Supplementary Material
OW-Rep: Open World Object Detection with Instance Representation Learning

Sunoh Lee[1]* Minsik Jeon[2]* Jihong Min[3] Junwon Seo[2]
[1]KAIST [2]Carnegie Mellon University [3]Agency for Defense Development
sunoh0131@kaist.ac.kr {minsikj, junwonse}@andrew.cmu.edu happymin77@gmail.com

Contents

A. Problem Setting: OWOD vs OVOD

To clarify the scope of our work, we distinguish between the problem settings of Open Vocabulary Object Detection (OVOD) and Open-World Object Detection (OWOD), and highlight their complementary nature. OVOD focuses on detecting objects in an image based on given text prompts [1–5, 7, 9–12]. These methods typically leverage pretrained language models to fuse vision and language modalities, grounding the detector through explicit textual supervision. In contrast, OWOD, including our proposed approach, operates without any external input beyond the image itself. The goal is to detect both known objects (seen during training) and unknown objects (unseen categories), which can then be relabeled by human annotators and incrementally incorporated into the model to expand its knowledge.

While OVOD naturally learns semantics of each object during grounding, previous OWOD methods focus on detecting unknown objects, without explicitly learning their semantics. Some OVOD models like Grounding DINO [5] and GLIP [4] are trained on vast, diverse detection datasets far exceeding COCO, with the primary objective of enabling zero-shot detection through text prompts. Consequently, their framework does not inherently address scenarios that lack text prompts or require incremental learn-

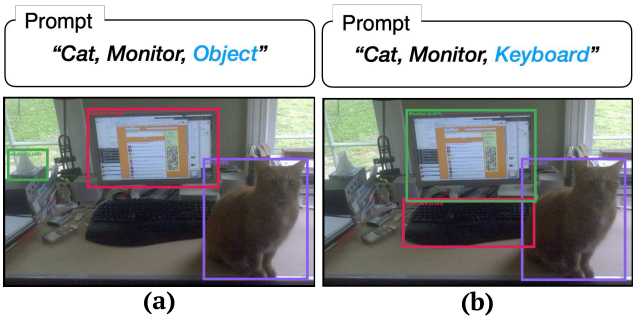

Figure S1. **Limitations of OVOD with text prompts.** Grounding DINO 1.6 [7] is utilized for the experiment. (a) When the prompt is given as *Cat, Monitor, Object*, the model detects the cat and the monitor, but fails to recognize the keyboard. (b) When the prompt includes *Keyboard*, the model successfully detects it. This demonstrates that OVOD methods are heavily dependent on explicit text prompts, and objects not included in the prompt cannot be detected.

*Equal contribution. Work primarily conducted while at Agency for Defense Development.

Project website: https://sunohlee.github.io/OW-Rep/

Table S1. **Details for OWDETR split and Unknown-Unknown split.** Unknown-Unknown split is a subset of the OWDETR split and shares the same Semantic split.

Task IDs→	**Task 1**	**Task 2**	**Task 3**	**Task 4**
Semantic Split→	Animals, Person, Vehicles	Appliances, Accessories, Outdoor, Furniture	Sports, Food	Electronic, Indoor, Kitchen
OWDETR split Training images	89490	55870	39402	38903
Unknown-Unknown split Training images	26296	23509	28558	38903

ing, which are central to the OWOD paradigm.

Our setting assumes that the set of objects to be detected is not known in advance and that corresponding text prompts are unavailable. This reflects a realistic open-world scenario where detectors inevitably encounter unexpected objects. In such cases, requiring text prompts for every possible object is impractical. Consequently, the fundamental difference between OWOD and OVOD lies in whether object detection depends on pre-specified text prompts.

Due to this difference in formulation, we do not directly compare our method with OVOD approaches. OVOD is suitable for cases where explicit prompts are provided, whereas OWOD addresses scenarios where such prompts cannot be assumed. For example, as shown in Fig S1, Grounding DINO can detect a keyboard only when given the corresponding prompt. In deployment contexts such as autonomous driving, detectors must recognize previously unseen objects without relying on textual input—conditions that naturally fall under the OWOD setting rather than OVOD.

Moreover, a central objective of OWOD is not only to detect unknown objects but also to distinguish them from known classes and enable incremental learning, where newly identified categories can be incorporated into the model over time. These differences emphasize that OWOD and OVOD are complementary but fundamentally distinct research directions, rather than directly comparable approaches.

B. Additional Experiments Material

B.1. Proposed Data Split

During real world deployment, a detector may encounter *unknown-unknown* objects that were neither labeled nor present in the training data. Therefore, an open-world object detector must be capable of detecting and obtaining instance embeddings for never-encountered classes. To facilitate this evaluation, we introduce a novel data split for open-world object detection named the Unknown-Unknown split.

The *Unknown-Unknown split* is built upon the OWDETR split, which was designed to minimize the presence of visually similar objects within the same data split. While the OWDETR split includes unknown class objects in the training set without labeling them, our proposed split goes a step further by entirely removing images containing unknown class objects from the training data. The details presented in Table S1. To illustrate, consider the Task 1 setting of the OWDETR split, where the model is trained on known classes such as animals, persons, and vehicles. In this setup, the training set often includes images where known objects co-occur with instances of unknown, future-task classes—for example, a person in a kitchen with a donut or a sink. The standard protocol discards only the annotations for donut and sink but still uses the image for training. In contrast, our *Unknown-Unknown split* excludes the entire image from the training data, thereby preventing the model from being visually exposed to any unknown objects. Consequently, the model has no prior exposure to these *unknown-unknown* objects. This data split enables a more rigorous evaluation of whether an open-world object detector can effectively identify truly unknown objects and capture their inter-proposal relationships by learning semantically rich feature representations.

Figure S2. **A failure case of the SAM-based box refinement. Left:** The initial bounding boxes predicted by the detector. **Right:** The refined boxes generated from SAM's masks. Boxes of the same color correspond to the proposals before and after refinement. The green box highlights a significant discrepancy that can occur without IoU thresholding, underscoring its importance.

B.2. Motivation for Evaluation Metrics

To evaluate the feature embedding quality of open-world detectors, existing methods typically cluster detected unknown objects into a predefined number of unknown classes, assign ground-truth labels to each cluster, and assess performance using metrics such as mAP. However, this approach assumes a fixed number of unknown classes and clusters all detected objects accordingly. In practice, not all objects in a dataset are labeled, and open-world object detectors may identify unlabeled objects, making clustering unreliable. Furthermore, numerous inaccurate detections

Table S2. **Ablation results for base Open World Object Detector.** U-Recall denotes the unknown detection recall. Incorporating our module enhances both instance embedding quality and unknown detection performance.

Task IDs→	**Task 1**		
Metrics→	U-Recall	Recall@1	Known mAP
CAT [6]	9.06	3.07	59.33
*CAT+**Ours***	**11.67**	**4.81**	**59.75**
OrthoDet [8]	21.56	11.84	**60.83**
*OrthoDet+**Ours***	**29.74**	**13.35**	59.41

Table S3. **Results using SAM and DINOv2 features for embedding transfer.** U-Recall denotes the unknown detection recall. Distilling DINOv2 features further improves feature quality.

Metrics→	U-Recall	Recall@1	Known mAP
Ours (w. SAM)	**30.63**	10.38	**59.38**
Ours (w. DINOv2)	30.56	**11.69**	58.89

Table S4. **Ablation results for the distillation method.** U-Recall denotes the unknown detection recall, and L1 Distill denotes the direct distillation of DINOv2 features via L1 loss. Our method outperforms the L1 Distill method and shows superior feature representation quality.

Metrics→	U-Recall	Recall@1	Known mAP
PROB	18.84	4.42	58.98
PROB + L1 Distill	31.19	8.69	59.20
Ours	**31.54**	**11.40**	**59.12**

further degrade clustering quality, rendering the evaluation metric unreliable.

To this end, we replace discrete clustering-based evaluation with a direct assessment of feature space quality using Recall@K. By employing Recall@K as an evaluation metric for feature learning, we can evaluate whether instances of the same unknown class are embedded closely without relying on a predefined number of unknown classes.

B.3. Additional Implementation Details

During the initial training phase, only L_o, L_c, and L_b are optimized. The *Embedding Transfer Module* and *Unknown Box Refine Module* are introduced after 14 and 35 epochs, respectively. For the *Unknown Box Refine Module*, four regular grid points are sampled from each input unknown proposal, and used as prompts to generate a instance mask. The refined unknowns are defined as the smallest bounding box containing this mask. Bounding boxes are used for training only if their IoU before and after refinement is 0.5 or higher, minimizing performance degradation on known classes. This ensures the detector from training with wrong supervisions, such as depicted in Fig S2

C. Ablation and Component Analysis

C.1. Additional Ablation for Baseline Detector.

Our approach is designed as modular components that can be integrated into any baseline open-world object detector. We primarily adopt *PROB* as the baseline detector to highlight the effectiveness of our method in improving both unknown object detection and instance-level feature representation. Additionally, we validate its general applicability by conducting experiments with multiple OWOD baselines Table 4 and Table S2.

C.2. Comparison with using SAM and DINOv2 features for ETM

Table S3 presents results of distilling feature similarity from SAM and DINOv2. Although embedding transfer can use any source model, we use DINOv2 for its superior feature learning performance.

C.3. Ablation Studies of Distillation Method

To evaluate the effectiveness of the *Embedding Transfer Module*, we compare our method with other distillation approaches using VFM embeddings. Specifically, we consider a baseline that directly aligns the detector's instance embeddings with source embeddings from DINOv2 using L1 loss (L1 Distill). Since the embedding dimension D of DINOv2 differs from the detector's instance embedding dimension d, we adjust the output dimension of the MLP layer for instance embedding accordingly. Experiments are conducted on a detector trained with Task 1 of the OWOD split.

The results are shown in Table S4. Our model demonstrates superior embedding quality compared to the L1 loss method, which simply forces detector features to match VFM features. This suggests that directly mimicking the high-dimensional feature space of VFMs is suboptimal. By using feature similarity as a weight in contrastive loss during distillation, our method learns a semantically rich feature space.

C.4. Ablation Studies of Module under Unknown-Unknown split

We conduct ablation studies under the *Unknown-Unknown split* to validate the effectiveness of each component. The results are summarized in Table S5. Consistent with the findings from the OWOD split, integrating the *Embedding Transfer Module* improves the quality of instance embeddings by distilling the rich semantic information of VFMs, while the *Unknown Box Refine Module* enhances

Table S5. **Ablation results for each component in Unknown-Unknown split.** Incorporating the *Embedding Transfer Module* (**ETM**) and *Unknown Box Refine Module* (**URM**) enhances the instance embedding quality and unknown detection performance, respectively.

Task IDs →	**Task 1**			**Task 2**			**Task 3**			**Task 4**
Metrics →	Unknown Recall	Recall @1	Known mAP	Unknown Recall	Recall @1	Known mAP	Unknown Recall	Recall @1	Known mAP	Known mAP
PROB	21.04	3.32	67.65	29.85	7.81	**44.80**	33.37	12.62	**39.65**	38.70
PROB+URM	36.71	5.14	66.77	44.80	9.85	43.71	44.84	15.29	38.16	38.27
PROB+ETM	21.83	5.49	**67.68**	31.80	11.15	43.78	33.61	13.96	39.58	**38.73**
Ours	**37.27**	**11.21**	66.82	**45.93**	**17.99**	43.42	**45.81**	**20.65**	38.57	38.12

Table S6. **Ablation results for each hyperparameters of ETM and URM on OWOD split.** The first row shows our default hyperparameter results. In subsequent rows, we vary a single hyperparameter, highlighted in green.

Task IDs→				**Task 1**		
ETM		**URM**		Unknown	Recall	Known
δ	σ	k	κ	Recall	@1	mAP
1	1	10	0.5	30.56	11.69	58.89
1	1	10	0.45	31.42	**11.98**	58.38
1	1	10	0.55	30.11	11.47	59.12
1	1	5	0.5	28.51	10.58	**59.31**
1	1	15	0.5	**32.01**	11.89	58.80
1	0.5	10	0.5	31.55	10.69	59.02
1	2.0	10	0.5	31.52	11.59	58.95
0.9	1	10	0.5	30.70	11.20	58.89
1.1	1	10	0.5	31.68	11.93	58.79

unknown object detection performance. When both modules are used together, feature learning is applied to refined proposals, leading to a significant improvement in feature embedding quality.

C.5. Ablation Studies of each Module's Hyperparameters

We performed a comprehensive hyperparameter ablation study to validate the robustness of the ETM and URM modules within the OWOD split. The experimental results are detailed in Table S6. Our model demonstrates insensitivity to variations in each hyperparameter. Furthermore, we observed that varying parameters like k and κ maintained performance without compromising the known mAP of the PROB baseline.

D. Additional Quantitative Results

D.1. Quantitative results on OWDETR split

Table S7 demonstrates that our method outperforms other models in both unknown object detection and feature learning on OWDETR split. This trend is consistent with the results on the Unknown-Unknown split shown in Table 2,3. The Unknown-Unknown split is a subset of the OWDETR split as described in B.1, which explains the similarity in performance trends.

Table S7. **Results on OWDETR split.** U-Recall denotes the unknown detection recall. Our method outperforms others in unknown detection and feature quality.

Task IDs→	**Task 1**		
Metrics→	U-Recall	Recall@1	Known mAP
PROB	16.77	2.27	**73.09**
OSODD	16.77	2.22	**73.09**
UC-OWOD	10.62	1.27	22.84
RNCDL	17.09	1.52	72.34
PROB+URM	**29.68**	3.59	71.21
PROB+ETM	15.31	3.45	72.8
Ours	29.51	**9.05**	71.90

D.2. Quantitative Results of Efficiency Comparisons including Direct VFM usage

Table S8 reports runtime and memory costs, including a comparison with *Ours*+DINO, which pools DINOv2 features from proposals at inference. Our method introduces minimal overhead relative to *PROB*, requiring only a few additional MLP layers for instance feature extraction, and thus achieves comparable inference speed. In contrast, *Ours*+DINO Pool increases computation. The number of trainable parameters grows by only 0.5%, since SAM and DINO features are extracted once per image and reused during training.

Table S8. **Efficiency comparisons including direct VFM usage.** Our method enables effective feature learning with minimal computational overhead, compared to directly using VFM.

Metrics ↓	*PROB*	*Ours*	*Ours*+DINO Pool
Inference Time (msec)	47.1	47.2	80.9
Inference # Params	39.7M	39.9M	344M
Peak Inference Memory (MiB)	3030	3036	3036
Train Time (relative)	x1.00	x1.74	x1.74
Peak Train Memory (MiB)	80660	80700	80700
Recall@1	4.42	11.69	15.53

Table S9. **Class similarity analysis on animal and vehicle superclasses.** Our method embeds unknown objects closer to their semantically similar known classes, preserving meaningful relationships in the feature space.

Superclass→	**Animal**		**Vehicle**	
Metrics→	Unknown Recall	Recall@1	Unknown Recall	Recall@1
PROB	87.80	43.30	60.97	33.12
OSODD	87.80	19.14	60.97	13.50
UC-OWOD	83.61	45.22	59.07	41.35
RNCDL	87.68	8.13	59.49	4.43
Ours	**89.00**	**59.33**	**65.82**	**47.68**

We distill knowledge from VFMs rather than use them directly for two reasons: (1) VFMs are not designed for open-world detection, which requires identifying and incrementally learning unknown objects, and (2) direct use at inference is computationally costly.

D.3. Class Similarity Analysis

We evaluate whether unknown objects are closely embedded to their semantically similar known classes in the feature space. Specifically, we compute Recall@1 for unknown objects, which measures the proportion of unknown objects whose nearest known class shares the same superclass. The nearest known class is identified by comparing the unknown object's embedding to the centroids of known classes, where each centroid is the average embedding of all objects classified as that class. The results presented in Table S9 demonstrate that our method outperforms both the baseline and other self-supervised learning approaches. By leveraging the rich feature space of VFMs, our model effectively captures the relationships between known and unknown objects, embedding unknown objects closer to their semantically similar known counterparts.

D.4. Quantitative results of Unknown Confusion Metrics

Our method slightly increases confusion metrics, possibly due to the refinement of misclassified unknown boxes. The results are shown in Table S10. Moreover, learning nuanced instance relationships rather than enforcing strict class boundaries may weaken the separation between known and unknown classes.

Table S10. **Results for unknown confusion metrics.** WI denotes Wilderness Impact, and A-OSE denotes Absolute Open-Set Error.

Task IDs→	**Task 1**		**Task 2**		**Task 3**	
Metrics→	WI	A-OSE	WI	A-OSE	WI	A-OSE
PROB	0.0550	4956	0.0317	5596	0.0165	3122
PROB+URM	0.0642	6821	0.0288	4399	0.0197	4360
PROB+ETM	0.0554	5065	0.0277	3667	0.0169	2808
Ours	0.0621	6056	0.0359	7739	0.0232	5099

E. Additional Qualitative Results

E.1. Qualitative Results of Inter-proposal Relationships.

Fig. S4 illustrates the results of inter-proposal relationships on the OWOD split. The embedding of *giraffe* is used as reference. Our method considers other giraffe as highly similar, sheep as moderately similar, and fire hydrant as dissimilar, which reflects their semantic relationship. This demonstrates that our method can simultaneously detect unknown object and extract semantically rich feature from objects. *PROB*, on the other hand, considers all the proposals as highly similar since they are not trained to distinguish different unknown objects, but to learn a generalizable concept of object. Both *UC-OWOD* and *RNCDL* which learns instance embedding with self-supervision struggles to learn semantic relationship between objects. For more comprehensive comparison of the feature space, refer to Fig. S3.

Fig. S5 presents the results of inter-proposal relationships on the *Unknown-Unknown split*. Using the embedding of unknown *clock* as reference, our method successfully captures the semantic relationship between objects even though they never observed *clock* during training. For example, that other *clock*s are highly similar, while *person* or *tennis racker* are dissimilar. Other methods suffers from learning such relationships correctly. This demonstrates that by distilling the rich feature through relaxed contrastive learning, our method learns a generalizable feature space of instance embeddings.

E.2. Qualitative Results of Open-World Tracking.

We present additional results from our open-world tracking experiment, conducted using the detector trained on Task 1 of the OWOD split. The results are shown in Fig. S6. Despite continuous shape changes in the *deer* and *rabbit*, the tracker built on our proposals and features successfully tracks the object. In contrast, when using *PROB* outputs, the tracker struggles to assign proposals correctly as object shapes change. Additionally, *PROB* fails to detect the un-

known *ball*. These results demonstrate that our method effectively extracts semantically rich features and detects unknown objects, improving open-world object tracking.

References

[1] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2021. 1

[2] Weicheng Kuo, Yin Cui, Xiuye Gu, AJ Piergiovanni, and Anelia Angelova. F-vlm: Open-vocabulary object detection upon frozen vision and language models. *International Conference on Learning Representations (ICLR)*, 2023.

[3] Liangqi Li, Jiaxu Miao, Dahu Shi, Wenming Tan, Ye Ren, Yi Yang, and Shiliang Pu. Distilling detr with visual-linguistic knowledge for open-vocabulary object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6501–6510, 2023.

[4] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10965–10975, 2022. 1

[5] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 38–55. Springer, 2024. 1

[6] Shuailei Ma, Yuefeng Wang, Ying Wei, Jiaqi Fan, Thomas H Li, Hongli Liu, and Fanbing Lv. Cat: Localization and identification cascade detection transformer for open-world object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19681–19690, 2023. 3

[7] Tianhe Ren, Qing Jiang, Shilong Liu, Zhaoyang Zeng, Wenlong Liu, Han Gao, Hongjie Huang, Zhengyu Ma, Xiaoke Jiang, Yihao Chen, et al. Grounding dino 1.5: Advance the" edge" of open-set object detection. *arXiv preprint arXiv:2405.10300*, 2024. 1

[8] Zhicheng Sun, Jinghan Li, and Yadong Mu. Exploring orthogonality in open world object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17302–17312, 2024. 3

[9] Jiong Wang, Huiming Zhang, Haiwen Hong, Xuan Jin, Yuan He, Hui Xue, and Zhou Zhao. Open-vocabulary object detection with an open corpus. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6759–6769, 2023. 1

[10] Tao Wang. Learning to detect and segment for open vocabulary object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7051–7060, 2023.

[11] Size Wu, Wenwei Zhang, Sheng Jin, Wentao Liu, and Chen Change Loy. Aligning bag of regions for open-vocabulary object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15254–15264, 2023.

[12] Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy. Open-vocabulary detr with conditional matching. In *European conference on computer vision*, pages 106–122. Springer, 2022. 1

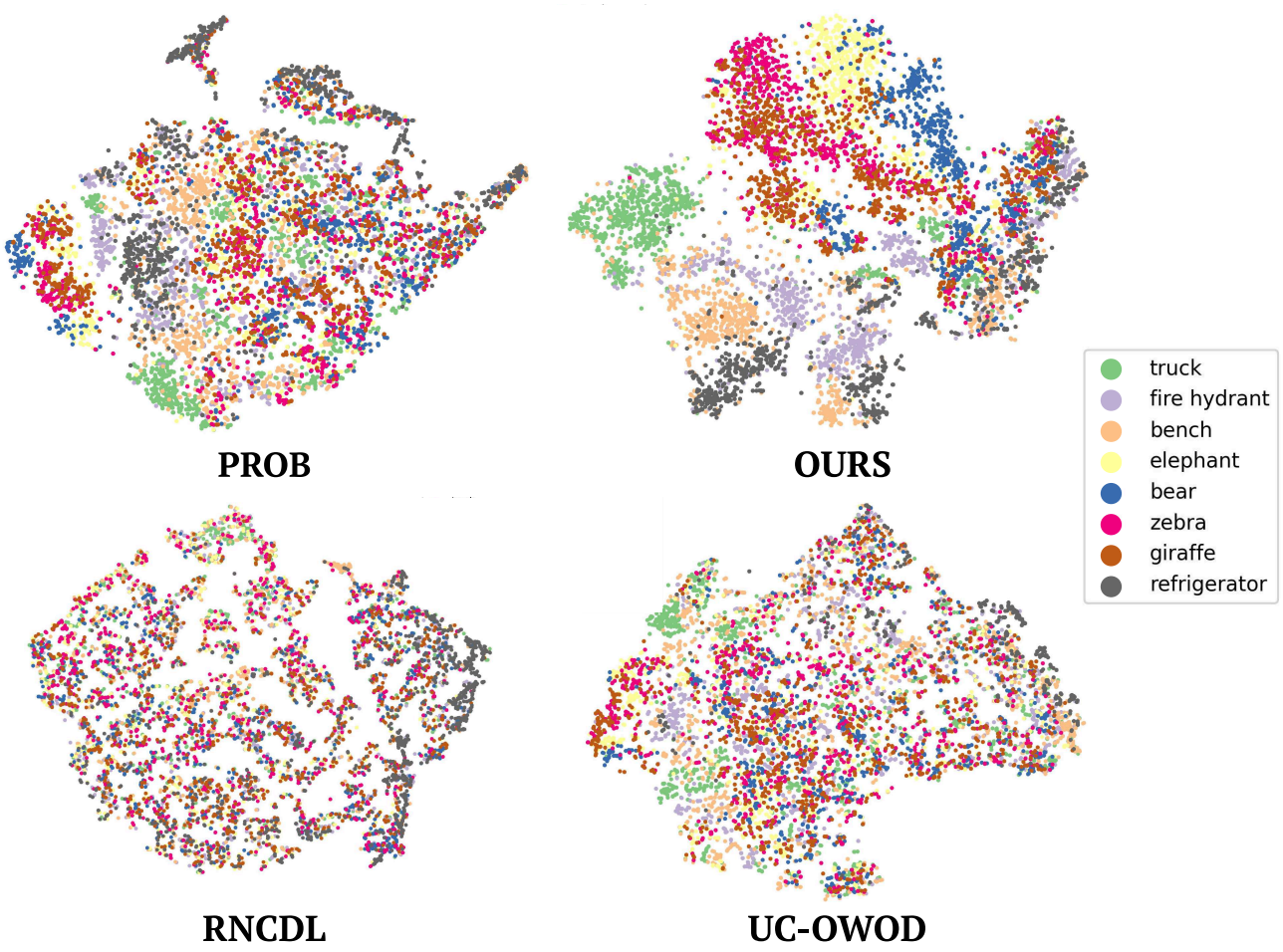

Figure S3. **Additional Qualitative Results on Inter-proposal Relationships.** We visualize the inter-proposal similarity of eight unknown classes. Our model groups proposals of the same class more compactly and allows similar animal classes to share a close feature space. In contrast, *RNCDL* and *UC-OWOD* exhibit a less effective feature space than *PROB*.

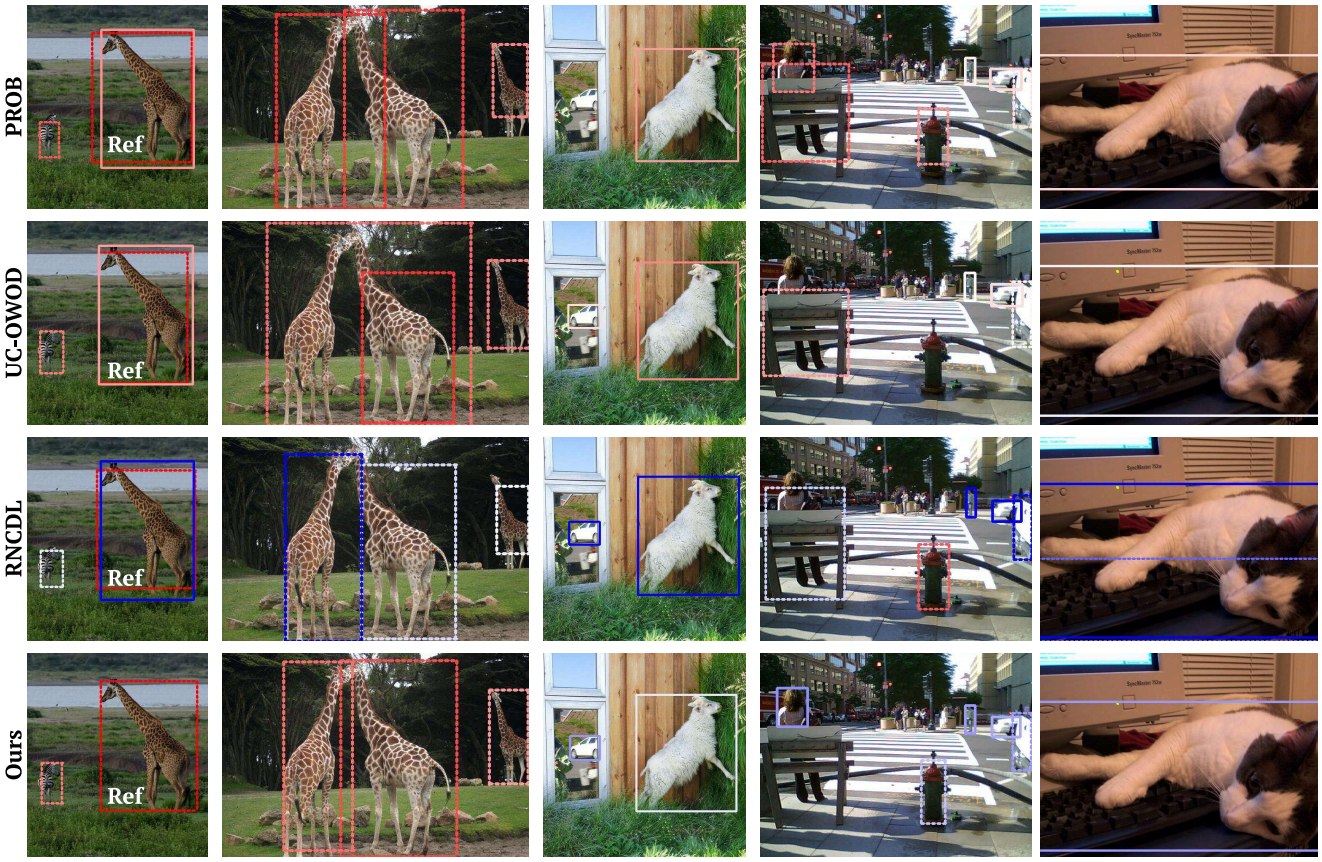

Figure S4. **Additional Qualitative Results on OWOD split.** We visualize the detection results of our model along with the instance embedding similarity with reference proposal. Red indicates high similarity, while blue represents high dissimilarity. Our method accurately detects both the known and unknown objects while simultaneously capturing fine-grained semantic relationships between proposals. For example, a *giraffe* is considered more similar to a *sheep* than to a *fire hydrant*. In contrast, other approaches fail to capture such detailed semantic relationships.

Figure S5. **Additional qualitative results on Unknown-Unknown split.** We visualize the detection results of our model along with the instance embedding similarity with the reference proposal of the *clock*. Red indicates high similarity, while blue represents high dissimilarity. Even for the *unknown-unknown*s that were not observed during training, our method accurately detects both the known and unknown objects while simultaneously capturing fine-grained semantic relationships between proposals. For example, a *clock* is considered more similar to other *clock* than to a *tennis racket*. In contrast, other approaches fail to capture such detailed semantic relationships.

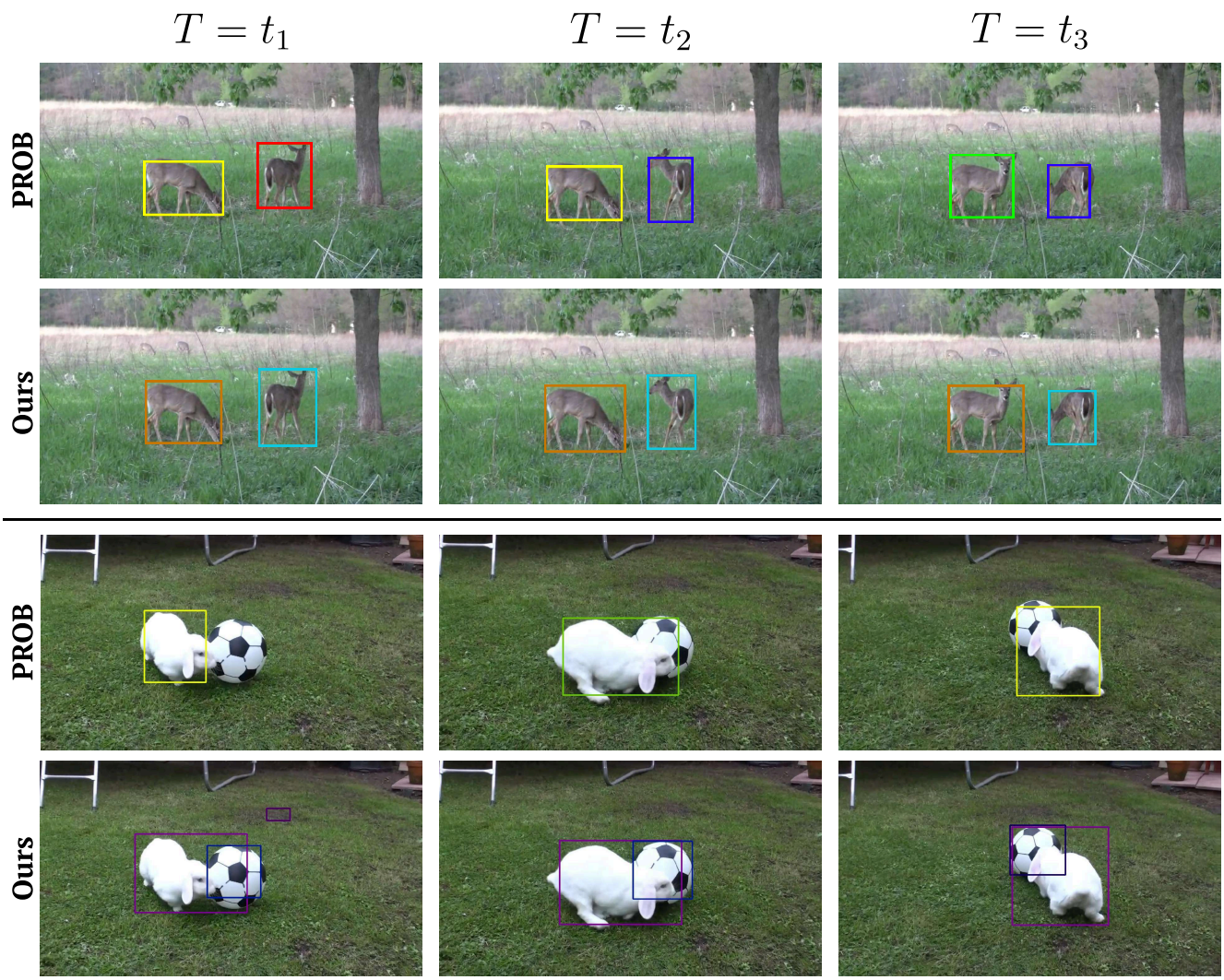

Figure S6. **Additional Qualitative Results of Open-World Object Tracking.** Bounding box color represents track IDs, with all the *deer*, *rabbit* and *balls* belonging to unknown classes. Our method successfully tracks the unknown objects by learning semantically rich instance embeddings, used to compute feature similarity between inter-frame proposals.

Algorithm 1 Embedding Transfer Module (ETM) within a Training Iteration

Input: Image batch $\{I\}$, Ground truth $\{y\}$, Model M, DINOv2 model M_{DINO}
Hyperparameters: Margin δ, Kernel bandwidth σ

$\{B_{pred}\}, \{Q_{emb}\} \leftarrow M(\{I\})$ ▷ Get all predictions and query embeddings

$\{b_k\} \leftarrow$ HungarianMatcher$(\{B_{pred}\}, \{y\})$ ▷ Get known proposals
$\{b_u\} \leftarrow$ Top-k ObjectnesswithRefinement$(\{B_{pred}\})$ ▷ Get top-k unknown proposals from URM
$B \leftarrow \{b_k\} \cup \{b_u\}$
$Q_B \leftarrow$ GetCorrespEmbeddings(Q_{emb}, B)

Step 1: Get Source and Instance Embeddings
$Z \leftarrow$ MLP(Q_B)
$F \leftarrow M_{DINO}(\{I\})$ ▷ Extract DINOv2 feature map for the image batch. (A one-time pre-processing step.)
$S \leftarrow$ AvgPool(F, B) ▷ Average Pool DINO embeddings from each proposals to obtain source embeddings

Step 2: Compute Pairwise Source Similarity (In practice, implemented using vectorized operations for efficiency.)
$N = |S|$
for $i = 1$ to N, $j = 1$ to N **do**
$w_{ij} \leftarrow \exp\left(-\frac{\|s_i - s_j\|_2^2}{\sigma}\right)$ ▷ σ : Gaussian kernel, w_{ij} : Pairwise Source Similarity
end for

Step 3: Compute Pairwise Instance Distance
for $i = 1$ to N, $j = 1$ to N **do**
$d_{ij} \leftarrow \|z_i - z_j\|_2$ ▷ d_{ij}: Pairwise Euclidean distance
end for

Step 4: Compute Relaxed Contrastive Loss
$\mathcal{L}_{et} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} \left[\mathbf{w}_{ij}\mathbf{d}_{ij}^2 + (1 - \mathbf{w}_{ij}) \left[\delta - \mathbf{d}_{ij}\right]_+^2\right]$ ▷ δ : margin

Return $\mathcal{L}_{et}$