Training-Free Open World Object Detection

Jun-Woo Heo
Korea University
dinleo11@korea.ac.kr

Chang-Sik Woo
Korea University
jasoncswoo@gmail.com

Gyeong-Moon Park*
Korea University
gm-park@korea.ac.kr

Abstract

Open World Object Detection (OWOD) requires a detector to recognize known categories, detect unknown objects, and incrementally learn new classes. Recent work such as OW-OVD adapts a pre-trained Vision-Language Object Detector (VLOD) to OWOD through fine-tuning. However, this approach demands costly backpropagation and often degrades the detector's own zero-shot performance on known categories, calling into question whether training is necessary. In this paper, we propose **CAVE** (**C**lustering in **A**ttribute-**V**isual **E**mbedding space), a training-free framework that adapts a VLOD to OWOD without any parameter updates. From a single forward pass, CAVE collects visual and attribute statistics for each class via von Mises-Fisher-based clustering, along with attribute co-occurrence patterns, without any gradient computation. At inference, for known-class detection, CAVE fuses the original VLOD's text-based scores with the **MAS** (**M**ixture-**A**ggregate **S**core) derived from the collected statistics, while for unknown discovery, it uses the **AOS** (**A**ttribute **O**bjectness **S**core), which combines superclass responses with attribute co-occurrence patterns. Experiments on M-OWODB show that CAVE outperforms OW-OVD (+16.2 known-class mAP at Task 4 and +28.0 average unknown recall), while surpassing the zero-shot VLOD across all incremental tasks (+1.3 known-class mAP at Task 4).

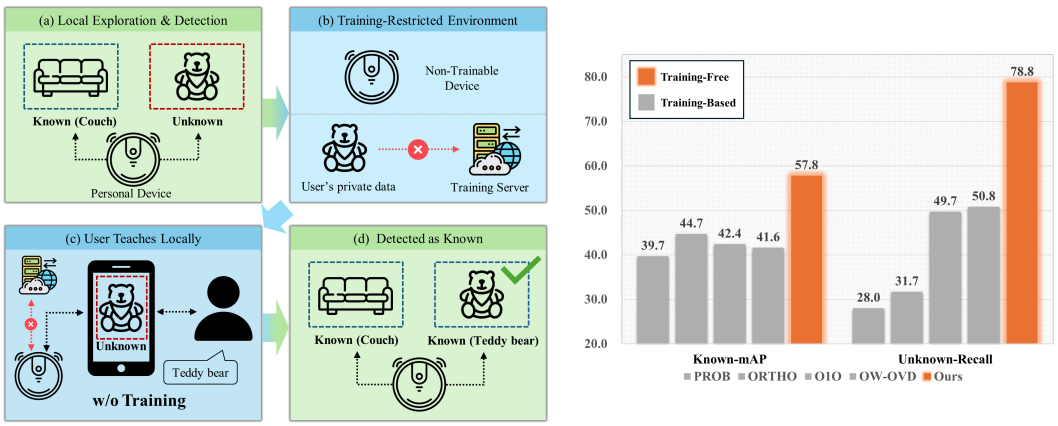
1 Introduction

Traditional object detectors assume a *closed set* of categories defined at training time, fundamentally at odds with real-world deployment where novel objects emerge continuously. Open World Object Detection (OWOD) [11] addresses this by requiring detectors to recognize known classes, proactively identify *unknown* objects, and incrementally learn newly labeled categories without forgetting [8, 41, 20, 33, 30, 26]. However, existing OWOD methods train from scratch, forgoing large-scale vision-language pre-training and lacking zero-shot generalization to unseen categories. Open Vocabulary Detection (OVD) [5, 19] takes a complementary approach, using Vision-Language Models (VLMs) such as CLIP [28] for zero-shot detection of arbitrary categories via natural language, yet OVD detectors cannot actively discover objects outside the given prompt vocabulary. OW-OVD [36] bridges this gap by adapting a pre-trained Vision-Language Object Detector (VLOD), YOLO-World [5], to the OWOD task. Treating OWOD as a downstream task of the VLOD, OW-OVD fine-tunes prompt embeddings while incorporating LLM-generated attributes for unknown prediction.

In practice, however, many real-world deployment scenarios make such fine-tuning infeasible. As illustrated in Fig. 1a, consider a robotic vacuum equipped with a pre-trained VLOD that explores a home environment, detecting known objects but inevitably encountering unknown ones. Edge devices such as FPGAs employ quantized, hard-coded parameters that preclude gradient computation [22], and privacy-sensitive applications prohibit uploading personal data to remote servers for retraining. Ideally, the user would simply provide labels, and the device would adapt locally without any gradient-based training to recognize both previously known and newly taught objects. This demand

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Real-world scenario for training-free OWOD. (b) Performance comparison on M-OWODB.

Figure 1: (a) A personal device detects known and unknown objects, but fine-tuning is infeasible due to device constraints and data privacy. When the user provides labels locally, previously unknown objects are recognized under the new labels without any training. Green and blue panels denote inference and adaptation stages, respectively. (b) Performance comparison with training-based OWOD methods on M-OWODB. Known-mAP denotes Task 4 mAP over all 80 classes; Unknown-Recall denotes the Task 1–3 average.

for *training-free* adaptation arises wherever on-device retraining is impractical, such as autonomous vehicles or mission-critical drones in disconnected environments.

OW-OVD falls short in these scenarios. Although it builds upon YOLO-World, a detector designed for efficient edge deployment, its reliance on backpropagation makes it difficult to apply in the same edge environments the base model targets. Moreover, tuning the VLOD to acquire open-world capability substantially degrades its known-class performance relative to the zero-shot baseline: on M-OWODB Task 1, YOLO-World achieves 74.7 mAP in zero-shot mode, whereas OW-OVD obtains only 69.4 mAP after prompt tuning — a gap of 5.3 points that widens in later tasks (Task 4: 56.5 *vs.* 41.6). This raises a fundamental question: *is training necessary to equip a VLOD with open-world detection capability, when the frozen model already possesses superior recognition capability?*

Prior training-free work has addressed such scenarios in classification [38, 22] and OOD detection [16], adapting to a task under the given supervision within a short wall-clock time and without any gradient updates, yet no prior method has tackled OWOD in this regime. To address this, we propose **CAVE** (**C**lustering in **A**ttribute-**V**isual **E**mbedding space), the first training-free framework for OWOD. In line with these works, CAVE adapts under the given supervision in a single forward pass, without backpropagation, parameter updates, or optimizer state. Although a pre-trained VLOD encodes powerful visual-semantic representations, its text embeddings alone may fail to capture the diverse visual and semantic characteristics specific to the target deployment. We supply this missing knowledge through two complementary signals. For known-class detection, we construct a unified visual-attribute embedding for each object and perform DP-vMF-means clustering [32, 2] to build multiple prototypes per class that capture intra-class visual and semantic diversity, complementing the VLOD's original text-based score via **MAS** (**M**ixture-**A**ggregate **S**core). For unknown detection, we collect attribute co-occurrence statistics that capture general object characteristics, and **AOS** (**A**ttribute **O**bjectness **S**core) leverages both superclass cues and these co-occurrence patterns to discover unknown objects. As shown in Fig. 1b, our approach surpasses all existing training-based OWOD methods by a large margin in both known-class mAP and unknown recall.

Our contributions can be summarized as follows:

- We formulate *training-free OWOD*, a novel scenario for real-world applications where a frozen VLOD is adapted to open-world detection without any parameter updates.
- We propose CAVE with MAS, an adaptation framework that captures intra-class diversity through multi-prototype clustering, improving known-class detection.

- We introduce AOS, which leverages the VLOD's inherent semantic representations to discover unknown objects more diversely and precisely than existing methods.
- Experiments on M-OWODB show that CAVE outperforms OW-OVD by +16.2 known mAP and +28.0 unknown recall, surpassing all existing training-based OWOD methods across all tasks.

2 Related Work

Open World and Open Vocabulary Detection. OWOD [11] requires detectors to recognize known objects, discover unknowns, and incrementally learn new categories. Subsequent works [8, 41, 20, 33, 10] have advanced this line of research, but all train detectors from scratch, forgoing the rich representations available in pre-trained VLMs, which have proven effective even for incremental learning [24]. Related efforts further detect unknown objects without access to source data [25]. OVD uses VLMs such as CLIP [28] for zero-shot detection via natural language [5, 19], yet cannot proactively discover objects outside the given vocabulary. FOMO [40] and OW-OVD [36] bridge these two settings by leveraging mid-level attributes (*e.g.*, color, material, shape) [12, 27] as auxiliary signals for unknown prediction, but still rely on gradient-based training that degrades zero-shot capability and on LLM-based attribute regeneration whenever new classes appear, limiting deployment on resource-constrained devices.

Training-Free Adaptation via Prototypes. Training-free methods adapt pre-trained models without gradients across classification [38, 22], OOD detection [16, 1], segmentation [13], and detection [18], typically via cached exemplars or closed-form operations on frozen weights—*e.g.*, Tip-Adapter's [38] key–value cache or SeTAR's [16] selective low-rank approximation—rather than gradient updates. These methods, however, target single-task known-class performance and do not address unknown discovery or incremental learning. A complementary line represents classes through prototypes: Prototypical Networks [31] use mean embeddings, but a single centroid cannot capture intra-class diversity; vMF mixtures [2] model directional data on the unit hypersphere, and DP-vMF-means [32] further determines the cluster count nonparametrically, while density-based and hierarchical methods [6, 3, 29] also determine the number of clusters automatically but must keep the raw points in memory. We unify these directions by adopting DP-vMF-means in a joint visual–attribute space, building multiple prototypes per class that support both known recognition and unknown discovery without gradients.

3 Clustering in Attribute-Visual Embedding Space

We propose **CAVE** (**C**lustering in **A**ttribute-**V**isual **E**mbedding space), a training-free framework that adapts a Vision-Language Object Detector (VLOD) to Open World Object Detection. CAVE operates in two phases without any parameter updates (Fig. 2): during a single forward pass, it collects per-class multi-prototype statistics and attribute co-occurrence patterns; at inference, these statistics enhance known-class detection and enable unknown object discovery.

3.1 Problem Formulation

OWOD [11] considers T sequential tasks. At each task t, a labeled set $\mathcal{D}_t = \{(\mathcal{I}_i, \mathcal{Y}_i)\}_{i=1}^{N}$ is available, where each annotation $(c_j, b_j) \in \mathcal{Y}_i$ consists of a class label $c_j \in \mathcal{K}_t$ and a bounding box $b_j = [x_j, y_j, w_j, h_j]$. The known class set $\mathcal{K}_t$ accumulates classes from all tasks up to and including task t, with $C_t = |\mathcal{K}_t|$ classes. The detector is expected to classify objects of all known classes in $\mathcal{K}_t$ while flagging instances whose class belongs to the open unknown set $\mathcal{U}_t = \{C_t + 1, \ldots\}$ as unknown. When task $t+1$ arrives, a subset $\mathcal{U}' \subset \mathcal{U}_t$ is revealed (i.e., labeled examples for them become available) and absorbed into the known set, $\mathcal{K}_{t+1} = \mathcal{K}_t \cup \mathcal{U}'$, while the unknowns shrink to $\mathcal{U}_{t+1} = \mathcal{U}_t \setminus \mathcal{U}'$. This cycle of detection and incremental labeling continues across all T tasks.

3.2 Model and Embedding Construction

Base Model. We adopt YOLO-World [5] as the base detector $\mathcal{M}_\theta$, which comprises a visual encoder $f_{\text{vis}}(\cdot;\theta)$, a text encoder $f_{\text{text}}(\cdot;\theta)$, and a bounding box head $f_{\text{bbox}}(\cdot;\theta)$. Given an input image $\mathcal{I}$, the visual encoder produces Q region proposal embeddings E_{vis} (indexed by q), while the text

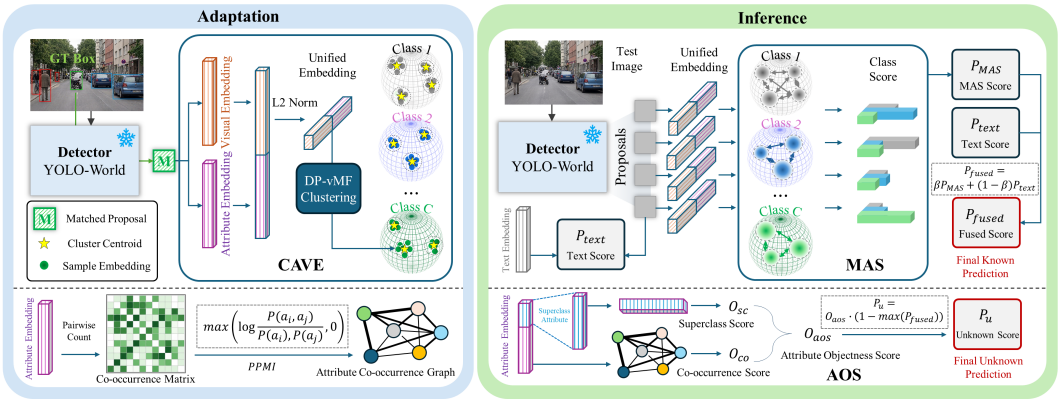


Figure 2: Overview of CAVE. **Left (Adaptation)**: In a single forward pass, GT-matched proposals are encoded into unified visual-attribute embeddings and clustered via DP-vMF-means to form multiple prototypes per class; attribute co-occurrences are accumulated into a co-occurrence graph via PPMI. **Right (Inference)**: At inference, MAS aggregates π/κ-weighted prototype similarities, fused with the text score for known-class detection (top), while AOS derives objectness from superclass and co-occurrence signals, modulated by fused confidence for unknown detection (bottom).

encoder maps $C_t = |\mathcal{K}_t|$ known class names into $E_{\text{text}}^{\text{cls}}$ in the same d-dimensional space:

$$E_{\text{vis}} = f_{\text{vis}}(\mathcal{I}) \in \mathbb{R}^{Q\times d}, \quad E_{\text{text}}^{\text{cls}} = f_{\text{text}}(\mathcal{K}_t) \in \mathbb{R}^{C_t\times d}. \tag{1}$$

Class logits and bounding box predictions are obtained from these embeddings:

$$z_{\text{cls}} = \gamma\,\text{BN}(E_{\text{vis}})\,(\|E_{\text{text}}^{\text{cls}}\|_2)^\top + b \in \mathbb{R}^{Q\times C_t}, \quad z_{\text{bbox}} = f_{\text{bbox}}(E_{\text{vis}}) \in \mathbb{R}^{Q\times 4}, \tag{2}$$

where $\|\cdot\|_2$ denotes row-wise ℓ_2 normalization, $\text{BN}(\cdot)$ is the batch normalization of the detection head, and γ and b are its learned logit scale and bias. The base detector outputs a set of detections $\hat{\mathcal{Y}} = \{(\hat{c}_q, \hat{s}_q, \hat{b}_q)\}_{q=1}^{Q}$ by selecting the highest-scoring class after sigmoid activation $\sigma(\cdot)$:

$$\hat{c}_q = \arg\max_c \sigma(z_{\text{cls}}[q,c]), \quad \hat{s}_q = \max_c \sigma(z_{\text{cls}}[q,c]), \quad \hat{b}_q = z_{\text{bbox}}[q]. \tag{3}$$

In our framework, we keep all parameters θ frozen, performing no gradient-based updates.

Attribute Embedding. We construct an attribute vocabulary $\mathcal{A}$ of $N_a = 512$ words, matching the visual embedding dimension $d = 512$ to balance the two modalities. We start from 325 visual properties (*e.g.*, color, material, shape, and texture) sourced from VAW [27], after removing duplicates and entries that overlap with class names. We then add 23 category-level superclass attributes (*e.g.*, "animal", "food", and "furniture"), forming a subset $\mathcal{A}_{\text{super}} \subset \mathcal{A}$ used later for unknown detection (section 3.5). Finally, we augment with 164 descriptive words mined from captions in the GoldG corpus [15], the dataset used to pre-train YOLO-World, for better generalization. The detailed attribute composition is described in Sec. A of the supplementary. Analogously to eqs. (1) and (2), the text encoder maps these into text embeddings of attributes and yields attribute logits:

$$z_{\text{att}} = \gamma\,\text{BN}(E_{\text{vis}})\,(\|E_{\text{text}}^{\text{att}}\|_2)^\top + b \in \mathbb{R}^{Q\times N_a}, \quad E_{\text{text}}^{\text{att}} = f_{\text{text}}(\mathcal{A}) \in \mathbb{R}^{N_a\times d}. \tag{4}$$

Each entry $z_{\text{att}}[q,a]$ is the similarity between proposal q and the text embedding of attribute a, indicating how strongly the proposal exhibits that attribute (*e.g.*, "furry" or "metallic"). Each row $z_{\text{att}}[q,:]$ can thus be viewed as a position vector of proposal q in an N_a-dimensional attribute space whose axes are the attributes in $\mathcal{A}$. However, the overall scale of $z_{\text{att}}[q,:]$ varies across proposals: a large, clearly visible object tends to respond strongly to all attributes, whereas a small or blurred one responds weakly to all of them. What characterizes an object is rather which attributes stand out relative to the others. We therefore normalize each row by its own mean and standard deviation to obtain the attribute embedding E_{att}:

$$E_{\text{att}}[q,:] = \frac{z_{\text{att}}[q,:] - \mu_q}{\sigma_q + \epsilon} \in \mathbb{R}^{N_a}, \tag{5}$$

where μ_q and σ_q are the mean and standard deviation of $z_{\text{att}}[q,:]$, and ϵ ensures numerical stability. Proposals with similar relative attribute patterns are thus mapped to nearly identical E_{att} vectors regardless of overall activation magnitude.

Unified Visual-Attribute Embedding. Visual and attribute embeddings capture complementary signals—appearance and semantics, respectively—which we concatenate into a unified embedding:

$$E_{\text{uni}} = [E_{\text{vis}};\, E_{\text{att}}] \in \mathbb{R}^{Q \times D}, \quad D = d + N_a. \tag{6}$$

Each row of E_{uni} jointly encodes both signals, more discriminatively than either alone.

3.3 Forward-Pass Statistics

During the forward pass, ground-truth box matching identifies foreground proposals and their class assignments. For each foreground proposal, we collect two types of statistics in a single pass:

DP-vMF-means Multi-Prototype Clustering. Even within the same class, E_{uni} exhibits substantial variation (*e.g.*, a sleeping black cat *vs.* a running white cat). A single centroid per class cannot faithfully represent these distinct sub-populations, motivating a multi-prototype representation. Since E_{uni} concatenates heterogeneous components (E_{vis} and E_{att}) that differ in scale, we project each row onto the unit hypersphere via ℓ_2 normalization, factoring out proposal-level activation magnitude:

$$\mathbf{e}_q = \|E_{\text{uni}}[q, :]\|_2 \in \mathbb{R}^D. \tag{7}$$

With each proposal placed as a unit-norm directional point—where cosine similarity reduces to a dot product—we adopt DP-vMF-means [32], a nonparametric vMF-based algorithm [2] with a Dirichlet Process prior that automatically determines the number of prototypes per class. Using this algorithm, we build a prototype bank $\mathcal{P} = \{\mathcal{P}_c\}_{c \in \mathcal{K}_t}$ that holds a prototype set $\mathcal{P}_c$ for each known class c. To reflect intra-class diversity, $\mathcal{P}_c$ contains J_c prototypes, where J_c is determined by the data and differs across classes. Each prototype k is stored as sufficient statistics, namely the sum $\mathbf{S}_{c,k}$ of the embeddings assigned to it and their count $N_{c,k}$:

$$\mathcal{P}_c = \big\{(\mathbf{S}_{c,k},\ N_{c,k})\big\}_{k=1}^{J_c}, \quad \mathbf{S}_{c,k} = \sum_{q \in \mathcal{Q}_{c,k}} \mathbf{e}_q, \quad N_{c,k} = |\mathcal{Q}_{c,k}|, \tag{8}$$

where $\mathcal{Q}_{c,k}$ is the set of foreground proposals of class c assigned to prototype k, and $\boldsymbol{\mu}_{c,k} = \mathbf{S}_{c,k}/\|\mathbf{S}_{c,k}\|$ is its mean direction. The bank is built in a single pass over the foreground proposals: each $\mathbf{e}_q$ of class c is compared with the mean directions in $\mathcal{P}_c$ by cosine similarity, and if the highest similarity exceeds a threshold τ, it is added to the $\mathbf{S}_{c,k}$ and $N_{c,k}$ of that prototype; otherwise, it starts a new prototype in $\mathcal{P}_c$. Only $(\mathbf{S}_{c,k}, N_{c,k})$ are stored, not raw feature vectors, making memory consumption independent of dataset size and proportional only to the total number of discovered prototypes. The threshold τ is the sole clustering hyperparameter: a higher τ yields tighter and more numerous prototypes.

Attribute Co-occurrence Graph. In parallel with clustering, we accumulate an attribute co-occurrence matrix $\mathbf{C} \in \mathbb{R}^{N_a \times N_a}$ over foreground proposals. Since E_{att} (eq. (5)) is z-score normalized, a positive entry marks an attribute that stands out for that proposal; we therefore binarize it at zero, $\mathbf{b}(q) = \mathbb{1}[E_{\text{att}}[q, :] > 0]$, and accumulate pairwise co-occurrences over all foreground proposals as $\mathbf{C} \mathrel{+}= \mathbf{b}(q)\,\mathbf{b}(q)^\top$. However, raw co-occurrence counts are still biased toward frequent attributes that co-occur with everything by chance. Since the downstream objectness score (section 3.5) relies on propagation through this graph, such frequency bias would produce noisy scores that cannot reliably distinguish unknown objects from known ones, degrading detection precision. To isolate genuine associations, we convert $\mathbf{C}$ into an attribute co-occurrence graph $\mathbf{A} \in \mathbb{R}^{N_a \times N_a}$ via Positive Pointwise Mutual Information (PPMI). For attributes $a_i, a_j \in \mathcal{A}$, let $P(a_i, a_j) = \mathbf{C}_{ij} / \sum_{i',j'} \mathbf{C}_{i'j'}$ and $P(a_i) = \sum_j \mathbf{C}_{ij} / \sum_{i',j'} \mathbf{C}_{i'j'}$ denote the joint and marginal frequencies derived from $\mathbf{C}$:

$$\text{PMI}(a_i, a_j) = \log \frac{P(a_i, a_j)}{P(a_i)\,P(a_j)}, \quad \mathbf{A}_{ij} = \max\big(\text{PMI}(a_i, a_j),\, 0\big), \tag{9}$$

where $\max(\cdot, 0)$ retains only positively associated pairs and discards negative associations that are uninformative for propagation. Self-loops are removed and rows ℓ_1-normalized to form a row-stochastic propagation matrix.

3.4 Known-Class Detection

At inference, we score each proposal against the prototype bank with MAS, calibrate it to the scale of the detector's text logits, and fuse the two for known-class detection.

Mixture-Aggregate Score (MAS). Given the prototypes $\boldsymbol{\mu}_{c,k}$ in $\mathcal{P}_c$, a naive approach simply selects the cosine similarity to the nearest prototype, $\max_k \mathbf{e}_q^\top \boldsymbol{\mu}_{c,k}$. However, not all prototypes are equally reliable: some represent compact, well-populated sub-clusters while others may be diffuse or rare. We therefore derive per-prototype statistics from the sufficient statistics:

$$\pi_{c,k} = \frac{N_{c,k}}{\sum_{k'} N_{c,k'}}, \quad \bar{R}_{c,k} = \frac{\|\mathbf{S}_{c,k}\|}{N_{c,k}}, \quad \kappa_{c,k} = \frac{\bar{R}_{c,k}(D - \bar{R}_{c,k}^2)}{1 - \bar{R}_{c,k}^2}, \tag{10}$$

where $\pi_{c,k}$ is the mixing proportion (fraction of samples in prototype k), $\bar{R}_{c,k}$ is the mean resultant length, and $\kappa_{c,k}$ is the concentration estimated via the Banerjee approximation [2]: higher κ indicates a tighter cluster around $\boldsymbol{\mu}$, and lower κ a diffuse cluster whose mean direction is less reliable. Using these statistics, we aggregate across all prototypes via a weighted sum:

$$z_{\text{MAS}}[q, c] = \sum_{k=1}^{J_c} w_{c,k}\ \mathbf{e}_q^\top \boldsymbol{\mu}_{c,k}, \quad w_{c,k} = \frac{\pi_{c,k}\ \log\max(\kappa_{c,k},\, 1)}{\sum_{k'} \pi_{c,k'}\ \log\max(\kappa_{c,k'},\, 1)}, \tag{11}$$

where $w_{c,k}$ up-weights prototypes that are both prevalent (high π) and compact (high κ), yielding more reliable similarity estimates than the naive maximum. Since MAS logits and text logits occupy different ranges, we apply z-score calibration to obtain the MAS probability:

$$P_{\text{MAS}}(c \mid q) = \sigma\big(\bar{z}_{\text{MAS}}[q, c]\big), \quad \bar{z}_{\text{MAS}} = \frac{z_{\text{MAS}} - \mu_m}{s_m} \cdot s_t + \mu_t, \tag{12}$$

where (μ_m, s_m) and (μ_t, s_t) are the mean and standard deviation of z_{MAS} and z_{cls} (eq. (2)) over all proposals and classes in the current image.

Score Fusion. The fused score linearly combines MAS with the original VLOD's text-based probability $P_{\text{text}}(c \mid q) = \sigma(z_{\text{cls}}[q, c])$:

$$P_{\text{fused}}(c \mid q) = \beta \cdot P_{\text{MAS}}(c \mid q) + (1 - \beta) \cdot P_{\text{text}}(c \mid q), \tag{13}$$

where $\beta \in [0, 1]$ controls the MAS contribution.

3.5 Unknown Object Detection

Attribute Objectness Score (AOS). AOS combines two complementary attribute-based signals. The **superclass score** O_{sc} takes the strongest response among superclass attributes (*e.g.*, "animal" or "furniture"), indicating that the proposal is some kind of object. The **co-occurrence score** O_{co} propagates the proposal's soft attribute activations $\sigma(z_{\text{att}}[q, :])$ through the graph $\mathbf{A}$ (eq. (9)), and is high when its attributes co-occur as they typically do on objects:

$$O_{\text{sc}}(q) = \max_{a \in \mathcal{A}_{\text{super}}} \sigma(z_{\text{att}}[q, a]), \quad O_{\text{co}}(q) = \max_j \big(\sigma(z_{\text{att}}[q, :])^\top \mathbf{A}\big)_j. \tag{14}$$

Unknown Score. The unknown score weights AOS by the complement of the known-class confidence:

$$P_u(q) = O_{\text{aos}}(q) \cdot \big(1 - \max_c P_{\text{fused}}(c \mid q)\big), \quad O_{\text{aos}}(q) = \max\big(O_{\text{sc}}(q),\, O_{\text{co}}(q)\big). \tag{15}$$

High AOS combined with low known-class confidence indicates an unknown. At inference, P_u is appended to the C_t known-class scores and each proposal takes the label with the highest of these $C_t + 1$ scores; since P_u vanishes whenever a known class is confident, a confidently known proposal is not also scored as unknown.

4 Experiments

4.1 Experimental Setting

Datasets and Metrics. Following previous OWOD work [11, 8, 36], we evaluate on two benchmarks composed of MS COCO [17] and PASCAL VOC [7]: **M-OWODB** (Superclass-Mixed) splits 80 classes into four non-overlapping tasks of 20 classes each, mixing superclass categories across

Table 1: Experimental results on M-OWODB (top) and S-OWODB (bottom). Results are reported in terms of mean average precision (mAP) for known classes, unknown class recall (U-Rec), and their harmonic mean (H-Score). The best performance is highlighted in **bold**, with the second-best performance underlined. † denotes reproduced results after correcting the M-OWODB annotation-duplication bug identified in [37]. ⋆ denotes the zero-shot VLOD baseline without OWOD capability.

Task IDs	Task 1			Task 2					Task 3					Task 4		
Method	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	Previous mAP	Current mAP	Known mAP
YOLO-World⋆ [5]	74.7	–	–	75.0	55.2	65.1	–	–	65.1	52.1	60.8	–	–	61.0	43.2	56.5
ORE [11]	56.0	4.9	9.0	52.7	26.0	39.4	2.9	5.4	38.2	12.7	29.7	3.9	6.9	29.6	12.4	25.3
OW-DETR [8]	59.2	7.5	13.3	53.6	33.5	42.9	6.2	10.8	38.3	15.8	30.8	5.7	9.6	31.4	17.1	27.8
CAT [20]	60.0	23.7	34.0	55.5	32.7	44.1	19.1	26.7	42.8	18.7	34.8	24.4	28.7	34.4	16.6	29.9
PROB† [41]	66.4	28.3	39.7	62.6	39.2	50.9	26.4	34.8	49.6	33.5	44.2	29.3	35.2	44.0	26.5	39.7
OrthogonalDet† [33]	65.1	36.3	46.6	58.2	44.2	51.2	30.2	38.0	50.9	40.1	47.3	28.7	35.7	49.1	31.5	44.7
O1O [37]	65.1	49.3	56.1	61.0	45.0	53.0	50.3	51.6	50.0	36.5	45.5	49.5	47.4	46.2	31.0	42.4
OWOBJ [39]	61.4	23.6	34.1	58.4	34.4	45.7	23.8	31.3	44.8	27.8	38.8	25.1	30.5	36.4	20.7	32.0
OW-OVD [36]	69.4	50.0	58.1	69.5	41.7	55.6	51.7	53.6	55.5	29.8	46.9	50.6	48.7	47.0	25.2	41.6
CAVE (Ours)	**75.1**	**75.7**	**75.4**	**75.3**	**56.2**	**65.8**	**79.6**	**72.0**	**65.9**	**52.0**	**61.3**	**81.0**	**69.8**	**61.3**	**47.3**	**57.8**
YOLO-World⋆ [5]	78.6	–	–	78.7	64.1	71.0	–	–	71.1	60.7	67.6	–	–	67.7	61.2	66.1
ORE [11]	61.4	1.5	2.9	56.5	26.1	40.6	3.9	7.1	38.7	23.7	33.7	3.6	6.5	33.6	26.3	31.8
OW-DETR [8]	71.5	5.7	10.6	62.8	27.5	43.8	6.2	10.9	45.2	24.9	38.5	6.9	11.7	38.2	28.1	33.1
CAT [20]	74.2	24.0	36.3	67.6	35.5	50.7	23.0	31.6	51.2	32.6	45.0	24.6	31.8	45.4	35.1	42.8
PROB [41]	73.4	17.6	28.4	66.3	36.0	50.4	22.3	30.9	47.8	30.4	42.0	24.8	31.2	42.6	31.7	39.9
OrthogonalDet [33]	71.6	24.6	36.6	64.0	39.9	51.3	27.9	36.1	52.1	42.2	48.8	31.9	38.6	48.7	38.8	46.2
O1O [37]	72.6	49.8	59.1	65.3	44.9	54.6	51.1	52.8	49.5	41.5	46.8	48.1	47.4	47.3	42.0	45.9
OWOBJ [39]	76.2	22.3	34.5	69.8	41.0	54.8	28.7	37.7	50.6	35.7	46.8	30.9	37.2	46.7	36.9	43.2
OW-OVD [36]	78.6	76.2	77.4	78.5	61.5	69.6	79.8	74.4	69.6	55.1	64.7	78.4	70.9	64.8	56.3	62.7
CAVE (Ours)	**79.0**	**76.9**	**77.9**	**78.7**	**64.3**	**71.1**	**81.5**	**75.9**	**70.2**	**60.4**	**66.9**	**80.5**	**73.1**	**65.9**	**61.1**	**64.7**

tasks. **S-OWODB** (Superclass-Separated) uses only COCO classes, also divided into four tasks, where each task contains classes from distinct superclass groups. Detailed dataset statistics are provided in Sec. B of the supplementary. For known classes, we report mean Average Precision (mAP) at IoU threshold 0.50, measured separately for previously known, currently introduced, and all known categories. For unknown classes, we adopt U-Recall (recall for unknowns) as the primary metric and U-mAP (unknown-class average precision) for precision-aware evaluation. We also report the harmonic mean (H-Score) of known mAP and U-Recall to assess the trade-off between the two objectives.

Implementation Details. We adopt YOLO-World [5] with the CLIP-ViT-B/32 [28] text encoder as the frozen VLOD, using the Large configuration (YOLOv8-L with 512-dimensional embeddings). All model parameters are frozen; the learning rate is set to 0. For fusion and clustering, we set $\beta = 0.30$ and $\tau = 0.40$. All experiments use a single RTX 4090 GPU. The forward pass over the labeled data completes in a single epoch with zero gradient computation. Our implementation is based on MMDetection [4]. Further implementation details are in Sec. B of the supplementary. CAVE's task adaptation finished in $\sim$76 **s** on a single GPU, whereas OW-OVD required $\sim$7.7 **h** of prompt tuning ($\sim$368$\times$ longer); detailed cost and memory comparisons are in Sec. D of the supplementary.

4.2 Comparison with State-of-the-Art

Table 1 presents results on M-OWODB (top) and S-OWODB (bottom). In the main results, CAVE achieves the highest known-class mAP and U-Recall at every task, surpassing both the zero-shot VLOD and OW-OVD. Especially on M-OWODB, the advantage over OW-OVD grows as tasks accumulate: the known-mAP gap widens from **+5.7** at Task 1 to **+16.2** at Task 4, because prompt tuning degrades the VLOD's original zero-shot capability across incremental stages while our training-free approach preserves it. The unknown-detection gap is similarly pronounced, with CAVE leading in U-Recall by an average of **+28.0** over OW-OVD across Tasks 1–3 and achieving the highest H-Score at every task, confirming a favorable balance between known and unknown performance. On S-OWODB, where the superclass-separated split makes prompt tuning less harmful, CAVE

Table 2: Ablation of embedding composition on M-OWODB. Attribute / Visual indicate which embeddings (E_{att} / E_{vis}) compose the unified embedding E_{uni} for CAVE. No checkmarks correspond to the zero-shot baseline (text-only). Best per column in **bold**, second-best underlined.

		Task 1	Task 2			Task 3			Task 4		
Attribute	Visual	Current mAP	Previous mAP	Current mAP	Known mAP	Previous mAP	Current mAP	Known mAP	Previous mAP	Current mAP	Known mAP
OW-OVD [36]		69.4	69.5	41.7	55.6	55.5	29.8	46.9	47.0	25.2	41.6
		74.7	75.0	55.2	65.1	65.1	52.1	60.8	61.0	43.2	56.5
✓		**75.1**	**75.3**	56.1	65.7	65.8	51.7	61.1	61.2	46.7	57.6
	✓	74.9	74.9	55.8	65.3	65.2	51.7	60.7	60.8	44.8	56.8
✓	✓	**75.1**	**75.3**	**56.2**	**65.8**	**65.9**	**52.0**	**61.3**	**61.3**	**47.3**	**57.8**

Table 3: Impact of β on M-OWODB. $\beta = 0$ reduces to text-only scoring. Selected β in **bold** header.

β	T1 mAP	T2 mAP	T3 mAP	T4 mAP
0	74.7	65.1	60.8	56.5
0.10	75.2	65.5	61.2	57.3
0.15	**75.3**	65.8	**61.3**	57.5
0.20	**75.3**	65.8	61.2	57.6
0.25	75.2	**65.9**	61.2	57.6
0.30	75.1	65.8	**61.3**	**57.8**
0.35	75.0	65.7	60.9	57.5
0.40	74.8	65.6	60.6	57.4
1.00	57.0	50.0	39.8	37.8

Table 4: Impact of τ on M-OWODB. All use $\beta = 0.30$. Selected τ in **bold** header.

τ	T1 mAP	T2 mAP	T3 mAP	T4 mAP
0.3	75.0	65.7	61.1	57.5
0.4	**75.1**	**65.8**	**61.3**	**57.8**
0.5	75.0	65.7	61.0	57.5
0.6	74.9	65.5	60.9	57.3

Table 5: MAS (eq. (11)) vs. Naive max (nearest-prototype cosine) on M-OWODB.

Mode	T1 mAP	T2 mAP	T3 mAP	T4 mAP
Naive max	75.0	65.6	60.8	56.9
MAS	**75.1**	**65.8**	**61.3**	**57.8**

still consistently outperforms OW-OVD and all other training-based methods—**+2.0** known mAP at Task 4 (64.7 *vs.* 62.7) and **+1.5** average U-Recall on Tasks 1–3. Additional experiments across YOLO-World scales, CLIP variants, and Grounding DINO confirm that these gains are not tied to a specific backbone or VLOD family (Sec. C of the supplementary).

4.3 Ablation and Analysis

Known Detection Analysis. Table 2 ablates the embedding composition: each modality alone improves the final Task 4 mAP over the zero-shot baseline, attribute-only outperforms visual-only, and their concatenation $E_{\text{uni}} = [E_{\text{vis}}; E_{\text{att}}]$ achieves the best Task 4 mAP (57.8), validating the unified representation. Tables 3 and 4 sweep the fusion weight β and the clustering threshold τ. $\beta = 0.30$ maximizes Task 4 (final-task) mAP, and held-out validation on the adaptation data selects nearly the same value (Sec. F.1). The result also remains stable across $\tau \in [0.3, 0.6]$, indicating robustness to the clustering granularity. Notably, $\beta = 1.0$ (MAS only, no text signal) still reaches 37.8 mAP at Task 4, showing that prototype-based scoring captures discriminative class structure on its own—useful when class names are unavailable or unreliable. This is enabled by the visual-attribute prototypes, which encode multi-modal appearance cues beyond what a single text embedding per class can capture. Table 5 compares MAS against naive max (nearest-prototype cosine): MAS's gain widens with task index (+0.9 at T4), confirming that cluster-informed weighting via $\pi_{c,k}$ and $\kappa_{c,k}$ scales better as intra-class diversity grows and a single nearest-prototype score becomes increasingly inadequate.

Unknown Detection Analysis. Table 6 reports U-mAP, a precision-aware metric for unknown detection: CAVE substantially outperforms all baselines at every task (*e.g.*, 18.2 *vs.* 8.6 for OW-OVD at Task 1), confirming that AOS detects unknowns with markedly higher precision. For all methods, U-mAP decreases as tasks progress and fewer unknowns remain annotated, whereas many

Table 6: Unknown detection precision (U-mAP) on M-OWODB across Tasks 1–3.

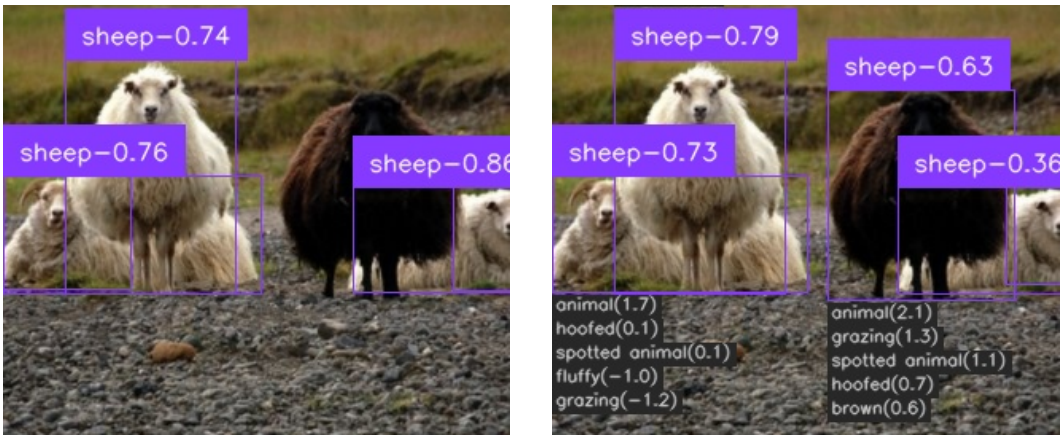
Method	Task 1	Task 2	Task 3
SGROD [9]	2.4	4.9	0.3
SKDF [21]	1.5	1.2	0.7
KTCN [35]	1.2	0.7	0.5
OW-OVD [36]	8.6	4.3	4.0
CAVE (Ours)	**18.2**	**14.9**	**9.4**

Table 7: Ablation of unknown detection components on M-OWODB. O_{sc}: superclass score, O_{co}: co-occurrence, PPMI: frequency normalization.

Components			Task 1		Task 2		Task 3	
O_{sc}	O_{co}	PPMI	U-Rec	U-mAP	U-Rec	U-mAP	U-Rec	U-mAP
✓			59.3	15.0	64.5	12.5	66.3	5.0
	✓		60.4	7.3	70.5	8.7	77.8	8.3
	✓	✓	60.3	11.9	70.5	11.5	77.1	9.0
✓	✓		**75.8**	10.6	79.4	10.7	**81.2**	4.3
✓	✓	✓	75.7	**18.2**	**79.6**	**14.9**	81.0	**9.4**

Table 8: Results on the new benchmarks V3-OWODB and RS-OWODB. Zero-shot and Baseline denote the base detector with text prompts only.

Benchmark	Method	Task 1			Task 2			Task 3			Task 4		
		Known	U-Rec	H-Score	Known	U-Rec	H-Score	Known	U-Rec	H-Score	Known	U-Rec	H-Score
V3-OWODB	Zero-shot	16.5	–	–	14.7	–	–	11.9	–	–	10.0	–	–
	CAVE (Ours)	**19.9**	**68.8**	**30.9**	**17.5**	**73.2**	**28.2**	**14.5**	**76.4**	**24.4**	**13.2**	**78.1**	**22.6**
RS-OWODB	Baseline	54.5	–	–	60.7	–	–	61.0	–	–	59.7	–	–
	CAVE (Ours)	**55.3**	**87.6**	**67.8**	60.6	**87.0**	**71.4**	**62.0**	**84.5**	**71.5**	**62.3**	–	–



(a) Text-only (baseline) (b) **CAVE (Ours)**

Figure 3: Qualitative comparison. (a) The baseline misses the black sheep. (b) CAVE detects all instances. Displayed attributes are the top-responding attribute logits for each instance.

unannotated unknowns are still detected in practice and counted as false positives. Even so, CAVE keeps more than twice the U-mAP of OW-OVD at every task (Sec. F.3). Table 7 ablates the three AOS components (O_{sc}, O_{co}, PPMI): combining O_{sc} and O_{co} yields markedly higher U-Recall than applying either alone, while adding PPMI trades marginal recall for substantial precision gains, achieving the best U-mAP at every task (18.2/14.9/9.4). The three components play complementary roles: O_{sc} provides recall via coarse categorical cues, O_{co} captures attribute-pattern deviations and contributes complementary recall, and PPMI normalizes co-occurrence frequencies to suppress false positives from frequency bias.

Generalization to New Benchmarks. To test generalization beyond COCO categories, we build V3-OWODB from the large-vocabulary V3Det [34] and RS-OWODB from the remote-sensing DIOR [14] (Sec. G). As shown in table 8, CAVE improves known-class mAP on both benchmarks while reaching high unknown recall, without any change to its hyperparameters.

Table 9: Comparison with simple alternatives on M-OWODB Task 4 (known mAP and memory of the stored adaptation state).

	Zero-shot	Centroid	K-means	DBSCAN	HDBSCAN	FINCH	Linear probe	kNN	DP-vMF	**CAVE**
T4 Known mAP	56.5	56.9	56.9	56.6	56.8	56.9	56.5	57.4	56.9	**57.8**
Memory	–	0.3 MB	241 MB	1,483 MB	267 MB	4,447 MB	–	23 MB	2.0 MB	2.0 MB



(a) O_{sc} (superclass score) only (b) O_{co} (co-occurrence score) only

Figure 4: AOS component comparison at Task 1. Known classes (*dining table*, *sofa*, *chair*) are omitted for clarity. (a) O_{sc} responds to objects with clear categorical identity. (b) O_{co} responds to objects with distinctive visual attribute patterns. The two signals are complementary.

Comparison with Simple Alternatives. We further replace DP-vMF-means and MAS with simpler alternatives on the same embedding (table 9): clustering methods such as K-means, DBSCAN [6], HDBSCAN [3], and FINCH [29], as well as a linear probe and kNN, all fall below CAVE, which reaches the best Task 4 mAP with only 2.0 MB of stored statistics. Notably, DP-vMF-means alone does not exceed the other clustering methods, and the gain comes from combining it with MAS.

Qualitative Analysis. Figure 3 shows that CAVE detects a visually atypical black sheep that the text-only baseline misses, by assigning distinct visual-attribute prototypes (white sheep $\rightarrow$ *white*, *fuzzy*; black sheep $\rightarrow$ *brown*, *grazing*). Figure 4 contrasts the two AOS components: O_{sc} activates on objects with clear categorical identity (*e.g.*, mirror, lamp), while O_{co} highlights objects with distinctive attribute patterns (*e.g.*, patterned cushion, glass ornament), confirming complementary unknown coverage. Additional visualizations are in Sec. E of the supplementary.

5 Conclusion

In this paper, we propose CAVE, a fully training-free framework for Open World Object Detection. Adapting a pre-trained VLOD to OWOD typically requires gradient-based fine-tuning, which not only limits deployment on resource-constrained devices but also degrades the detector's own zero-shot capability. CAVE addresses both issues by extracting visual and attribute statistics from a single forward pass and leveraging them at inference through MAS for known-class detection and AOS for unknown discovery, all without any parameter updates. This adaptation completes in about 76 s, roughly 368$\times$ faster than prompt tuning. Experiments on M-OWODB and S-OWODB show that CAVE surpasses training-based OWOD methods across all incremental tasks, and further results on a large-vocabulary and a remote-sensing benchmark show that it generalizes beyond them. These results open a practical path toward real-world OWOD where a detector adapts to new environments without any gradient updates.

Limitations. CAVE relies on an attribute vocabulary $\mathcal{A}$ and a superclass subset $\mathcal{A}_{\text{super}}$ derived from VAW and GoldG, which target generic everyday concepts. On domains where these attributes provide weak coverage, CAVE may require domain-specific vocabulary curation.

Acknowledgments and Disclosure of Funding

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2026-25480253), the "Advanced GPU Utilization Support Program" funded by the Government of the Republic of Korea (Ministry of Science and ICT), and the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (IITP-2026-RS-2026-25615490, AI Star Fellowship Support Program, RS-2019-II190079, Artificial Intelligence Graduate School Program (Korea University), and RS-2024-00457882, AI Research Hub Project).

References

[1] Yong Hyun Ahn, Gyeong-Moon Park, and Seong Tae Kim. LINe: Out-of-distribution detection by leveraging important neurons. In *CVPR*, 2023.

[2] Arindam Banerjee, Inderjit S Dhillon, Joydeep Ghosh, Suvrit Sra, and Greg Ridgeway. Clustering on the unit hypersphere using von mises-fisher distributions. *JMLR*, 6:1345–1382, 2005.

[3] Ricardo J. G. B. Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *PAKDD*, pages 160–172, 2013.

[4] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[5] Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-world: Real-time open-vocabulary object detection. In *CVPR*, pages 16901–16911, 2024.

[6] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, pages 226–231, 1996.

[7] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88:303–338, 2010.

[8] Akshita Gupta, Sanath Narayan, KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Mubarak Shah. Ow-detr: Open-world detection transformer. In *CVPR*, pages 9235–9244, 2022.

[9] Yulin He, Wei Chen, Siqi Wang, Tianrui Liu, and Meng Wang. Recalling unknowns without losing precision: An effective solution to large model-guided open world object detection. *IEEE Trans. Image Process.*, 2024.

[10] Jun-Woo Heo, Keonhee Park, and Gyeong-Moon Park. Detecting unknown objects via energy-based separation for open world object detection. In *CVPR*, 2026.

[11] KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Towards open world object detection. In *CVPR*, pages 5830–5840, 2021.

[12] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE TPAMI*, 36(3):453–465, 2014.

[13] Mengcheng Lan, Chaofeng Chen, Yiping Ke, Xinjiang Wang, Litong Feng, and Wayne Zhang. Proxyclip: Proxy attention improves clip for open-vocabulary segmentation. In *ECCV*, 2024.

[14] Ke Li, Gang Wan, Gong Cheng, Liqiu Meng, and Junwei Han. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS Journal of Photogrammetry and Remote Sensing*, 159:296–307, 2020.

[15] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. Grounded language-image pre-training. In *CVPR*, pages 10965–10975, 2022.

[16] Yixia Li, Boya Xiong, Guanhua Chen, and Yun Chen. Setar: Out-of-distribution detection with selective low-rank approximation. In *NeurIPS*, 2024.

[17] 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*, pages 740–755, 2014.

[18] Mingxuan Liu, Tyler L. Hayes, Elisa Ricci, Gabriela Csurka, and Riccardo Volpi. Shine: Semantic hierarchy nexus for open-vocabulary object detection. In *CVPR*, 2024.

[19] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *ECCV*, 2024.

[20] 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 *CVPR*, pages 19681–19690, 2023.

[21] Shuailei Ma, Yuefeng Wang, Ying Wei, Jiaqi Fan, Xinyu Sun, Peihao Chen, and Enming Zhang. A simple knowledge distillation framework for open-world object detection. *arXiv preprint arXiv:2312.08653*, 2023.

[22] Shuaicheng Niu, Chunyan Miao, Guohao Chen, Pengcheng Wu, and Peilin Zhao. Test-time model adaptation with only forward passes. In *ICML*, pages 38298–38315, 2024.

[23] Jiancheng Pan, Yanxing Liu, Yuqian Fu, Muyuan Ma, Jiahao Li, Danda Pani Paudel, Luc Van Gool, and Xiaomeng Huang. Locate anything on earth: Advancing open-vocabulary object detection for remote sensing community. In *AAAI*, 2025.

[24] Keon-Hee Park, Kyungwoo Song, and Gyeong-Moon Park. Pre-trained vision and language transformers are few-shot incremental learners. In *CVPR*, 2024.

[25] Keon-Hee Park, Seun-An Choe, and Gyeong-Moon Park. SFUOD: Source-free unknown object detection. In *ICCV*, 2025.

[26] Can Peng, Kun Zhao, and Brian C Lovell. Faster ilod: Incremental learning for object detectors based on faster rcnn. *Pattern Recognition Letters*, 140:109–115, 2020.

[27] Khoi Pham, Kushal Kafle, Zhe Lin, Zhihong Ding, Scott Cohen, Quan Tran, and Abhinav Shrivastava. Learning to predict visual attributes in the wild. In *CVPR*, pages 13018–13028, 2021.

[28] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763, 2021.

[29] M. Saquib Sarfraz, Vivek Sharma, and Rainer Stiefelhagen. Efficient parameter-free clustering using first neighbor relations. In *CVPR*, pages 8934–8943, 2019.

[30] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. Incremental learning of object detectors without catastrophic forgetting. In *ICCV*, pages 3400–3409, 2017.

[31] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, volume 30, pages 4077–4087, 2017.

[32] Julian Straub, Trevor Campbell, Jonathan P How, and John W Fisher III. Small-variance nonparametric clustering on the hypersphere. In *CVPR*, pages 334–342, 2015.

[33] Zhicheng Sun, Jinghan Li, and Yadong Mu. Exploring orthogonality in open world object detection. In *CVPR*, pages 17302–17312, 2024.

[34] Jiaqi Wang, Pan Zhang, Tao Chu, Yuhang Cao, Yujie Zhou, Tong Wu, Bin Wang, Conghui He, and Dahua Lin. V3Det: Vast vocabulary visual detection dataset. In *ICCV*, 2023.

[35] Xing Xi, Yangyang Huang, Jinhao Lin, and Ronghua Luo. Ktcn: Enhancing open-world object detection with knowledge transfer and class-awareness neutralization. In *IJCAI*, pages 1462–1470, 2024.

[36] Xing Xi, Yangyang Huang, Ronghua Luo, and Yu Qiu. Ow-ovd: Unified open world and open vocabulary object detection. In *CVPR*, pages 25454–25464, 2025.

[37] Mısra Yavuz and Fatma Güney. O1o: Grouping of known classes to identify unknown objects as odd-one-out. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 614–629, 2024.

[38] Renrui Zhang, Rongyao Fang, Peng Gao, Wei Zhang, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of clip for few-shot classification. In *ECCV*, pages 493–510, 2022.

[39] Shan Zhang, Yao Ni, Jinhao Du, Yuan Xue, Philip H. S. Torr, Piotr Koniusz, and Anton van den Hengel. Open-world objectness modeling unifies novel object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 30332–30342, 2025.

[40] Orr Zohar, Alejandro Lozano, Shelly Goel, Serena Yeung, and Kuan-Chieh Wang. Open world object detection in the era of foundation models. *arXiv preprint arXiv:2312.05745*, 2023.

[41] Orr Zohar, Kuan-Chieh Wang, and Serena Yeung. Prob: Probabilistic objectness for open world object detection. In *CVPR*, pages 11444–11453, 2023.

Training-Free Open World Object Detection Supplementary Material

Supplementary Outline

A. Attribute Vocabulary Composition p.2
B. Experiment Settings p.2
B.1 Benchmark Configuration p.2
B.2 Implementation Details p.3
C. Extended Analysis p.4
C.1 Cross-Model Generalization p.4
C.2 Cross-VLOD Generalization p.4
C.3 Extended Embedding Composition Analysis p.4
D. Computational Cost Analysis p.5
D.1 Training vs. Training-Free Cost p.5
D.2 Deployment Overhead p.5
E. Qualitative Results p.6
E.1 Incremental Open-World Detection p.6
E.2 Attribute Activation Examples p.6
E.3 Unknown Detection Examples p.6
F. Additional Analyses p.8
F.1 Held-out Selection of β p.8
F.2 Comparison with Simple Alternatives p.8
F.3 Unknown Ground Truth and U-mAP p.8
F.4 Per-Class Contribution of the Two Embeddings p.9
F.5 Adaptation on Predicted Boxes p.9
F.6 Further Notes on Scoring p.9
G. Additional Benchmarks p.10
G.1 V3-OWODB p.10
G.2 RS-OWODB p.10

In this supplementary material, we provide additional details and analyses that complement the main paper. section A describes the attribute vocabulary construction. section B covers benchmark settings and implementation details. section C presents extended analysis on cross-model and cross-VLOD generalization, and embedding composition. section D provides computational cost analysis. section E presents qualitative detection results. section F provides additional analyses, and section G reports results on two additional benchmarks, V3-OWODB and RS-OWODB.

A Attribute Vocabulary Composition

Our attribute vocabulary $\mathcal{A}$ ($N_a = 512$) is composed of *three complementary sources*: a curated visual-attribute taxonomy (VAW), a paper-defined superclass list, and a caption-derived discovery corpus (GoldG). This composition is deliberate—relying on VAW alone would inherit its annotator-driven bias, whereas augmenting with caption-mined descriptors aligns the vocabulary with the distribution of natural-language object descriptions seen during VLM pre-training, and superclass terms provide a coarse-grained signal for unknown-object detection.

(1) VAW seed (325 attributes). We initialize a seed vocabulary $\mathcal{V}_{\text{vaw}}$ from the Visual Attributes in the Wild (VAW) [27] dataset, which provides attribute annotations across diverse types (color, material, shape, texture, pattern, state, *etc.*). Two filters are applied: *(i) Class-name leakage:* attributes matching a known class name (*e.g.*, "riding bike" for *bicycle*) are discarded to prevent vocabulary leakage. *(ii) Bias-sensitive descriptors:* we exclude terms related to hair color, skin color, and race (*e.g.*, "blond", "dark skinned", "asian"), as well as person-specific activity phrases (*e.g.*, "playing baseball", "wearing glasses") that describe human actions rather than visual object properties. After deduplication, $|\mathcal{V}_{\text{vaw}}| = 325$.

(2) Superclass (23 attributes). We add $|\mathcal{V}_{\text{super}}| = 23$ paper-defined superclass terms for the objectness signal (Sec. 3.5 of the main paper): *animal, plant, food, fruit, clothing, furniture, vehicle, tool, weapon, electronic device, machine, container, instrument, sports equipment, artwork, document, natural object, accessory, toy, street fixture, component, kitchenware, and appliance*.

(3) GoldG augmentation (164 attributes). We augment $\mathcal{V}_{\text{vaw}} \cup \mathcal{V}_{\text{super}}$ with new descriptive words mined from GoldG captions [15], the grounded image–text corpus used to pre-train YOLO-World. In a single pass over the captions, we accumulate a frequency map $\mathcal{F}$ over lemmas that (a) do not already appear in $\mathcal{V}_{\text{vaw}} \cup \mathcal{V}_{\text{super}}$, and (b) play a syntactically descriptive role—adjective ("shiny"), participial modifier ("broken" in "broken window"), compound modifier ("leather" in "leather bag"), or noun in a material phrase ("stone" in "made of stone")—as determined by the spaCy dependency parser. The same class-leakage and bias filters are applied to $\mathcal{F}$. We then rank the surviving lemmas by descending frequency and take the top-K, with K chosen so that the final vocabulary

$$\mathcal{A} = \mathcal{V}_{\text{vaw}} \cup \mathcal{V}_{\text{super}} \cup \mathcal{V}_{\text{gold}}$$

reaches the target $N_a = 512$, giving $|\mathcal{V}_{\text{gold}}| = 164$.

S.Table 1 summarizes the composition by source. Notably, caption-mined descriptors contribute $\sim$32% of the final vocabulary, so the attribute space is not dominated by VAW's annotator-curated set.

S.Table 1: Composition of the attribute vocabulary $\mathcal{A}$ ($N_a = 512$) by source. The vocabulary is treated as a flat list at runtime; cross-source duplicates are removed (earlier sources keep precedence).

Source	#	%	Representative examples
Superclass (paper-defined)	23	4.5	animal, vehicle, kitchenware, appliance
VAW seed [27]	325	63.5	black, wooden, striped, round, broken, sliced
GoldG caption-mined [15]	164	32.0	parked, holding, decorated, vintage, exposed
Total (unique)	**512**	**100.0**	

B Experiment Settings

B.1 Benchmark Configuration

S.Table 2 summarizes the benchmark configurations. **M-OWODB** (Superclass-Mixed) places all 20 Pascal VOC classes into Task 1 and distributes the remaining 60 MS-COCO classes across Tasks 2–4. Because super-categories are mixed across tasks (*e.g.*, vehicle classes appear in both Task 1 and Task 2), this benchmark introduces inter-task semantic overlap. **S-OWODB** (Superclass-Separated) groups all categories within a super-category into a single task, providing a stricter split with no semantic leakage. Further details on the per-task class assignments, which follow the standard splits established by Gupta *et al.* [11], are provided in S.Table 3.

S.Table 2: Benchmark configuration for M-OWODB and S-OWODB.

	M-OWODB				S-OWODB			
	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
# of Classes	20	20	20	20	19	21	20	20
# of training images	16,551	45,520	39,402	40,260	89,490	55,870	39,402	38,903
# of training objects	47,223	113,741	114,452	138,996	421,243	163,512	114,452	160,794
# of test images	10,246				4,952			
# of test objects	14,976	4,966	4,826	6,039	17,786	7,159	4,826	7,010

S.Table 3: Per-task class lists for M-OWODB and S-OWODB.

Benchmark	Task	Classes	#
M-OWODB	T1	aeroplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant, sheep, sofa, train, tv monitor	20
	T2	truck, traffic light, fire hydrant, stop sign, parking meter, bench, elephant, bear, zebra, giraffe, backpack, umbrella, handbag, tie, suitcase, microwave, oven, toaster, sink, refrigerator	20
	T3	frisbee, skis, snowboard, sports ball, kite, baseball bat, baseball glove, skateboard, surfboard, tennis racket, banana, apple, sandwich, orange, broccoli, carrot, hot dog, pizza, donut, cake	20
	T4	bed, toilet, laptop, mouse, remote, keyboard, cell phone, book, clock, vase, scissors, teddy bear, hair drier, toothbrush, wine glass, cup, fork, knife, spoon, bowl	20
S-OWODB	T1	aeroplane, bicycle, bird, boat, bus, car, cat, cow, dog, horse, motorbike, sheep, train, elephant, bear, zebra, giraffe, truck, person	19
	T2	traffic light, fire hydrant, stop sign, parking meter, bench, chair, dining table, potted plant, backpack, umbrella, handbag, tie, suitcase, microwave, oven, toaster, sink, refrigerator, bed, toilet, sofa	21
	T3	frisbee, skis, snowboard, sports ball, kite, baseball bat, baseball glove, skateboard, surfboard, tennis racket, banana, apple, sandwich, orange, broccoli, carrot, hot dog, pizza, donut, cake	20
	T4	laptop, mouse, remote, keyboard, cell phone, book, clock, vase, scissors, teddy bear, hair drier, toothbrush, wine glass, cup, fork, knife, spoon, bowl, tv monitor, bottle	20

B.2 Implementation Details

We use YOLO-World-L with a frozen CLIP-ViT-B/32 text encoder ($d = 512$) as the base detector. All detection-related settings follow the YOLO-World defaults: input images are resized to 640×640; for post-processing, a score threshold of 0.001 is applied before non-maximum suppression (NMS) with an IoU threshold of 0.7 and up to 300 detections are retained per image; for GT-to-prediction matching, we employ the TaskAligned assigner with an alignment metric weight of 0.5, an IoU exponent of 6.0, and CIoU as the IoU measure.

Since CAVE operates in a training-free regime—all model parameters remain frozen and no gradient is computed—we make the following modifications to the standard training pipeline: (1) all data augmentations (*e.g.*, mosaic, mixup, random flip, and HSV jitter) are removed, so that the training and test pipelines are identical; (2) the assigner top-k is set to 1, as there is no need for soft positive assignment; (3) the learning rate is 0 and the number of epochs is 1, reducing the "training" phase to a single forward pass through the labeled data for collecting sufficient statistics; and (4) we use a batch size of 16 on a single RTX 4090 GPU.

C Extended Analysis

C.1 Cross-Model Generalization

S.Table 4 evaluates CAVE across different YOLO-World backbones and text encoders on M-OWODB. We test three model scales—Small (S), Medium (M), and Large (L)—and two CLIP text encoders: ViT-B/32 ($d = 512$) and ViT-L/14 ($d = 768$). All configurations use the same hyperparameters ($\tau = 0.40$, $\beta = 0.30$) without per-model tuning. The starred configuration (YOLO-World-L with CLIP-ViT-B/32) matches the main paper. CAVE consistently improves Task 4 Known mAP over the zero-shot baseline across model scales and text encoders, indicating that the gain does not hinge on a particular backbone or embedding dimension.

S.Table 4: Cross-model generalization on M-OWODB. $\star$ marks the configuration used in the main paper (YOLO-World-L, CLIP-ViT-B/32). All variants share the same hyperparameters ($\tau = 0.40$, $\beta = 0.30$).

		Task 1			Task 2			Task 3			Task 4
Model	Method	Known mAP	U-Rec	H-Score	Known mAP	U-Rec	H-Score	Known mAP	U-Rec	H-Score	Known mAP
YOLO-World-S CLIP-ViT-B/32	Zero-shot	68.3	–	–	58.2	–	–	53.1	–	–	48.2
	CAVE	**69.3**	77.9	73.3	**59.3**	79.6	68.0	**53.3**	79.0	63.7	**49.2**
YOLO-World-M CLIP-ViT-B/32	Zero-shot	74.0	–	–	63.5	–	–	58.8	–	–	54.3
	CAVE	73.9	75.7	74.8	**64.4**	79.8	71.3	**59.1**	79.7	67.9	**55.6**
YOLO-World-L* CLIP-ViT-B/32	Zero-shot	74.7	–	–	65.1	–	–	60.8	–	–	56.5
	CAVE	**75.1**	**75.7**	**75.4**	**65.8**	**79.6**	**72.0**	**61.3**	**81.0**	**69.8**	**57.8**
YOLO-World-L CLIP-ViT-L/14	Zero-shot	73.5	–	–	65.1	–	–	61.1	–	–	56.7
	CAVE	**73.9**	72.4	73.1	**66.1**	76.0	70.7	**61.9**	75.6	68.1	**58.1**

C.2 Cross-VLOD Generalization

To verify that CAVE is not tied to a specific VLOD architecture, we additionally port it to Grounding DINO [19], a DETR-style transformer detector with a BERT text encoder—structurally distinct from YOLO-World's CNN-based dense detector with a CLIP text encoder. The same prototype-bank, attribute-vocabulary, and AOS modules are reused without modification; only the host-detector adaptation layer (text-feature injection and proposal embedding extraction) is re-implemented. S.Table 5 reports M-OWODB results for Grounding DINO Base under identical hyperparameters ($\tau = 0.40$, $\beta = 0.30$), validating that the gains observed with YOLO-World transfer to a structurally distinct VLOD family.

S.Table 5: Cross-VLOD generalization on M-OWODB. CAVE applied to Grounding DINO Base under the same hyperparameters as the main paper ($\tau = 0.40$, $\beta = 0.30$).

		Task 1			Task 2			Task 3			Task 4
Model	Method	Known mAP	U-Rec	H-Score	Known mAP	U-Rec	H-Score	Known mAP	U-Rec	H-Score	Known mAP
Grounding DINO	Zero-shot	69.4	–	–	63.8	–	–	63.6	–	–	59.9
	CAVE	**69.8**	64.1	66.8	**64.8**	59.2	61.9	**64.4**	66.1	65.2	**61.3**

C.3 Extended Embedding Composition Analysis

S.Table 6 extends the embedding ablation (table 2 in the main paper) by crossing three embedding configurations with four clustering thresholds τ. Across all τ values, the unified representation ($D = d + N_a$) consistently achieves the highest Task 4 Known mAP, confirming that the advantage of combining visual and attribute embeddings is not an artifact of a particular τ setting.

S.Table 6: Known mAP (AP_{50}) on M-OWODB across embedding configurations and clustering thresholds τ ($\beta = 0.30$). **Bold** indicates the best per τ. $\star$ denotes the setting reported in the main paper.

τ	Embedding	T1 mAP	T2 mAP	T3 mAP	T4 mAP
0.30	Attribute only	**75.0**	65.6	61.0	57.4
	Visual only	74.8	65.2	60.7	56.7
	Unified	**75.0**	**65.7**	**61.1**	**57.5**
$\mathbf{0.40}^{\star}$	Attribute only	**75.1**	65.7	61.1	57.6
	Visual only	74.9	65.3	60.7	56.8
	Unified	**75.1**	**65.8**	**61.3**	**57.8**
0.50	Attribute only	74.9	65.6	60.9	57.4
	Visual only	74.7	65.2	60.5	56.6
	Unified	**75.0**	**65.7**	**61.0**	**57.5**
0.60	Attribute only	74.8	65.4	60.8	57.2
	Visual only	74.6	65.0	60.4	56.4
	Unified	**74.9**	**65.5**	**60.9**	**57.3**

D Computational Cost Analysis

S.Table 7 summarizes the cost comparison between CAVE and the OW-OVD baseline; the subsections below break down each component.

S.Table 7: Cost summary on M-OWODB Task 1. CAVE's training-free adaptation completes in less time than a single OW-OVD epoch and uses less than half the VRAM, while adding only $\sim$0.5% of the base detector's parameters and incurring $\sim$5% inference overhead.

	CAVE (Ours)	**OW-OVD baseline**	**Ratio**
Per-epoch time	75.7 s	696.6 s	$\sim$9.2$\times$ slower
Total wall clock	**75.7 s**	$\sim$**7.7 h** (40 epochs)	$\sim$368$\times$ slower
GPU VRAM	2,498 MB	5,261 MB	$\sim$2.1$\times$ more
Added memory	$\sim$2.3 MB ($\sim$0.5%)	$\sim$4.2 MB ($\sim$1.0%)	$\sim$1.8$\times$ more
Inference	88.3 FPS ($\sim$5% overhead)	—	—

D.1 Training vs. Training-Free Cost

We compare CAVE against OW-OVD, a prompt-tuning baseline, on Task 1 of M-OWODB. Being training-free, CAVE only performs a single data-collection pass over the training set, completing in 75.7 s ($\sim$1.3 min) with 2,498 MB GPU VRAM. This pass is lightweight by design: DP-vMF-means compares each foreground embedding only with the J_c prototypes of its class at $O(J_c D)$, so the whole pass costs $O(N_{\text{fg}} \bar{J} D)$ ($\bar{J} = 3.8$) with no re-assignment or second pass. OW-OVD, by contrast, runs 40 epochs of prompt tuning at 696.6 s ($\sim$11.6 min) per epoch and 5,261 MB VRAM—a single epoch alone is $\sim$9.2$\times$ slower than CAVE's full pass, and the full 40-epoch run takes $\sim$7.7 hours, $\sim$368$\times$ longer.

D.2 Deployment Overhead

Additional memory. CAVE stores the adjacency matrix $\mathbf{A} \in \mathbb{R}^{N_a \times N_a}$ and per-class sufficient statistics (directional sums $\mathbf{S}_{c,k} \in \mathbb{R}^{d+N_a}$ and counts $N_{c,k}$) on top of the pre-trained detector. The total number of additional float32 elements is:

$$M_{\text{add}} = \underbrace{N_a^2}_{\text{adjacency}} + \underbrace{C_t \bar{J}\,(d + N_a + 1)}_{\text{prototypes}},$$

where N_a is the attribute vocabulary size, d the visual embedding dimension, C_t the number of known classes, and $\bar{J}$ the average number of prototypes per class. In our setting ($N_a = 512$, $d = 512$, $C_t = 80$, $\bar{J} = 3.8$), this amounts to 2.3 MB ($\sim$0.5% of the 424 MB YOLO-World-L

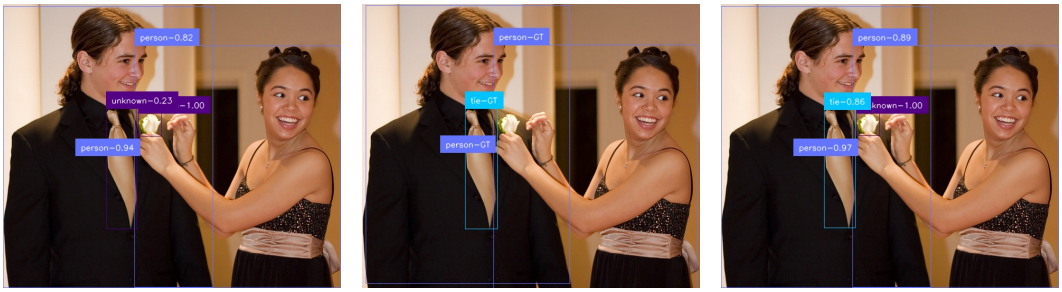
model). For comparison, OW-OVD persistently stores per-class learnable text embeddings ($C_t \times d$, $\sim$0.16 MB at $C_t = 80$) and a pool of selected attribute embeddings ($\sim$25 per class, $\sim$4 MB), totaling $\sim$4.2 MB—also negligible relative to the base detector but $\sim$1.8$\times$ larger than CAVE's footprint.

Runtime. On the M-OWODB test set (9,119 unique images after deduplication), the zero-shot baseline runs at 92.9 FPS (98.1 s), while CAVE achieves 88.3 FPS (103.3 s), an overhead of $\sim$0.57 ms per image ($\sim$5%). This confirms that CAVE can endow a pre-trained VLOD with open-world detection capability without any training and at negligible inference overhead.

E Qualitative Results

E.1 Incremental Open-World Detection

S.Figure 1 demonstrates the open-world detection behavior across incremental tasks. At Task 1, the detector recognizes two *person* instances and flags the necktie as unknown (score 0.23), since *tie* has not yet been introduced. At Task 2, *tie* enters the known vocabulary, and the model now correctly classifies it with high confidence (0.86). Meanwhile, the boutonnière—which never appears in any task vocabulary—remains unknown throughout, showing that the model successfully distinguishes truly novel objects from incrementally learned ones.



(a) Task 1: *tie* is unknown (b) Ground truth (c) Task 2: *tie* is known

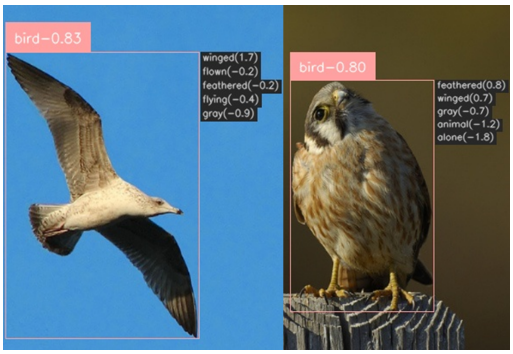
S.Figure 1: Incremental open-world detection on M-OWODB. As the task vocabulary grows from Task 1 (a) to Task 2 (c), the previously unknown *tie* is recognized as known, while the boutonnière—absent from all tasks—remains correctly flagged as unknown.

E.2 Attribute Activation Examples

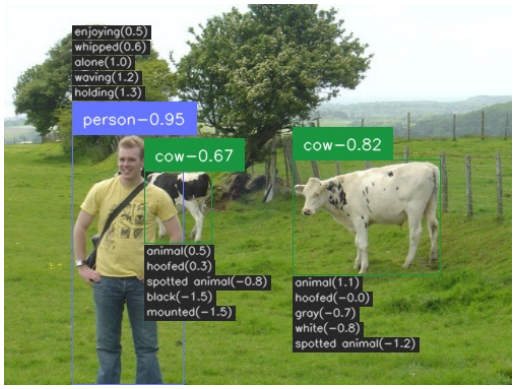
S.Figure 2 visualizes the top-5 attribute logits for individual detections. In (a), two birds of the same class show distinct activations: the flying gull triggers `winged`, `flying`, while the perched falcon highlights `feathered`, `alone`. In (b), two cows in the same image are distinguished by `black`, `spotted animal` (left) vs. `white`, `gray` (right). These examples show that the attribute branch captures meaningful intra-class diversity and provides interpretable per-detection evidence without any training.

E.3 Unknown Detection Examples

The main paper compares O_{sc} and O_{co} individually. Here we additionally show the combined Attribute Objectness Score (AOS) output ($\max(O_{\text{sc}}, O_{\text{co}})$), which is the signal actually used for unknown detection. S.Figure 3 visualizes all three signals side by side on two M-OWODB Task 1 scenes; known classes (*dining table*, *chair*, *sofa*) are omitted for clarity. The two component signals are complementary: O_{sc} (panels a, d) responds to objects with strong categorical identity such as painting and glass table, while O_{co} (panels b, e) captures objects that lack clear categorical identity but exhibit salient attribute patterns—including small or ambiguous objects such as flower arrangements and candles. By taking the per-proposal maximum of the two, the combined AOS (panels c, f) recovers detections from both signals and yields broader unknown coverage than either component alone, validating the design choice in Sec. 3.5 of the main paper.



(a) `flying`, `winged` (left bird) vs. `feathered`, `alone` (right bird).



(b) `black`, `spotted animal` (left cow) vs. `white`, `gray` (right cow).

S.Figure 2: Per-detection attribute activations at Task 1. Each box displays the top-5 attribute logits. Instances of the same class exhibit distinct activation profiles depending on appearance, pose, and context, demonstrating that the attribute branch captures intra-class diversity without training.



(a) Scene 1: O_{sc} only



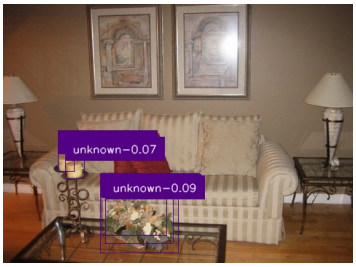
(b) Scene 1: O_{co} only



(c) Scene 1: AOS



(d) Scene 2: O_{sc} only



(e) Scene 2: O_{co} only



(f) Scene 2: AOS

S.Figure 3: AOS components (O_{sc}, O_{co}) and their maximum on two M-OWODB Task 1 scenes.

F Additional Analyses

F.1 Held-out Selection of β

To further validate the choice of β, we build a validation split: at each task, the task's own training images are split 9:1 into a fit part and a held-out part, the adaptation runs on the fit part alone, and β is swept on the held-out part, keeping the value with the highest known AP_{50}. We repeat the whole procedure with three seeds. As shown in S.Table 8, all three seeds select a value within 0.04 of $\beta = 0.30$, and validation mAP is flat in this range (55.2, 55.2, and 55.0 at Task 4 for $\beta = 0.25$, 0.30, and 0.35).

S.Table 8: β selected on the held-out split (averaged over the four tasks) on M-OWODB.

	Seed 1	Seed 2	Seed 3
Selected β	0.34	0.30	0.28

F.2 Comparison with Simple Alternatives

S.Table 9 replaces DP-vMF clustering and MAS with simpler ways of using the same unified embedding E_{uni} while keeping the text-score fusion fixed. CAVE outperforms every alternative, including kNN with 70 exemplars per class, which stores ten times more state. The density-based methods must hold the raw points in memory and need one to three orders of magnitude more memory than CAVE, which stores only $(\mathbf{S}_{c,k}, N_{c,k})$.

S.Table 9: Simple alternatives to DP-vMF + MAS on M-OWODB Task 4 (known AP_{50} and peak memory of the adaptation state).

Method	Detail	Known mAP	Peak memory
Zero-shot (text scores only)	–	56.5	–
Single centroid	all 394k points	56.9	0.3 MB
K-means (K=2 / 3 / 5)	all 394k points	56.9 / 56.9 / 56.9	241 MB
DBSCAN [6]	$\epsilon = 0.05$, all 394k points	56.6	1,483 MB
HDBSCAN [3]	1,000 points per class	56.8	267 MB
FINCH [29]	20,000 points per class	56.9	4,447 MB
Linear probe	cosine to the probe direction	56.5	–
kNN	70 exemplars per class	57.4	23 MB
DP-vMF	–	56.9	2.0 MB
DP-vMF + MAS (CAVE)	–	**57.8**	2.0 MB

F.3 Unknown Ground Truth and U-mAP

As new classes become known, fewer classes remain unknown, so the unknown ground truth shrinks with each task. In addition, many unannotated unknown objects are still detected in practice and counted as false positives, which keeps U-mAP low for all methods. S.Table 10 shows that the number of unknown ground-truth boxes falls by 62% from Task 1 to Task 3, while CAVE's U-mAP falls by 48%; the decline in U-mAP therefore mainly reflects the smaller unknown set, and CAVE keeps more than twice the U-mAP of OW-OVD at every task.

S.Table 10: Unknown ground truth and U-mAP on the M-OWODB test split.

	Task 1	Task 2	Task 3
Unknown classes	60	40	20
Unknown ground-truth boxes	15,831	10,865	6,039
U-mAP, OW-OVD [36]	8.6	4.3	4.0
U-mAP, CAVE	**18.2**	**14.9**	**9.4**

F.4 Per-Class Contribution of the Two Embeddings

Table 2 of the main paper reports class averages, which hide how differently the two embeddings behave across classes. S.Table 11 compares attribute-only and visual-only prototypes class by class at Task 4 over all 80 known classes, counting a class as tied when the two are within 0.05. On both benchmarks, each embedding leads on a substantial and largely disjoint set of classes, so the two are complementary, and S.Table 12 shows that their combination is best on both. Since the unknown score depends on the known side only through the factor $1 - \max_c P_{\text{fused}}$, unknown recall is the same to within 0.1 across the three compositions (S.Table 13).

S.Table 11: Per-class comparison of attribute-only and visual-only prototypes at Task 4 (known AP_{50} margin).

		Attribute stronger	Visual stronger	Tied
M-OWODB	Classes	51	24	5
	Mean margin	+1.6	+1.0	–
	Largest margins	*sports ball* +8.7, *book* +6.7	*apple* +4.4, *kite* +3.1	–
V3-OWODB	Classes	41	20	19
	Mean margin	+3.9	+0.8	–
	Largest margins	*bowling* +37.4, *coscoroba* +18.0	*whisky* +5.3, *vodka* +2.5	–

S.Table 12: Task 4 known AP_{50} by embedding composition.

	Zero-shot	Attribute	Visual	**Unified**
M-OWODB	56.5	57.6	56.8	**57.8**
V3-OWODB	10.0	12.8	11.1	**13.2**

S.Table 13: U-Recall on M-OWODB by embedding composition.

	Task 1	Task 2	Task 3
Attribute only	75.8	79.6	81.1
Visual only	75.7	79.6	81.0
Unified	75.7	79.6	81.0

F.5 Adaptation on Predicted Boxes

The main experiments adapt on the dataset's ground-truth boxes, as prior OWOD work does. To test adaptation without them, the detector is run over the training split, every predicted box above 0.1 confidence is kept as predicted, and the ground truth is used only to assign a class to each kept box, which is what a user labeling the model's own detections would supply. As shown in S.Table 14, CAVE stays within 0.5 known mAP of the ground-truth setting at every task, so it does not depend on ground-truth boxes.

S.Table 14: Known AP_{50} on M-OWODB when CAVE adapts on ground-truth *vs.* predicted boxes.

Adaptation boxes	Task 1	Task 2	Task 3	Task 4
Ground truth (main paper)	75.1	65.8	61.3	57.8
Predicted	74.8	65.6	60.8	57.7

F.6 Further Notes on Scoring

PPMI and unknown recall. In table 7 of the main paper, adding PPMI nearly doubles U-mAP while U-Recall stays within 0.1. Common attributes such as *round* and *black* fire on almost any object and co-occur with everything by chance; PPMI removes these chance associations from the propagation graph, which raises U-mAP from 10.6 to 18.2 at Task 1 and from 4.3 to 9.4 at Task 3.

Robustness of the calibration. The z-score calibration (eq. (12)) does not estimate a per-class distribution; it is a single affine map, computed over all proposals and known classes in the image, that places the MAS logits on the range of the text logits. It therefore leaves the ordering of MAS scores unchanged and does not depend on how many proposals match a given class.

G Additional Benchmarks

G.1 V3-OWODB

To evaluate CAVE beyond COCO categories, we build V3-OWODB from V3Det [34], a detection dataset with 13,204 categories that do not overlap COCO's and are fine-grained (*e.g.*, *daisy*, *red paper lantern*, *rollerball pen*). It introduces 80 known classes over four tasks, and every V3Det category outside the known classes remains unknown, so unknown detection is also evaluated at Task 4. Results are reported in table 8 of the main paper; CAVE improves known mAP over the zero-shot VLOD at every task while keeping U-Recall above 68.

G.2 RS-OWODB

To test imagery other than natural photographs, we build RS-OWODB on the optical remote-sensing dataset DIOR [14], following the M-OWODB protocol. The 20 DIOR classes are divided into four tasks of five classes each (S.Table 15); at each task, only the classes introduced so far are known and every remaining class is evaluated as unknown. The four tasks share a single test set of 11,738 images.

S.Table 15: RS-OWODB task split.

Task	Known classes introduced	Train images
T1	baseball field, dam, ground track field, stadium, vehicle	5,394
T2	basketball court, expressway service area, harbor, storage tank, windmill	3,445
T3	airplane, bridge, expressway toll station, overpass, tennis court	4,111
T4	airport, chimney, golf field, ship, train station	3,247

Since the YOLO-World detector used throughout the paper is pretrained on natural images, we pretrain a base detector with the same architecture on LAE-1M [23], a remote-sensing detection dataset, using AdamW with a learning rate of 5×10^{-4}, weight decay of 0.02, and batch size of 64 for 40 epochs. CAVE uses the hyperparameters of the main paper; only the superclass entries of the attribute vocabulary are replaced with domain terms (*aerial object*, *ground object*, *sea object*, *industrial structure*, *field planar*, *site infrastructure*). As reported in table 8 of the main paper, CAVE improves Task 4 known mAP by +2.6 and reaches 84–88 U-Recall, comparable to the main benchmarks.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state our three contributions (training-free OWOD formulation, CAVE with MAS for known detection, AOS for unknown detection) and the quantitative claims (+16.2 known mAP, +28.0 U-Recall on M-OWODB), all supported by experiments in Sec. 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: A dedicated *Limitations* paragraph in the conclusion (Sec. 5) discusses the dependence on VAW/GoldG-derived attribute vocabulary and reduced effectiveness on out-of-distribution domains (e.g., medical, remote sensing) without domain-specific vocabulary curation.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper presents an empirical method without formal theoretical results requiring proofs.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Sec. 4.1 and Sec. B of the supplementary fully specify the base detector (YOLO-World-L, CLIP-ViT-B/32), hyperparameters ($\beta = 0.30$, $\tau = 0.40$), benchmarks (M-OWODB, S-OWODB with per-task class lists), and pipeline modifications (frozen weights, learning rate 0, batch size 16, single forward pass). The attribute vocabulary construction is detailed in Sec. A of the supplementary.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: All datasets used (MS COCO, PASCAL VOC, VAW, GoldG, V3Det, DIOR, LAE-1M) and pre-trained models (YOLO-World, CLIP, Grounding DINO) are publicly available; the method, hyperparameters, and pipeline modifications described in the paper and supplementary are sufficient to reimplement CAVE on top of MMDetection. Code will be released upon publication.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Sec. 4.1 specifies hyperparameters ($\beta = 0.30$, $\tau = 0.40$ chosen via the sweeps in Tab. 3 and Tab. 4) and the frozen-detector training-free regime (learning rate 0, 1 epoch). Sec. B of the supplementary covers data splits, post-processing (NMS, score thresholds), assigner configuration, and batch size.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: CAVE is training-free with a single deterministic forward pass over the labeled data; there is no random initialization or optimization to introduce variance. Single-run results are reported as in prior OWOD work [11, 8, 36], and the magnitude of improvements over baselines (e.g., +16.2 known mAP, +28.0 U-Recall) far exceeds typical run-to-run variability.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Sec. 4.1 reports a single RTX 4090 GPU and $\sim$76 s adaptation time. Sec. D of the supplementary provides a detailed cost analysis (per-task timing, VRAM usage of 2,498 MB, additional memory of $\sim$ 2.3 MB, inference at 88.3 FPS) and compares against the OW-OVD baseline.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses publicly available datasets (MS COCO, PASCAL VOC, V3Det, DIOR, LAE-1M) and pre-trained models (YOLO-World, CLIP, Grounding DINO) under their original licenses and conforms to the NeurIPS Code of Ethics. Bias-sensitive descriptors (hair color, skin color, race) and person-specific activity phrases are explicitly excluded from the attribute vocabulary (Sec. A of the supplementary).

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The paper presents foundational research on training-free open-world object detection with no direct path to specific applications. Potential positive impacts include enabling on-device adaptation in privacy-sensitive or resource-constrained settings (motivated in Sec. 1); we do not foresee specific negative impacts beyond those generic to object detection.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release new high-risk models, datasets, or generative components.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All used assets are cited in their original publications: YOLO-World [5], CLIP [28], Grounding DINO [19], MS COCO [17], PASCAL VOC [7], VAW [27], GoldG [15], V3Det [34], DIOR [14], LAE-1M [23], MMDetection [4]. We use these assets within their published licenses for academic research.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: No new datasets or pre-trained models are released. The attribute vocabulary and CAVE adaptation code, when released upon publication, will be documented with the construction procedure already detailed in Sec. A of the supplementary.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper involves no crowdsourcing or human-subject research.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper involves no human-subject research, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: No LLM is used in the core method. CAVE relies on a frozen Vision-Language Object Detector (YOLO-World/CLIP, or Grounding DINO/BERT) and constructs the attribute vocabulary via spaCy-based dependency parsing of GoldG captions (Sec. A of the supplementary), not via LLM generation.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.