Detecting Unknown Objects via Energy-based Separation for Open World Object Detection

Supplementary Material

Supplementary Outline

In this supplementary material, we provide additional details and analyses that complement the main paper. Section A covers implementation details and benchmark settings. Section B presents ablation studies. Section C presents extended analysis including hyperparameter studies and visualizations. Section D provides computational cost analysis. Section E presents qualitative results.

A. Details of Experiment Settings

Implementation Details We follow the implementation details of OrthogonalDet [25]. OrthogonalDet is based on a Fast R-CNN [5] architecture and a ResNet-50 [8] backbone pre-trained on ImageNet [23]. We used a linear classifier for the classification and Batch Normalization [9] for the objectness branch, following OrthogonalDet. Our DEUS framework is trained using the AdamW optimizer [14]. RoI pooling is applied to 500 random proposals, which are then fed into the detection heads for localization, objectness, and classification training. During training, DEUS is supervised using ground-truth annotations and pseudo-labels selected based on our pseudo-labeling process in Sec. A. During inference, we omit prediction randomness by using 1,000 pre-defined object proposals, which are pruned via non-maximum suppression at an IoU threshold of 0.6. The final detections are selected using a score threshold of 0.10. For loss hyperparameters and balancing, we set the focal loss parameters $\alpha = 0.25$ and γ = 2.0. The classification loss weight is 2.0, L1 regression loss weight is 5.0, and GIoU loss weight is 2.0. Our proposed EUS and EKD loss weights are both set to 1.0, and ETF energy margin m is 0.5. Following OrthogonalDet [25] and RandBox [27], we train for 20,000 steps in Task 1 and 15,000 steps in Tasks 2–4. For memory replay, while prior works [30] report using 50 exemplars per class, we follow the actual pre-defined exemplar lists provided by OrthogonalDet and RandBox for fair comparison, which amount to 1,743, 2,361, and 2,749 images (including all previously learned classes) for Tasks 2, 3, and 4, respectively. We used three NVIDIA RTX 4090 GPUs with a batch size of 12 per GPU, employing AMP with bfloat16 precision for efficient training. Our implementation is based on the MM-Detection [2] framework.

Pseudo-labeling process. Following the baseline approach [25], we provide pseudo-labels for unknown objects as supervision signals to the detector. While the baseline method simply selects a fixed number τ (e.g., 20) of proposals with the highest objectness scores among those unmatched with ground truth, we introduce a more sophisticated pseudo-labeling process. First, the number of pseudo-labels k_{pseudo} is determined proportionally to the known ground truth count, following a dynamic scaling mechanism $k_{\text{pseudo}} = k_{\text{gt}} \cdot \max\left(1, \frac{2\tau}{N_{\text{known}}}\right)$, where N_{known} represents the number of learned classes including the current task and we set $\tau = 20$. This provides more unknown supervision in early stages when fewer classes have been learned, gradually reducing as the model learns more classes. Second, we only consider objects whose bounding-box length is at least 0.5 times the minimum image size as unknown labels, filtering out excessively small or noisy detections. Lastly, we use final unknown logits (Eq. 14 of the main paper) as our criterion, selecting only objects with scores above zero as pseudo-labels to maximize avoidance of known objects. This creates a beneficial self-improving cycle: our EUS method enables better unknown detection, which provides higher-quality pseudo-labels that further enhance unknown representation learning, leading to progressively improved performance without additional unknown supervision.

M-OWODB The superclass-mixed benchmark [10] groups all Pascal VOC classes and data into the initial task, Task 1. The remaining 60 classes from MS-COCO are then divided into three incremental tasks, introducing semantic drifts. Due to the overlap between Pascal VOC and MS-COCO classes, super-categories are mixed across tasks. Consequently, while M-OWODB ensures non-overlapping classes between sequential tasks, super-categories such as

S.Table 1. Benchmark Configuration for M-OWODB, S-OWODB, and RS-OWODB.

Metrics	M-OWODB				S-OWODB				RS-OWODB			
	Task 1	Task 2	Task 3	Task 4	Task 1	Task 2	Task 3	Task 4	Task 1	Task 2	Task 3	Task 4
Classes	VOC Classes	Outdoor, Accessories, Appliances, Truck	Sports, Food	Electronic, Indoor, Kitchen, Furniture	Animals, Person, Vehicles	Outdoor, Accessories, Appliances, Furniture	Sports, Food	Electronic, Indoor, Kitchen	Baseballfield, Dam, Groundtrackfield, Stadium, Vehicle	Basketballcourt, Expressway Area, Harbor, Storagetank, Windmill	Aeroplane, Bridge, Expressway Station, Overpass, Tenniscourt	Airport, Chimney, Golffield, Ship, Trainstation
# of Classes	20	20	20	20	19	21	20	20	5	5	5	5
# of training images	16,551	45,520	39,402	40,260	89,490	55,870	39,402	38,903	5,394	3,445	4,111	3,247
# of training objects	47,223	113,741	114,452	138,996	421,243	163,512	114,452	160,794	18,378	9,928	10,093	29,674
# of test images	10,246				4,952				11,738			
# of test objects	14,976	4,966	4,826	6,039	17,786	7,159	4,826	7,010	8,212	32,695	20,614	37,967

Vehicles and Animals may still overlap across tasks. The left section of S.Table 1 shows the benchmark configuration of M-OWODB, where Task 1 consists of the VOC classes that can share the super-categories with subsequent tasks.

S-OWODB The superclass-separated benchmark [7] provides a stricter MS-COCO split compared to M-OWODB. While M-OWODB allows data leakage across tasks due to the inclusion of different classes from the same super-categories (*e.g.*, most classes from vehicle and animal super-categories are introduced in Task 1, while related classes such as truck, elephant, bear, zebra, and giraffe appear in Task 2), S-OWODB groups all categories within a super-category into a single task rather than spreading them across tasks. As shown in the middle of S.Table 1, Task 1 contains all related classes from Animals, Person, and Vehicles, while Task 2 includes Appliances, Accessories, Outdoor, and Furniture. This strict separation by super-categories makes S-OWODB a more challenging OWOD benchmark.

RS-OWODB To further evaluate the generalizability of DEUS, we introduce a new benchmark setting called RS-OWODB (Remote Sensing OWODB) as shown in the right side of S.Table 1. Unlike M-OWODB and S-OWODB, which focus on natural images, RS-OWODB utilizes remote sensing images. This benchmark is constructed using the DIOR [11] dataset with each task consisting of 5 classes. The benchmark maintains balanced data distribution across tasks, with the number of images and object instances evenly distributed. This benchmark provides an additional evaluation environment for the OWOD scenario in the remote sensing domain.

B. Ablation Studies

B.1. Ablation of Pseudo-labeling Process

As noted in the main paper, the ablation baseline (without EUS and EKD) differs slightly from the OrthogonalDet results due to our improved pseudo-labeling process. To isolate the contribution of DEUS from this engineering improvement, we conducted an ablation study with and without our pseudo-labeling process (Psd). S.Table 2 presents the results on M-OWODB.

S.Table 2. Ablation of the pseudo-labeling (Psd) process on M-OWODB. The best performance is highlighted in bold.

Task IDs		Task 1			Task 2			Task 3			Task 4
DEUS	Psd	Known mAP	U-Rec	H-Score	Known mAP	U-Rec	H-Score	Known mAP	U-Rec	H-Score	Known mAP
		65.1	36.3	46.6	51.2	30.2	38.0	47.3	28.7	35.7	44.7
✓		65.6	**65.3**	65.4	52.9	**66.3**	58.9	48.2	**69.2**	56.8	45.4
	✓	66.0	36.8	47.2	52.0	29.0	37.3	49.7	30.3	37.6	44.7
✓	✓	**66.2**	65.1	**65.6**	**53.3**	66.2	**59.0**	**50.1**	69.0	**58.0**	**46.0**

Comparing rows 1 vs 2 and rows 3 vs 4, DEUS clearly provides substantial performance gains regardless of whether the pseudo-labeling process is applied. Without the pseudo-labeling process, DEUS alone improves U-Rec from 36.3 to 65.3 in Task 1, demonstrating that the core methodological contribution is independent of the pseudo-labeling enhancement. The pseudo-labeling process further improves known mAP by filtering noisy or known-class proposals from pseudo-labels. Regarding pseudo-label quality, we measured matching statistics between pseudo-labels and real unknown ground-truth: across tasks, recall remains stable (20.3%→21.8%) while precision improves substantially (5.0%→12.0%), contributing to improved known mAP.

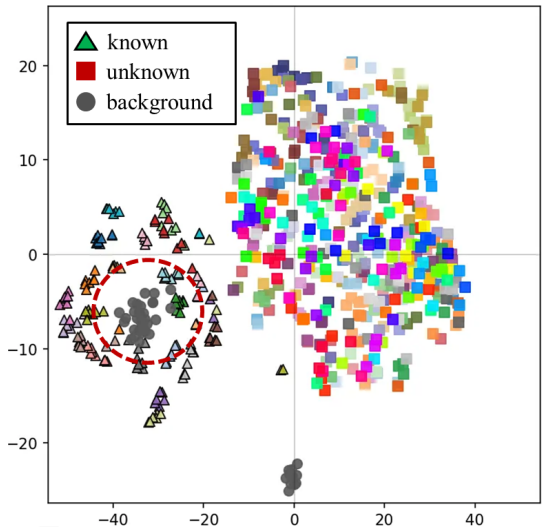
B.2. Ablation of EUS Components

We further ablated the two loss components of EUS: the energy-based margin loss $\mathcal{L}_{\text{energy}}$ ($\mathcal{L}_{en}$, Eq. 11 of the main paper) and the subspace focal loss $\mathcal{L}_{\text{subspace}}$ ($\mathcal{L}_{sub}$, Eq. 12 of the main paper). S.Table 3 presents the results on M-OWODB with EKD applied in all cases.

Comparing rows 1 and 3, $\mathcal{L}_{en}$ is the primary driver of unknown detection, achieving strong U-Rec through explicit margin enforcement. Row 2 shows that $\mathcal{L}_{sub}$ alone fails

S.Table 3. Ablation of EUS loss components on M-OWODB (with EKD applied). $\mathcal{L}_{sub}$ = subspace focal loss, $\mathcal{L}_{en}$ = energy margin loss. The best performance is highlighted in bold.

Task IDs		Task 1			Task 2			Task 3			Task 4
$\mathcal{L}_{sub}$	$\mathcal{L}_{en}$	Known mAP	U-Rec	H-Score	Known mAP	U-Rec	H-Score	Known mAP	U-Rec	H-Score	Known mAP
		66.0	36.8	47.2	52.6	40.0	45.4	**50.3**	38.9	43.9	45.9
✓		41.1	43.7	42.4	38.1	42.3	40.1	36.3	54.6	43.6	31.5
	✓	65.8	**65.3**	65.5	52.3	**66.5**	58.6	48.8	**70.3**	57.6	44.3
✓	✓	**66.2**	65.1	**65.6**	**53.3**	66.2	**59.0**	50.1	69.0	**58.0**	**46.0**



S.Figure 1. t-SNE visualization of proposals without $\mathcal{L}_{\text{subspace}}$ (using only $\mathcal{L}_{\text{energy}}$). Without the subspace loss, background proposals leak into the known space, resulting in ambiguous boundaries between known and background regions.

severely, as this auxiliary loss primarily stabilizes background proposals and cannot function without the main $\mathcal{L}_{en}$. As shown in S.Figure 1, using only $\mathcal{L}_{en}$ (row 3) effectively separates known from unknown but leaves background boundaries ambiguous due to the lack of explicit background supervision. Combining both losses (row 4) refines these boundaries, achieving the best H-Score across all tasks.

B.3. EKD on Incremental Object Detection

To further validate the effectiveness of EKD beyond OWOD benchmarks, we evaluated it on COCO-IOD (Incremental Object Detection) benchmarks with various class splits. As shown in S.Table 4, EKD consistently improved both Previous and Current mAP across all splits (10+10, 15+5, 19+1), demonstrating that the energy-based separation between task-specific heads effectively mitigates catastrophic forgetting regardless of the task configuration. Notably, the improvements are observed for both old and new classes simultaneously, confirming that EKD reduces cross-influence rather than simply trading off between them.

S.Table 4. Evaluation of EKD on COCO-IOD benchmarks with different class splits. Pre = Previous mAP, Cur = Current mAP. The best performance is highlighted in bold.

	Split	Pre	Cur	mAP
w/o $\mathcal{L}_{\text{EKD}}$	10+10	74.5	70.2	72.3
	15+5	76.3	71.8	75.2
	19+1	75.7	74.9	75.6
w/ $\mathcal{L}_{\text{EKD}}$	10+10	**76.2**	**71.8**	**74.0**
	15+5	**77.2**	**73.2**	**76.2**
	19+1	**76.5**	**75.2**	**76.4**

S.Table 5. Ablation of the energy margin m on M-OWODB. The best performance is highlighted in bold, and the second-best performance underlined.

Task IDs	Task 1			Task 2			Task 3			Task 4
m	Known mAP	U-Rec	H-Score	Known mAP	U-Rec	H-Score	Known mAP	U-Rec	H-Score	Known mAP
0.25	**66.8**	64.6	**65.7**	**53.9**	64.5	58.7	50.5	67.0	57.6	45.5
0.5	66.2	65.1	65.6	53.3	66.2	**59.0**	**50.1**	69.0	**58.0**	**46.0**
1.0	63.1	**65.5**	64.3	50.9	**66.8**	57.8	48.2	**69.5**	56.9	43.0

C. Extended Analysis

C.1. Sensitivity of Energy Margin m

We ablated the energy margin hyperparameter m in $\mathcal{L}_{\text{energy}}$. As shown in S.Table 5, $m = 0.5$ achieves the best overall balance across tasks. Smaller values ($m = 0.25$) provide weaker separation signals between known and unknown subspaces, while larger values ($m = 1.0$) enforce overly strict constraints that harm training stability.

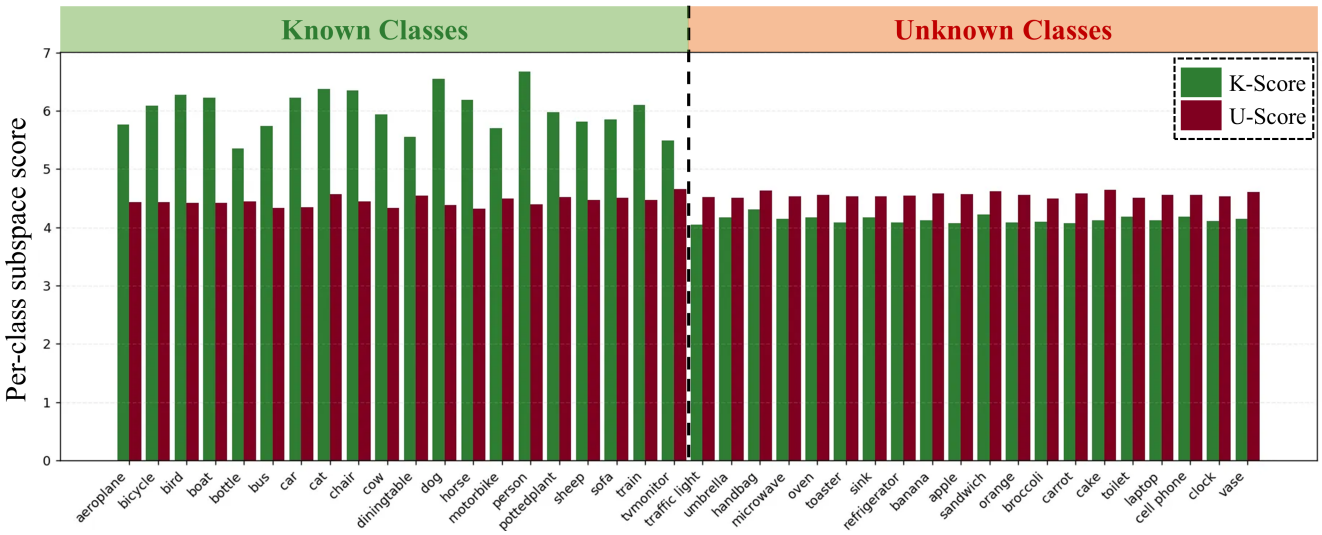
C.2. Analysis on Simplex ETF

As defined in the main paper (Sec. 3.3), the Simplex ETF [19] provides equiangular, equal-norm basis vectors, allowing subsets to form non-overlapping known and unknown spaces with a margin of $-\frac{1}{K-1}$.

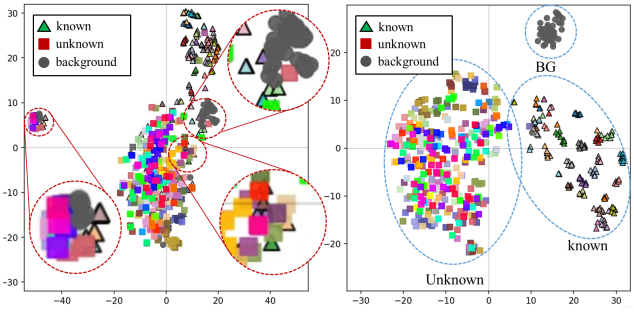
To validate the effectiveness of the known and unknown spaces constructed using the Simplex ETF, we conducted an ablation study to evaluate the impact of the number of K vectors used in the Simplex ETF. Increasing the number of K helps create known and unknown spaces with sufficient vectors to represent each space and improve the model's ability to capture unknown patterns, resulting in generally increased U-Rec performance. However, this enhanced unknown detection capability leads to a slight decrease in Known mAP due to the increased number of unknown detections, though the difference is not significant. As shown in S.Table 6, we empirically identified the optimal value for K. We observed that larger K values consistently improve unknown recall performance, with $K = 128$ achieving the highest U-Rec of

S.Table 6. Analysis of the number of K for Simplex ETF on M-OWODB. The best performance is highlighted in bold, and the second best is underlined.

Task IDs	Task 1			Task 2					Task 3				
# of K	Current mAP	**U-Rec**	**H-Score**	Previous mAP	Current mAP	Known mAP	**U-Rec**	**H-Score**	Previous mAP	Current mAP	Known mAP	**U-Rec**	**H-Score**
w/o EUS	66.0	36.8	47.2	59.2	<u>45.9</u>	52.6	40.0	45.4	<u>53.6</u>	**43.6**	<u>50.3</u>	38.9	43.9
K=32	**66.4**	62.9	64.6	**61.0**	**46.3**	**53.6**	63.8	<u>58.3</u>	**53.8**	**43.6**	**50.4**	65.5	<u>57.0</u>
K=64	66.0	<u>64.4</u>	<u>65.2</u>	<u>60.6</u>	45.5	53.0	<u>64.0</u>	58.0	53.4	43.1	50.0	<u>66.0</u>	56.9
K=128	<u>66.2</u>	**65.1**	**65.6**	**61.0**	45.7	<u>53.3</u>	**66.2**	**59.0**	53.4	<u>43.3</u>	50.1	**69.0**	**58.0**



S.Figure 2. Comprehensive per-class subspace score comparison between known (green) and unknown (red) across all Task 1 and Task 2 classes. The model was trained only on Task 1 classes (1-20). Task 1 classes show higher known scores, while Task 2 classes (21-40) show higher unknown scores, demonstrating effective known-unknown separation.



S.Figure 3. t-SNE visualization of proposals matched to actual ground-truth at inference time (Task 1). Without EUS (left), known, unknown, and background proposals are entangled. With EUS (right), the three groups are clearly separated into distinct regions.

69.0. Although there is a minor trade-off in Known mAP, the overall H-Score, which represents the harmonic mean between Known mAP and U-Rec, reaches its peak at $K = 128$. Therefore, we select $K = 128$ as our final configuration to achieve the best balance between known and unknown object detection performance.

C.3. Visualization of EUS

To further validate the effectiveness of EUS in distinguishing between known and unknown objects, we present an extended per-class subspace score analysis beyond what was shown in the main paper. While the main paper (Fig. 3a) showed results for a subset of representative classes, here we present a comprehensive analysis across all classes from Task 1 and Task 2 using a model trained only on Task 1.

S.Figure 2 illustrates the subspace scores for proposals matched to ground-truth objects across all 40 classes. The results demonstrate that the ETF-based separation mechanism generalizes well beyond the examples shown in the main paper. For all known classes learned in Task 1 (classes 1-20: aeroplane, bicycle, bird, ..., tvmonitor), proposals consistently exhibit higher known scores (green bars) than unknown scores (red bars), indicating strong affinity toward the known subspace. Conversely, for all Task 2 classes not

yet encountered during training (classes 21-40: truck, traffic light, ..., refrigerator), proposals show higher unknown scores compared to known scores. The consistency of this separation across such a diverse set of object categories demonstrates that our ETF-based geometric modeling creates a robust decision boundary between known and unknown objects.

Additionally, we present t-SNE visualizations of proposals matched to actual ground-truth objects at inference time (Task 1) in S.Figure 3. Without EUS (left), known classes, unknown objects, and background proposals are entangled in the feature space, making it difficult for the detector to distinguish between them. With EUS (right), the three groups are clearly separated: known class proposals form distinct clusters, unknown proposals are well-separated from known classes, and background proposals occupy a separate region. This confirms that EUS enables clean 3-way separation between known, unknown, and background objects.

D. Computational Cost Analysis

S.Table 7. Per-image computational cost comparison between OrthogonalDet and DEUS. Measurements are averaged over 1,000 images with batch size 1 on a single NVIDIA RTX 4090 GPU.

Method	Inference Time (sec)	FLOPs (Tera)	Training Time (sec)	Training Memory (MB)
OrthogonalDet	0.261	1.031	0.130	2,934
DEUS	0.266	1.036	0.138	2,966
Overhead	+1.9%	+0.5%	+6.2%	+1.1%

We analyze the computational overhead introduced by DEUS compared to the baseline OrthogonalDet model. DEUS extends the baseline by incorporating two additional components: ETF-Subspace Unknown Separation (EUS) and Energy-based Known Distinction (EKD) loss. During training, these additions incur extra computational costs for ETF loss and EKD loss computation. During inference, additional memory and computation are required for projecting features onto the ETF subspaces and computing the unknown logit term. However, these computational overheads are minimal. The memory overhead is limited to storing the ETF projection results, which is proportional to the number of ETF basis vectors ($K = 128$ in our implementation). The inference time overhead consists solely of the matrix multiplication for ETF projection and the subsequent energy score calculation, both of which are lightweight operations.

S.Table 7 presents detailed measurements comparing OrthogonalDet and DEUS, where all measurements represent the average per image across 1,000 images. Note that both methods employ diffusion-style iterative refinement, which involves multiple denoising steps during inference, explaining why inference time is comparable to or even exceeds training time per image. The results show that DEUS introduces only modest computational overhead: inference time increases by 1.9% (from 0.261 to 0.266 seconds per image), FLOPs increase by 0.5% (from 1.031T to 1.036T per image), training time increases by 6.2%. These marginal increases demonstrate that DEUS achieves significant performance improvements in unknown object detection with negligible additional cost, making it highly practical for real-world deployment.

E. Qualitative Results

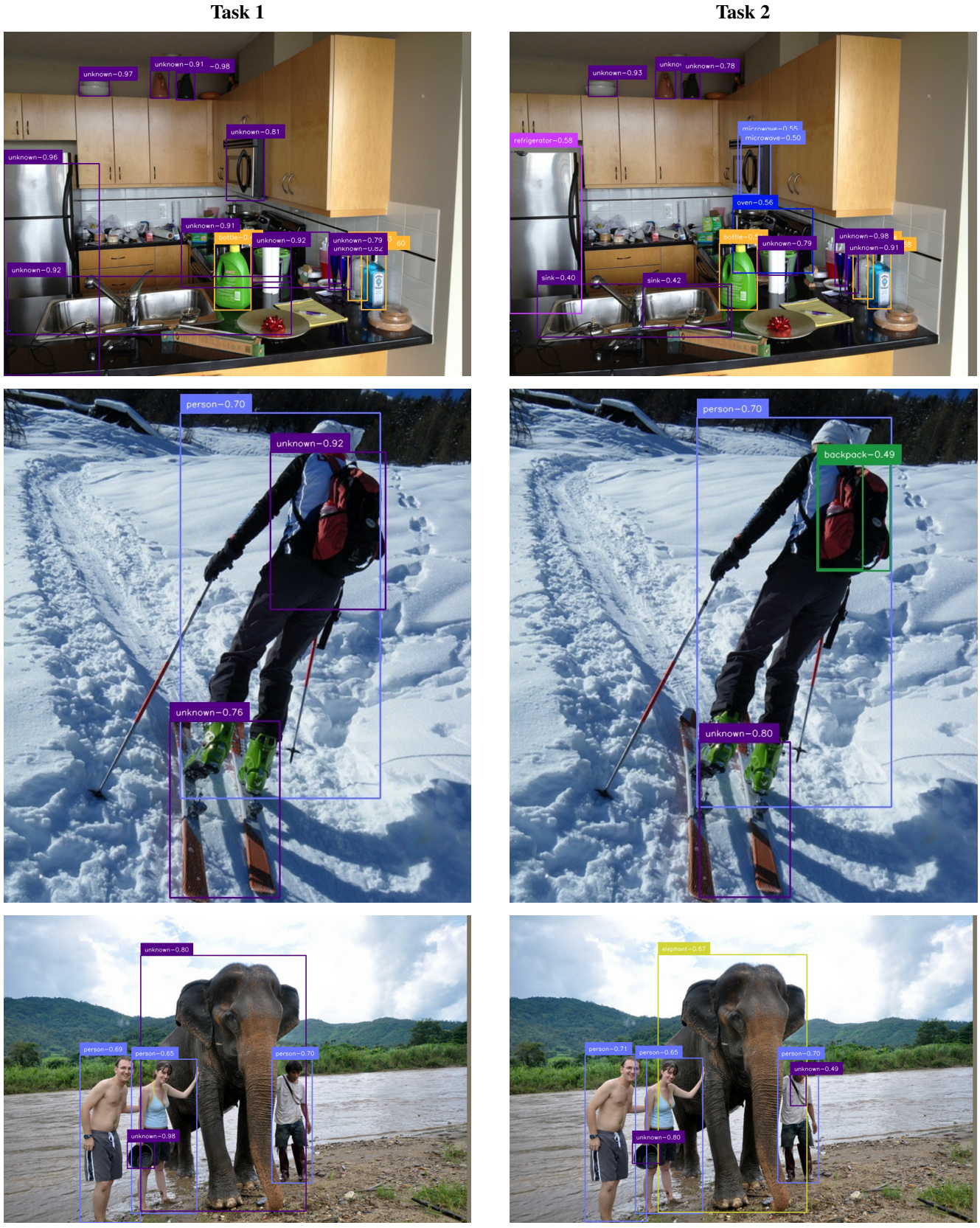
We present qualitative results of DEUS across incremental tasks to visualize how the model adapts as new classes are introduced. S.Figure 4 shows detection results from Task 1 (left) and Task 2 (right) on three different scenes. Purple boxes indicate unknown predictions, while other colored boxes indicate known class predictions.

Indoor. In a cluttered kitchen scene with numerous objects, DEUS correctly detects appliances (microwave, refrigerator, sink, oven) as unknown in Task 1. After learning Task 2 classes, all of these are successfully recognized as known, while objects from later tasks such as wine glasses and pottery remain correctly classified as unknown.

Outdoor. Objects within the same activity context may belong to different tasks. In Task 1, the person is detected as known while the backpack and skis are both identified as unknown. In Task 2, the backpack is selectively recognized as known, while skis (belonging to a later task) remain correctly detected as unknown.

Wildlife. As previously unknown objects transition to known, new unknowns can simultaneously emerge. In Task 1, persons are detected as known while the elephant is identified as unknown. After Task 2 training, the elephant is correctly recognized as known. Meanwhile, a basket remains unknown across both tasks, and a crossbag is newly detected as unknown in Task 2.

These results demonstrate that DEUS effectively maintains unknown detection across incremental tasks: previously unknown objects are correctly reclassified as known when their classes are learned, while objects from future tasks remain detected as unknown.



S.Figure 4. Qualitative results of DEUS across Task 1 (left) and Task 2 (right). Purple boxes indicate unknown predictions, while other colored boxes indicate known class predictions. As new classes are learned in Task 2, previously unknown objects are correctly reclassified as known, while objects from future tasks remain detected as unknown.