo-labeling, novelty classification and objectness branch) in OW-DETR enable better reasoning w.r.t. the characteristics of unknown objects leading towards a more accurate detection in the open-world setting.

Evolution of predictions: Figs. A3 and A4 illustrate an evolution of predictions when evaluating the proposed OW-DETR in different tasks of the OWOD setting on MS-COCO images. For each image, the objects detected by our OW-DETR when trained only on Task-1 classes is shown on the left. Similarly, the predictions after incrementally training with Task 2 classes is shown on the right. In the top row of Fig. A3, a *parking meter* (unknown in Task 1) is correctly detected as an unknown object during Task 1 evaluation and is rightly predicted as known class (*parking meter*) during Task 2 after learning it incrementally. In Fig. A4 (top row), the unknown objects (*giraffe* and *zebra*) are localized accurately but they are confused as a known class (*horse*) during Task 1, which can be attributed to visual similarity of these unknown objects with the known class *horse*. However, these are correctly classified when the OW-DETR is trained incrementally in Task 2 with *giraffe* and *zebra* included as new known classes. In the bottom row of Fig. A4, despite the localization not being accurately performed, multiple *traffic lights* are correctly predicted as an unknown class in Task 1 and these are detected accurately in Task 2 after incremental learning. These results show promising performance of our OW-DETR in initially detecting likely unknown objects and later accurately detecting them when their respective classes are incrementally introduced during the continual learning process.

In summary, these additional quantitative and qualitative results along with those in the main paper show the benefits of our proposed contributions in detecting unknown objects in an open-world setting, leading towards a more accurate OWOD detection.

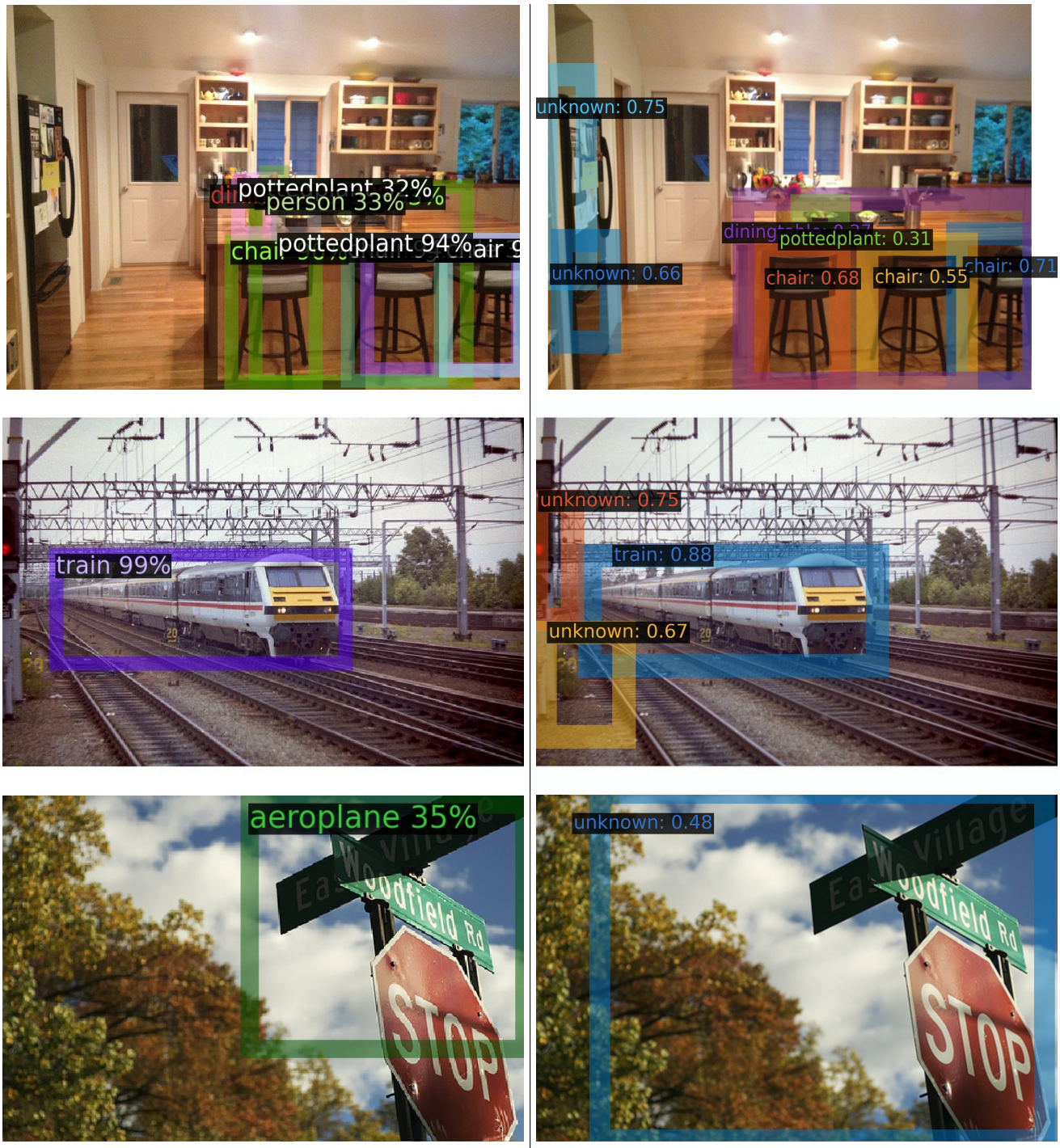
A3. Societal Impact and Limitations

The open-world learning is a promising real-world setting which incrementally discovers novel objects. However, situations can arise where a particular object or fine-grained category must not be detected due to privacy or legal concerns. Similarly, an incremental model should be able to *unlearn* (or forget) certain attributes or identities (object types in our case) whenever required. Specific solutions to these problems are highly relevant and significant, however, beyond the scope of our current work.

Although our results in Table 1 (main paper) demonstrate significant improvements over ORE in terms of Recall and mAP, the performances are still on the lower side due to the challenging nature of the open-world detection problem. We hope that this work will inspire further efforts on this challenging but practical setting.

A4. Additional Implementation Details

The multi-scale feature maps extracted from the backbone are projected to feature maps with 256-channels (D) using convolution filters and used as multi-scale input to deformable transformer encoder, as in [7]. We use the PyTorch [5] library and eight NVIDIA Tesla V100 GPUs to train our OW-DETR framework. In each task, the OW-DETR framework is trained for 50 epochs and finetuned for 20 epochs during the incremental learning step. Following [7], we train our OW-DETR using the Adam optimizer with a base learning rate of 2×10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and weight decay of 10^{-4}. For finetuning during incremental step, the learning rate is reduced by a factor of 10 and trained using a set of 50 stored exemplars per known class.

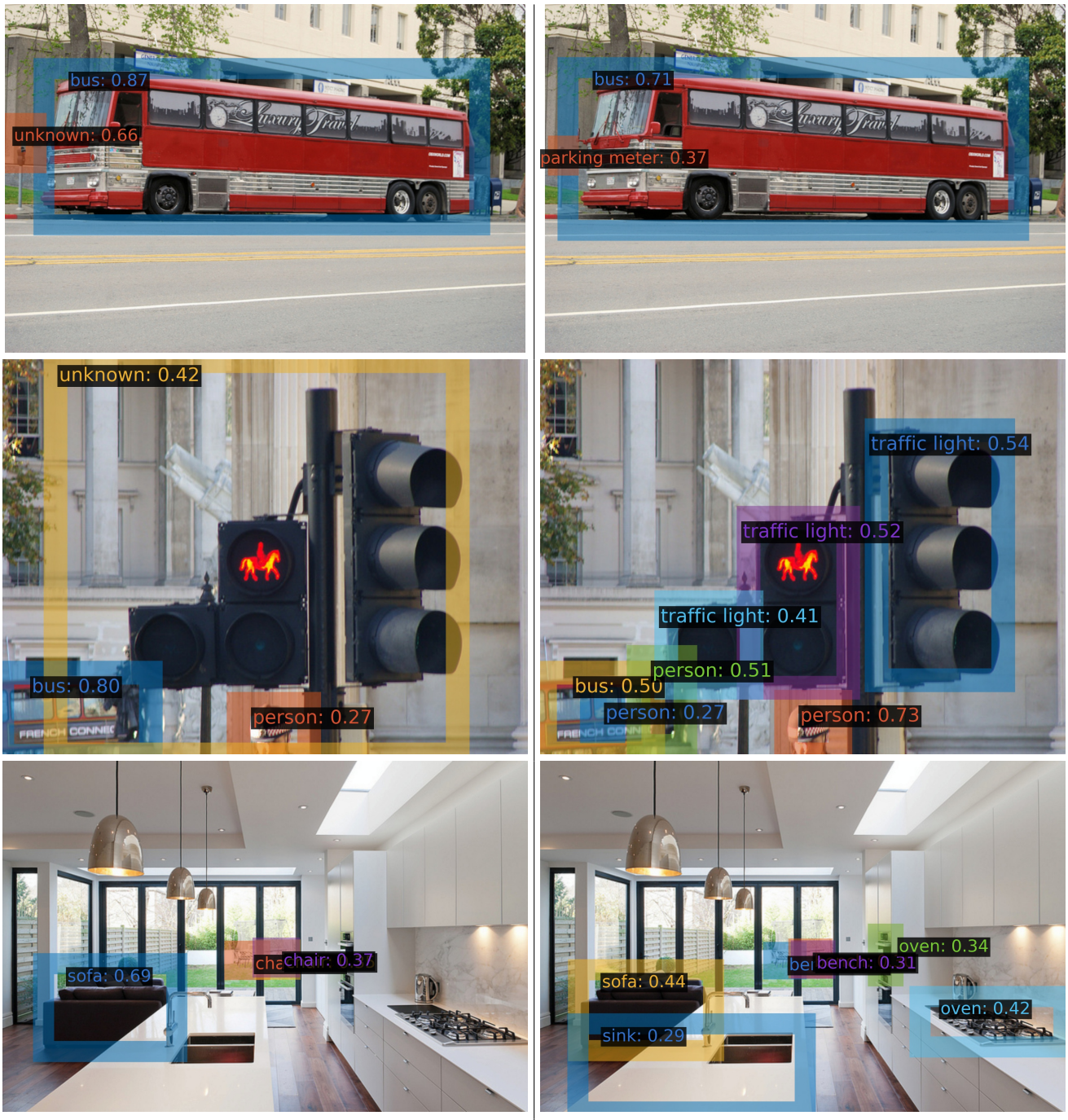


ORE [4] **Ours: OW-DETR**

Figure A1. **OWOD qualitative comparison between ORE [4] and our OW-DETR on example images in the MS-COCO test-set for Task 1 evaluation.** The predictions of ORE are shown on the left, while those from our OW-DETR are shown on the right. We observe that, in comparison to ORE, our OW-DETR obtains improved detections for the unknown instances. *E.g.*, in top row, the *refrigerator* (unknown in Task 1) in the left part of the image is detected as unknown by OW-DETR, while it is missed by ORE. Similarly, in the second row, *traffic light* (not part of known classes in Task 1) in the left part of the image are detected by our OW-DETR. Furthermore, while ORE wrongly detects the sign boards as an *aeroplane* in the third row, our OW-DETR detects an unknown object in its place. See Fig. A2 for more examples. These results show that the proposed OW-DETR achieves improved detection of unknown objects, in comparison to ORE.

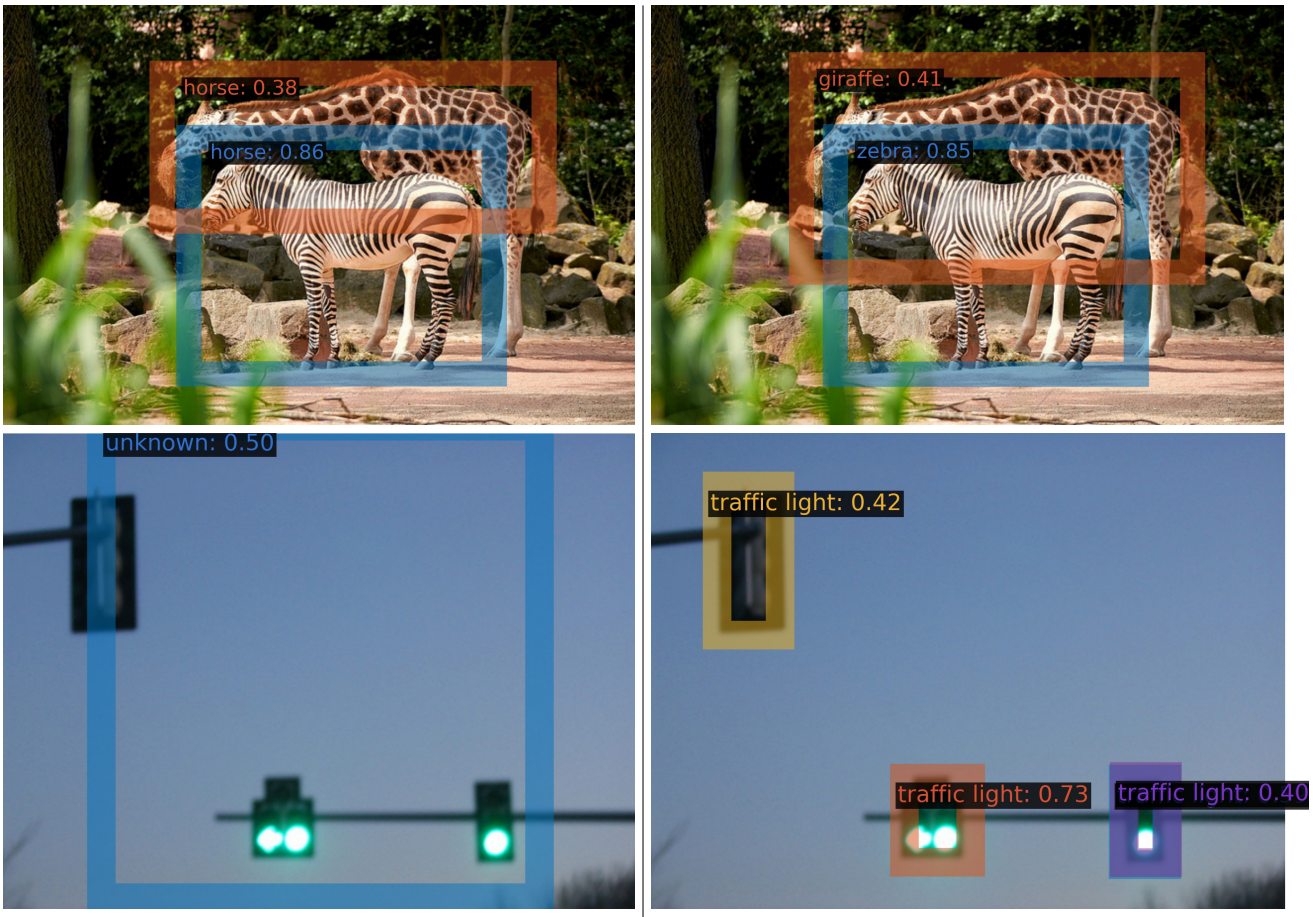
ORE [4] **Ours: OW-DETR**

Figure A2. **OWOD qualitative comparison between ORE [4] and our OW-DETR on example images in the MS-COCO test-set for Task 2 evaluation.** The predictions of ORE are shown on the left, while those from the proposed OW-DETR are shown on the right. We observe that, in comparison to ORE, our OW-DETR achieves promising detections for the unknown objects. *E.g.*, in top row, ORE wrongly predicts *traffic light* on a road sign (true unknown), whereas our OW-DETR correctly detects it as an unknown object. In addition, our OW-DETR also detects the smaller *traffic light* accurately. In the second row, while ORE detects cupboards as *oven*, our OW-DETR detects it as unknown. Furthermore, ORE detects multiple objects on *fire hydrant*, which is mitigated by our OW-DETR. These results show that the proposed OW-DETR captures better reasoning w.r.t. unknown objects, in comparison to ORE. See Sec. A2 for additional details.



Task 1 evaluation **Task 2 evaluation**

Figure A3. **Illustration showing the evolution of predictions of the proposed OW-DETR in the OWOD setting on MS-COCO images.** The objects detected by our OW-DETR when trained only on Task-1 classes is shown on the left. The predictions for the same images after incrementally training with Task 2 classes is shown on the right. In the top row, an unknown prediction during Task 1 evaluation is correctly predicted as *parking meter* during Task 2 evaluation. In the second row, *traffic lights* that are correctly detected as unknown objects during Task 1 evaluation are correctly detected as known objects during Task 2 evaluation. In the third row, potential unknown objects (*bench*) are detected but confused as *chair* due to their visual similarity during Task 1. However, they are correctly classified in Task 2 after *bench* class is incrementally learned. These results show promising performance of our OW-DETR in initially detecting potential unknown objects and later correctly detecting them when their corresponding classes are incrementally introduced for learning.



Task 1 evaluation **Task 2 evaluation**

Figure A4. **Illustration showing the evolution of predictions of the proposed OW-DETR in the OWOD setting on MS-COCO images.** On the left: The objects detected by our OW-DETR when trained only on Task 1 classes. On the right: predictions for same images after incrementally training with Task 2 classes. In the top row, although the unknown objects (*giraffe* and *zebra*) are localized accurately, they are confused as a known class (*horse*) during Task 1. However, these are corrected to their actual labels when trained incrementally in Task 2. In the bottom row, despite being localized not so accurately, multiple *traffic lights* are correctly predicted as unknown class in Task 1 and these are detected accurately in Task 2. These results show promising performance of our OW-DETR in initially detecting potential unknown objects and later correctly detecting them when their corresponding classes are incrementally introduced.

References

[1] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 2

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 2

[3] Akshay Dhamija, Manuel Gunther, Jonathan Ventura, and Terrance Boult. The overlooked elephant of object detection: Open set. In *WACV*, 2020. 1

[4] KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Towards open world object detection. In *CVPR*, 2021. 1, 2, 4, 5

[5] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 2019. 3

[6] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 2

[7] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2021. 2, 3