Supplementary Material for "O1O: Grouping of Known Classes to Identify Unknown Objects as Odd-One-Out"

Mısra Yavuz[1] and Fatma Güney[1]

Department of Computer Engineering, Koç University
KUIS AI Center
{myavuz21,fguney}@ku.edu.tr

Abstract. This supplementary document provides our superclass formulation (Section 1), implementation details for our method (Section 2), additional ablations for superclass groups as well as source and number of pseudo-labels (Section 3), and qualitative examples including some failure cases and comparison to previous methods (Section 4).

1 Superclass Grouping

The OWOD benchmarks are based on the set of COCO classes, which follow a superclass hierarchy specified in the official COCO annotation files. Following the same hierarchy and the benchmark splits, superclasses introduced at each task for both S-OWOD and M-OWOD benchmarks can be found in Table 1. Since S-OWOD tasks are designed to have perfect superclass separation, each superclass introduced in a task is learned from scratch and to completion. On the other hand, on M-OWOD, a task might have only some of the classes belonging to a certain superclass in the current training set. For example, in M-OWOD Task 1, only *bird*, *cat*, *cow*, *dog*, *horse*, and *sheep* classes are learned from the animal superclass; while *elephant*, *bear*, *zebra*, and *giraffe* are learned in Task 2. Regardless, if any class of a superclass is present, we introduce it at the current task. Unseen classes are still considered unknown, following the benchmark rules.

Benchmark	Task	Superclasses Used During Training
S-OWOD	1	animal, person, vehicle
	2	accessory, appliance, furniture, outdoor
	3	food, sports
	4	electronic, indoor, kitchen
M-OWOD	1	animal, electronic, furniture, kitchen, person, vehicle
	2	accessory, animal, appliance, outdoor, vehicle
	3	food, sports
	4	electronic, furniture, indoor, kitchen

Table 1: **Superclass Separation Across Tasks.**

As the unseen classes become available in the continual learning procedure, we continue training the specific weights of the partially introduced superclasses. We underline partially learned superclasses in Table 1. Although partially learning the superclass representations hurt the odd-one-out scoring in theory, our experimental results show that our method can still generalize to such a setting.

2 Implementation Details

Our pipeline can be considered as a two-step approach: first, we train the RPN to extract pseudo-labels, then we train our open-world detection model.

Pseudo-labels Extraction Step: We train the RPN with surface normal maps estimated by the DPT-Hybrid model [11] from the Omnidata repository [3] as in GOOD [6] and store the proposals. We train the RPN of each task from scratch to obey the benchmark rules and not use any unknown instances for training. We select the pseudo-labels by thresholding the RPN's confidence score with 0.5 and then merge them with real targets by applying Non-Maximum Suppression with an IoU threshold of 0.5.

OWOD Step: Our detection model is based on the DN-DAB-Deformable DETR [9,8] architecture, with the same feature extraction backbone as previous work: ResNet-50 FPN [5] pre-trained on ImageNet [2] in a self-supervised manner [1]. We decompose our loss function into localization and classification. The localization losses are applied to queries matched to both knowns and pseudo-unknowns, while classification losses are only applied to queries matched to known targets. We use loss coefficients 5 for bounding box loss, 2 for giou loss, 2 for per-class classification loss, and 2 for superclass classification loss. We follow the default hyperparameters of DN-DAB-Deformable DETR for the denoising tasks. In addition, we add a denoising task for superclass classification, using the same hyperparameters as the per-class denoising loss, to randomly flip the ground truth labels for noised queries and teach the model to correct them.

M-OWOD Bug Fix: We found duplicate image IDs in the test set of M-OWOD, which can affect the performance of DETR-based methods based on how the matching operation is done during evaluation. We detected that OW-DETR and PROB were affected by this bug. The codes of more recent DETR-based methods, CAT and USD, were not available to check. In Table 2, we report performance changes of OW-DETR and PROB on Task 1 after fixing the bug.

Method	U-Recall (↑)	mAP (↑)
OW-DETR†	7.6 → 10.1	58.8 → 65.6
PROB	19.4 → 28.3	59.5 → 66.4
O1O	33.4 → 49.3	58.4 → 65.1

Table 2: **Performance Improvements after M-OWOD Bug Fix**

We trained OW-DETR from scratch for Task 1 (denoted with †), because we couldn't reproduce their results with the checkpoints provided. Therefore, we can only report the performance change in Task 1. PROB's results on all tasks, with and without the fix, are also reported in Table 1 of the main paper. O1O's performance, with and without the fix, are consistent. We achieve significantly higher Unknown Recall while maintaining competitive performance in Known mAP. Compared to PROB, we obtain +74.2% increase in Unknown Recall with only -2.0% decrease in Known mAP.

3 Additional Ablations

3.1 Varying Superclass Groups

Group ID	**Superclasses**
A	living beings, objects
B	pets, wild animals, land vehicles, air/water vehicles, seating furniture, household items, person
C	pets, farm animals, wild animals, bikes (2-wheel), land vehicles (4-wheel), air/water vehicles, seating furniture, electronics, household items, person
D (Default)	animals, vehicles, person, furniture, electronics, kitchen

(a) **M-OWOD**

Group ID	**Superclasses**
A	living beings, objects
B	domestic animals, wild animals, person, land vehicles, air/water vehicles
C	large animals, medium animals, small animals, person bikes (2-wheel), land vehicles (4-wheel), air vehicles, water vehicles
D (Default)	animals, person, vehicles

(b) **S-OWOD**

Table 3: **Different Groupings of Classes into Superclasses**

To evaluate the robustness of our method to different superclass groups, we explore broader and narrower sets of superclasses than the default grouping (D) used in the main paper in Table 3. First, in A, we combined all non-living things into a single objects class and grouped animals and humans as living beings. Conversely, we explored more fine-grained groupings in B and C by dividing animals into categories like pets, farm animals, domestic or wild animals, or by

size into large, medium, and small. Additionally, we organized vehicles based on the number of wheels and their typical usage. The default (D) groups for each benchmark are formed using the official superclass labels in COCO annotations. We used the default groups in our main results in the paper to comply with recognized standards and minimize subjective biases.

Group ID	M-OWOD		S-OWOD	
	U-Recall	mAP	U-Recall	mAP
A	40.0	64.9	48.2	72.1
B	48.5	65.0	51.3	72.5
C	48.9	64.8	51.4	70.8
D (Default)	49.3	65.1	49.8	72.6

Table 4: **Results by Varying Superclass Groups**

We report the results of this ablation on Task 1 of both benchmarks in Table 4. Group A consists of living beings and objects. On M-OWOD, forcing various objects such as furniture, electronics, kitchen items, and vehicles into a single group leads to an oversimplification. This results in a generic distribution to which any class can belong including unknowns, hence the drop in unknown recall. For S-OWOD, since objects only consist of vehicles, the setup is very close to default, with the only difference being the merging of animal and person classes. As a result, the performance drop is not as significant. Note that animal and person classes are similar enough that the 'living beings' distribution does not unintentionally cover representations of other classes.

For B, the results are similar to the default case. As superclasses narrow down in Group C, known performance drops while unknown performance increases slightly. This can be attributed to unknowns standing out more as odd-one-outs as superclass representation groups become smaller in the feature space. Overall, these results confirm that our method performs well across different superclass groups with varying granularity, supporting the robustness of our approach.

3.2 Source of Pseudo-labels

To analyze the strength of geometric pseudo-labels with our method, we performed an additional study to compare them to RGB-based proposals. We trained the RPN module using three sources of input: RGB images, predicted depth maps, and predicted surface normal maps. For clarity, the detection model always receives the RGB images as input. Here, we only modify the input to the RPN module. As shown in Table 5, while the Unknown Recall of RGB and surface normals are close on the M-OWOD benchmark, the difference becomes visible in S-OWOD. The reason behind is the effect of dataset size. The M-OWOD Task 1 dataset is approximately one-sixth the size of the S-OWOD Task

Source	M-OWOD		S-OWOD	
	U-Recall (↑)	mAP (↑)	U-Recall (↑)	mAP (↑)
RGB	50.5 → 51.5	53.2 → 62.8	48.7	70.3
Depth	46.8 → 48.0	54.9 → 63.0	45.9	72.6
Normals	50.7 → 51.6	55.0 → 63.6	52.9	71.0

Table 5: **Varying the Source of Pseudo-Labels**

1 dataset. Hence, any use of extra supervision helps the model to improve its recall significantly. However, in the abundance of training instances, such as in S-OWOD, the detection model already leverages the full spectrum of information available from RGB input. Therefore, when surface normals are used, it outperforms the RGB variant by +4.2 in Unknown Recall with a +0.7 better Known mAP. In both benchmarks, RGB pseudo-labels cause the most confusion, hurting the known mAP considerably.

Depth alone has the poorest performance among the three methods in terms of unknown performance. Due to the tradeoff between known and unknown performance, this leaves room for known performance to be relatively higher. Overall, surface normals yield the highest unknown recall on both benchmarks and the least negative impact on known performance.

Furthermore, we performed additional experiments without the superclass component to showcase the improvements gained by using them (trained without → with). We used the M-OWOD benchmark since the smaller dataset size allows for faster experiments. The results confirm that superclasses generalize to pseudo-labels from different sources, and shaping the representation helps both Unknown Recall and Known mAP across all three settings.

3.3 Number of Pseudo-labels

After deciding which source of pseudo-labels to use, the next question is how many to use. We conducted experiments with different numbers of pseudo-labels on M-OWOD Task 1, utilizing surface normals as the source. We implement the number of pseudo-labels as a constraint after the NMS module and confidence thresholding. This means that after removing overlapping or low-confidence boxes, we select the top-k boxes, where k is the number of known ground truth targets plus the number of pseudo-labels.

We report how Known mAP values evolve against Unknown Recall in Fig. 1. We start with small numbers such as 1, 3, and 5, which are typically used in previous work [7,4,10], and increase the number up to half of total number of queries, which is 100. Our experiment results demonstrate the trade-off between known and unknown performance. As the number of pseudo-labels increases, Unknown Recall shows a clear improvement until the number 20. Known mAP has an unstable response to the number of pseudo-labels until the number 5, remaining in a range, after which it exhibits a clear decreasing trend, as expected.

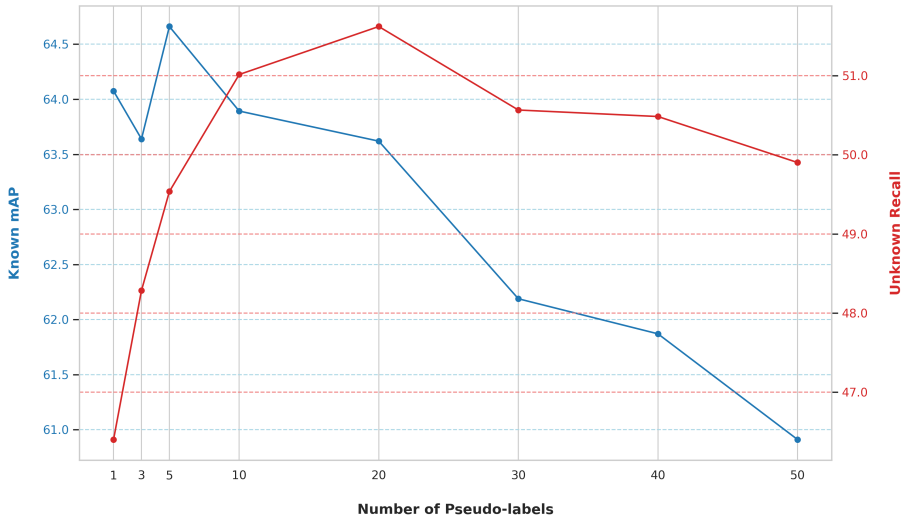

Fig. 1: **Number of Pseudo-labels Ablation.**

After the number 20, both Known mAP and Unknown Recall decline because the noisy and inaccurate pseudo-supervision dominates the signals coming from the actual ground truth supervision. As the equilibrium point, we use 20 pseudo-labels per image in our method. This is not implemented as a hard constraint, meaning if there are fewer than 20 pseudo-boxes with a confidence score greater than our confidence threshold (0.5), we use exactly how many there are.

4 Qualitative Examples

We visualize the top-10 proposals of OW-DETR [4], PROB [12] and our model O1O in Fig. 2 for S-OWOD and in Fig. 3 for M-OWOD. In most cases, OW-DETR and our method have a better ranking than PROB, prioritizing known objects with higher confidence and representing unknown predictions with lower scores. As shown in Fig. 2, our model can locate the monitor, frame and tape precisely in the first row, can detect the box on the ground and and the tires of the plane as separate objects, and finds the pans and the towel in the third row, while previous work failed to do so. Some detected unknowns are not even annotated in the official COCO dataset, therefore not contributing to the Unknown Recall metric, such as tape and towel. Lastly, without any unknowns present, such as the last two rows, our method can label known objects without any confusion and does not produce unnecessary unknown predictions with high confidence scores. Our method performs similarly on M-OWOD as illustrated in Fig. 3. It can precisely detect the lamps, frames, buildings, or small objects in the background while maintaining a meaningful ranking between known and unknown predictions.

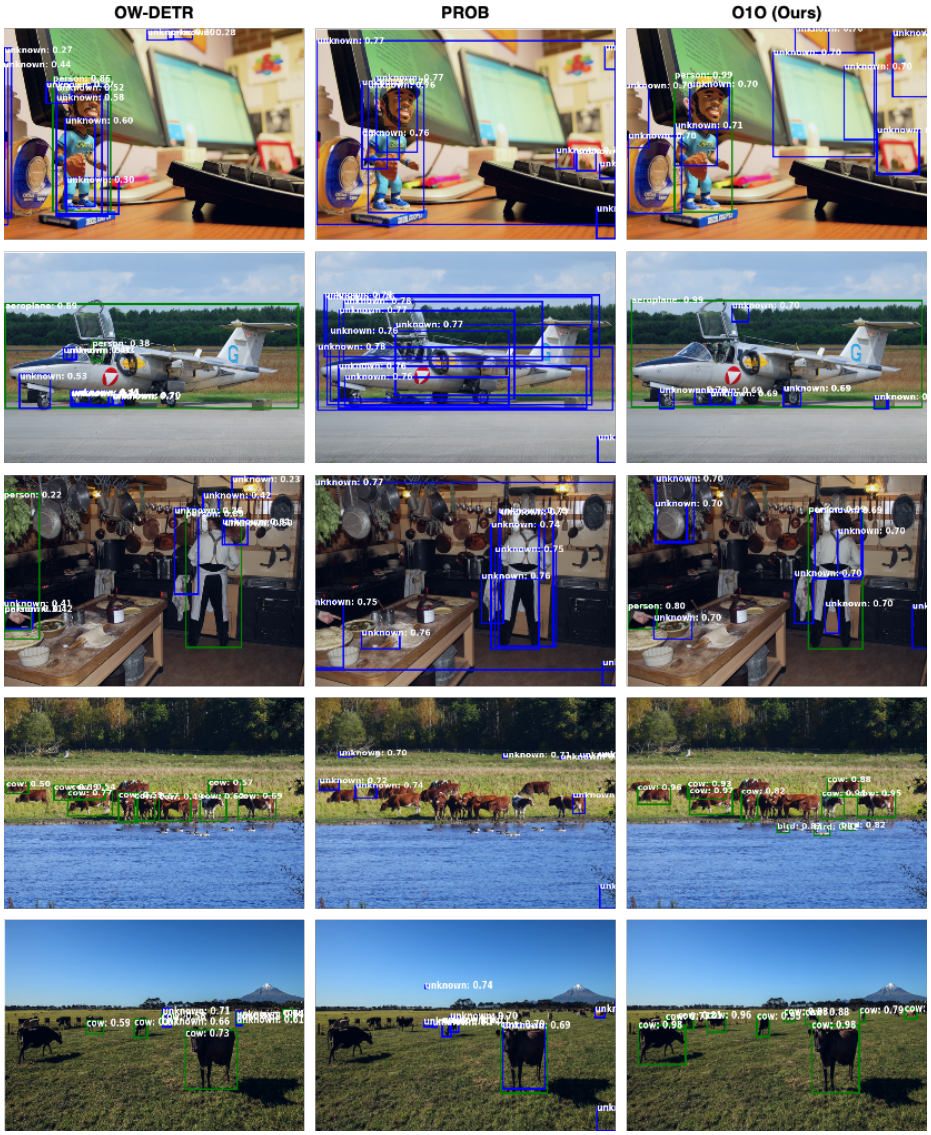

Fig. 2: **Qualitative Comparison on S-OWOD.** Visualizations of top-10 proposals of OW-DETR, PROB, and O1O(Ours).

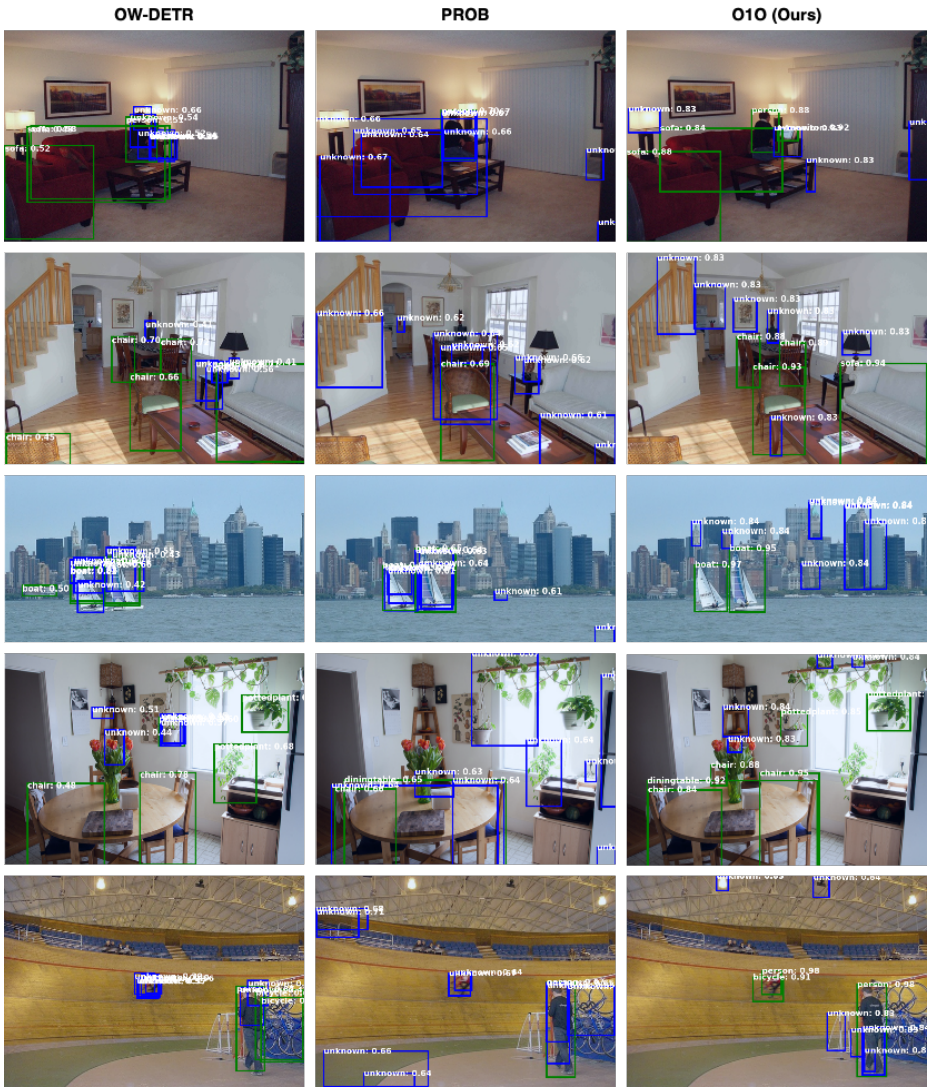

Fig. 3: **Qualitative Comparison on M-OWOD.** Visualizations of top-10 proposals of OW-DETR, PROB, and O1O(Ours).

Failure Cases: Since our pseudo-labels depend on predicted surface normals, O1O is vulnerable to cases when surface normals cannot be estimated accurately. One failure mode is when textual differences in a large background, like the sky or floor, cause relative differences in local normal values. This encourages the model to predict objects in such regions, as in the first row of Fig. 4. Another failure mode occurs in complex indoor scenes with many objects, sometimes because the objects have flat shapes, such as keyboards, or because of crowded regions with entangled objects, as in the second row of Fig. 4. Our model occasionally ranks blank regions that do not correspond to real objects higher than known classes, such as the person and dog in the upper left, the table in the upper right, the sofa/chair in the lower left, or the person in the lower right of Fig. 5.

Incremental Learning: In Fig. 6 and 7, we visualize the predictions of O1O across different tasks of S-OWOD and M-OWOD respectively. Different than previous top-k visualizations, we show the predictions matched to ground truth objects to showcase the development clearly. Although some classes are unknown in the first tasks, our model can still locate them. As the space of known categories expands, O1O learns to label them correctly without forgetting the previously learned classes, showing the effectiveness of our exemplar replay fine-tuning.

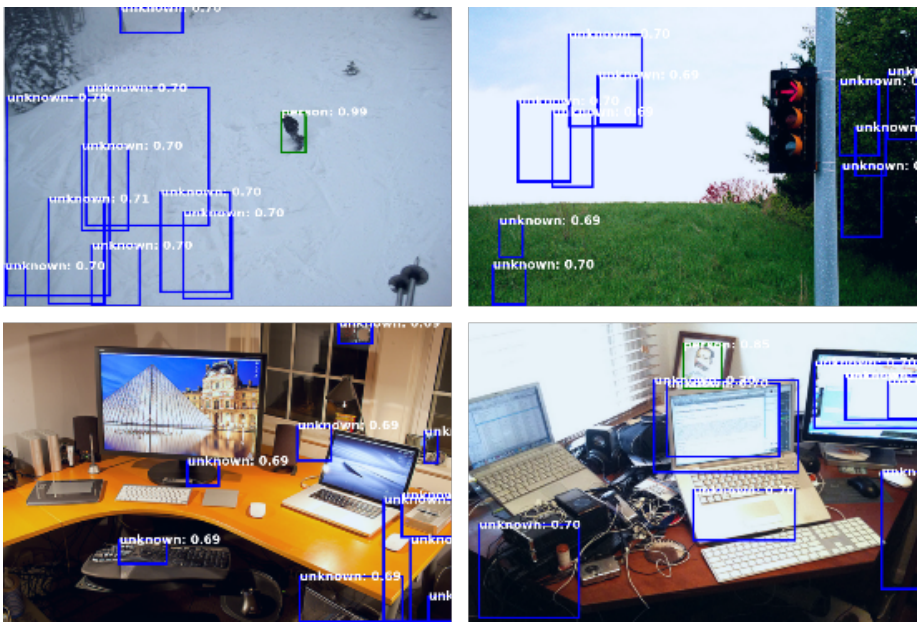

Fig. 4: **Failure cases on S-OWOD.** Due to reliance on estimated surface normals, our model may generate redundant pseudo-labels in areas with local texture differences (**top**) or struggle to detect objects in crowded scenes (**bottom**).

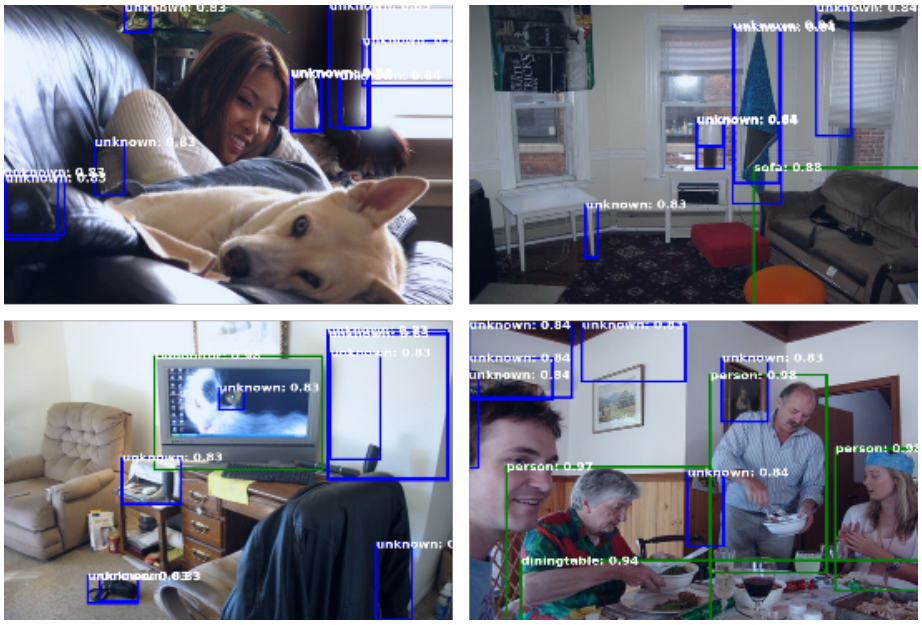

Fig. 5: **Failure cases on M-OWOD.** Examples of ranking redundant unknown predictions higher than some known instances.

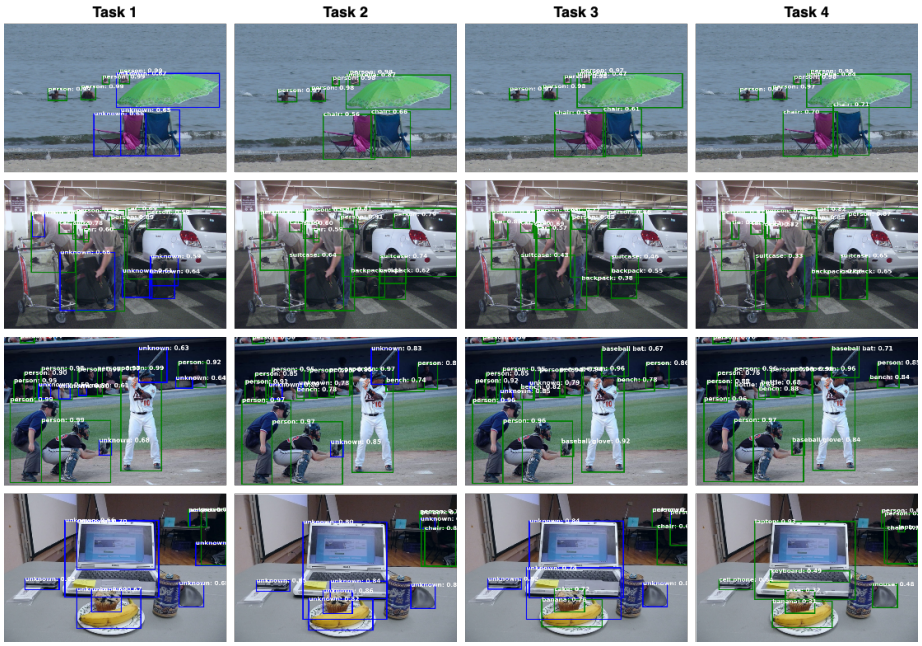

Fig. 6: **Incremental Learning Performance Across Tasks.** Visualizations of predictions matched to ground truth objects across the tasks of S-OWOD.

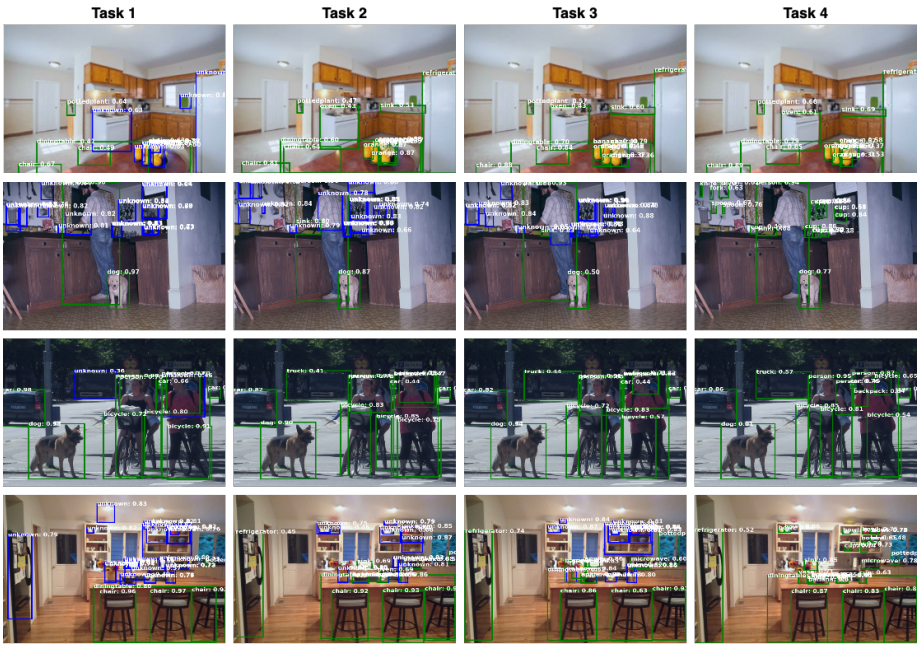

Fig. 7: **Incremental Learning Performance Across Tasks.** Visualizations of predictions matched to ground truth objects across the tasks of M-OWOD.

References

1. Caron, M., Touvron, H., Misra, I., J'egou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: ICCV (2021)
2. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: CVPR (2009)
3. Eftekhar, A., Sax, A., Malik, J., Zamir, A.: Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In: ICCV (2021)
4. Gupta, A., Narayan, S., Joseph, K., Khan, S., Khan, F.S., Shah, M.: OW-DETR: open-world detection transformer. In: CVPR (2022)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2015)
6. Huang, H., Geiger, A., Zhang, D.: GOOD: Exploring geometric cues for detecting objects in an open world. In: ICLR (2023)
7. Joseph, K.J., Khan, S., Khan, F.S., Balasubramanian, V.N.: Towards open world object detection. In: CVPR (2021)
8. Li, F., Zhang, H., Liu, S., Guo, J., Ni, L.M., Zhang, L.: DN-DETR: Accelerate detr training by introducing query denoising. In: CVPR (2022)
9. Liu, S., Li, F., Zhang, H., Yang, X., Qi, X., Su, H., Zhu, J., Zhang, L.: Dab-detr: Dynamic anchor boxes are better queries for detr. In: ICLR (2021)
10. Ma, S., Wang, Y., Fan, J., yu Wei, Y., Li, T.H., Liu, H., Lv, F.: CAT: localization and identification cascade detection transformer for open-world object detection. In: CVPR (2023)
11. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. In: ICCV (2021)
12. Zohar, O., Wang, K.C., Yeung, S.: PROB: probabilistic objectness for open world object detection. In: CVPR (2023)