Exploring Hierarchical Consistency and Unbiased Objectness for Open-Vocabulary Object Detection

Supplementary Material

In this supplementary material, we provide more details (Sec. S1), discussions (Sec. S2), results (Sec. S3) and limitations and future work (Sec. S4).

S1. More details

In this section, we provided detailed descriptions for our experimental setup for obtaining results in Fig. 1 (Sec. S1.1), and proof/analysis for Eqs. (7) and (8) (Sec. S1.2). We then describe the training details for LoCLIP (Sec. S1.3), and a process for refining a LLM-generated hierarchy (Sec. S1.4).

S1.1. Experimental setup for Fig. 1

Our approach is based on the observation that regions that well-localize an object tend to yield hierarchically consistent predictions, while irrelevant ones (*e.g.*, misaligned and background regions) do not. To demonstrate this observation, we leverage the class names and the ground-truth bounding box annotations of the base object classes in OV-COCO [19]. Specifically, we first establish a semantic hierarchy (*i.e.*, super- and sub-categories) for the base object classes using GPT-OSS-120b [1] as the LLM. For example, this process yields *picnic bench* as a sub-category of the class *bench*, which itself belongs to the super-category *furniture*. We then gather object proposals from an RPN [26] and categorize them based on their IoU with the ground-truth boxes. The proposals with an IoU score higher than 0.8 are labeled as foreground (well-localized), while those with an IoU score lower than 0.5 are labeled as background (irrelevant). We use this experimental setup to visualize the hierarchical consistency for well-localized and irrelevant regions separately in Fig. 1. Note that this analysis is performed using the class names and the ground-truth annotations for base object classes only. We provide in Fig. S2 more visualizations and quantitative analysis in Sec. S3.

S1.2. Eqs. (7) and (8)

Proof of Eq. (7). We wish to show that

$$\arg\max(\mathbf{p}) = \arg\max(\mathbf{z}_{\text{sub}}) \;\Rightarrow\; \max(\mathbf{r}_{\text{sub}}) \geq \hat{p}, \tag{S1}$$

where $\hat{p} = \max(\mathbf{p})$. Let l be the common index of the maximum elements, defined as:

$$l = \arg\max(\mathbf{p}) = \arg\max(\mathbf{z}_{\text{sub}}). \tag{S2}$$

Invoking the generalized mean inequality, specifically for orders set to 1 and ∞, we have:

$$\sum_m \mathbf{p}(m)\mathbf{z}_{\text{sub}}(m) \leq \max(\mathbf{z}_{\text{sub}}) = \mathbf{z}_{\text{sub}}(l). \tag{S3}$$

Applying Eq. (S3) to the definition of the l-th element of $\mathbf{r}_{\text{sub}}$, we obtain:

$$\begin{aligned}\mathbf{r}_{\text{sub}}(l) &= \frac{\mathbf{p}(l)\mathbf{z}_{\text{sub}}(l)}{\sum_m \mathbf{p}(m)\mathbf{z}_{\text{sub}}(m)} \\ &\geq \frac{\mathbf{p}(l)\mathbf{z}_{\text{sub}}(l)}{\mathbf{z}_{\text{sub}}(l)} \\ &= \mathbf{p}(l).\end{aligned} \tag{S4}$$

Here, Eq. (S4) holds because replacing the denominator $\sum_m \mathbf{p}(m)\mathbf{z}_{\text{sub}}(m)$ with its upper bound $\mathbf{z}_{\text{sub}}(l)$ yields a lower bound for the fraction. Finally, since $\mathbf{p}(l) = \hat{p}$, it follows that

$$\max(\mathbf{r}_{\text{sub}}) \geq \hat{p}. \tag{S5}$$

Property in Eq. (8). In the following, we demonstrate that Eq. (8) holds under two conditions: 1) $\max(\mathbf{p})$ is sufficiently large, *e.g.*, exceeding 0.5, and 2) the score distribution of $\mathbf{z}_{\text{sub}}$ is sufficiently flat. We further show that these conditions can be enforced into our framework with minimal effort.

Let $l = \arg\max(\mathbf{p})$ denote the index of the maximum element in $\mathbf{p}$. We characterize the flatness of $\mathbf{z}_{\text{sub}}$ by bounding the ratio between its maximum and minimum values with a small constant $\beta \geq 1$ as follows:

$$1 \leq \frac{\max(\mathbf{z}_{\text{sub}})}{\min(\mathbf{z}_{\text{sub}})} \leq \beta. \tag{S6}$$

To prove that $\max(\mathbf{r}_{\text{sub}}) < \max(\mathbf{p})$, we analyze two cases. First, we examine the value of $\mathbf{r}_{\text{sub}}$ at the index l. Under the inconsistency assumption (*i.e.*, $\arg\max(\mathbf{z}_{\text{sub}}) \neq l$), the element $\mathbf{z}_{\text{sub}}(l)$ is strictly less than the maximum of $\mathbf{z}_{\text{sub}}$. Given the condition in Eq. (S6) and $\mathbf{z}_{\text{sub}}(l) < \max(\mathbf{z}_{\text{sub}})$, the mean $\sum_m \mathbf{p}(m)\mathbf{z}_{\text{sub}}(m)$ strictly exceeds the value $\mathbf{z}_{\text{sub}}(l)$. That is,

$$\mathbf{r}_{\text{sub}}(l) = \mathbf{p}(l)\left(\frac{\mathbf{z}_{\text{sub}}(l)}{\sum_m \mathbf{p}(m)\mathbf{z}_{\text{sub}}(m)}\right) < \mathbf{p}(l). \tag{S7}$$

Second, for any index $n \neq l$, we can represent $\mathbf{r}_{\text{sub}}(n)$ as follows:

$$\mathbf{r}_{\text{sub}}(n) = \mathbf{p}(n)\frac{\mathbf{z}_{\text{sub}}(n)}{\sum_m \mathbf{p}(m)\mathbf{z}_{\text{sub}}(m)}. \tag{S8}$$

Note that the scaling factor $\frac{\mathbf{z}_{\text{sub}}(n)}{\sum_m \mathbf{p}(m)\mathbf{z}_{\text{sub}}(m)}$ is strictly bounded by β, since $\sum_m \mathbf{p}(m)\mathbf{z}_{\text{sub}}(m) \geq \min(\mathbf{z}_{\text{sub}})$.

Given the condition that $\mathbf{p}(l)$ is sufficiently large, implying $\mathbf{p}(n)$ is small, it follows that

$$\mathbf{p}(n) \cdot \beta < \mathbf{p}(l). \tag{S9}$$

Since β is close to 1, this condition holds naturally for any $\max(\mathbf{p}) > 0.5$, implying $\mathbf{r}_{\text{sub}}(n) < \max(\mathbf{p})$. Combining both cases, we can conclude that $\max(\mathbf{r}_{\text{sub}}) < \max(\mathbf{p})$.

In practice, we find that applying a minimum threshold to $\hat{p}$ is sufficient to ensure that Eq. (8) holds. This is because $\mathbf{z}_{\text{sub}}$ and $\mathbf{z}_{\text{sup}}$ are derived by max-pooling the probability distribution computed over the entire set of sub- and super-categories, respectively. Since the softmax operation is applied across a large number of sub-/super-categories, the score is inherently distributed, resulting in a flat distribution. Consequently, explicitly constraining the flatness is unnecessary, and enforcing a minimum threshold on $\hat{p}$ as a prerequisite for HCC is sufficient to guarantee the properties in Eqs. (7)-(8). In our experiments, we adopt a threshold of 0.5, and candidate regions with lower confidence scores $\hat{p}$ are simply discarded in the pseudo labeling process. Note that enforcing the score distributions of $\mathbf{z}_{\text{sub}}$ and $\mathbf{z}_{\text{sup}}$ can be performed by simply applying a multiplicative temperature parameter in Eq. (3) set to the value in a range of $(0, 1)$.

S1.3. LoCLIP training

Dataset. We construct a dedicated dataset for training LoCLIP by extracting candidate regions from an RPN [26]. Each candidate region is assigned a binary label based on its intersection over union (IoU) with any object instance in the ground-truth annotations of the base classes. Specifically, candidate regions with an IoU score greater than 0.8 are labeled as positive, while those with an IoU score of 0.8 or lower are labeled as negative. This process yields a dataset of image patches (*i.e.*, cropped candidate regions) paired with their binary labels. Notably, we empirically observe that LoCLIP converges quickly, and using only 1% of the training split from the COCO [19] or LVIS [8] is sufficient to achieve meaningful predictions. Constructing this dataset thus requires only a negligible computational overhead.

Loss. The candidate regions in our curated dataset are typically imbalanced, with background regions substantially outnumbering foreground regions. Directly training LoCLIP on this imbalanced dataset tends to induce a strong bias toward the background class, leading to objectness scores that are frequently close to the value of 0. To mitigate this issue, we adopt a weighted binary cross-entropy loss, using the ratios of foreground and background regions as weights for the positive and negative terms, respectively. This helps LoCLIP training by suppressing bias toward the background class in its predictions.

Training. We train LoCLIP for approximately 8k iterations using a cosine learning rate scheduler, with the maximum learning rate set to 0.001. The parameters are optimized with SGD using a batch size of 192 on a single NVIDIA A6000 GPU. We note that, unlike previous works [31, 32, 37, 38] which estimate the objectness of candidate regions directly from the RPN, our approach requires an additional training step for LoCLIP. However, this extra step takes no more than 5 minutes, which is negligible in practice, while providing significantly more reliable objectness estimations for unseen object classes (Table 4).

Hyperparameter τ. We select τ by evaluating the binary classification accuracy on the dataset used for training LoCLIP. We perform a grid search for τ in the range $[0.1, 0.9]$ with a 0.1 interval, and choose the value that yields the best accuracy. While a more extensive approach to searching for the optimal τ could be used, *e.g.*, by curating a dedicated validation set, we found this simple method to be sufficient.

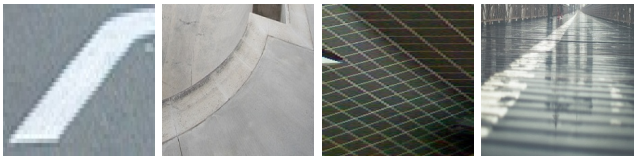
S1.4. LLM hierarchy refinement

We establish an initial LLM-generated hierarchy by querying LLMs to provide K super- and sub-categories of a given object class [20, 24]. Directly exploiting LLM responses, however, can introduce incorrect entries into the generated hierarchical structure, as LLMs are prone to make mistakes. We point out that previous methods [20, 24] that use LLMs for establishing a hierarchy structure between object classes, have typically been limited to using only a small number of K for establishing the hierarchy, *e.g.*, 3 in [20]. On the one hand, using a larger K allows to consider the various super-/sub-categories and helps to capture diverse semantic relationships. On the other hand, it could introduce incorrect concepts in the LLM-generated hierarchy, since the risk of LLM providing incorrect response increases as K grows. To address this problem, we additionally refine the initial super-/sub-categories through an LLM-driven process based on three main criteria: 1) correctness, 2) discriminability, and 3) near-duplicate removal. We describe in the following each step in detail. Note that applying these processes may result in a varying number of super- and sub-categories per class and we have assumed in Eqs. (3) and (4) a uniform number per class for notational simplicity.

Correctness. For each entry given by the LLM, we ask the LLM if the given entry is indeed correct, *i.e.*, if the given association (*e.g.*, super- or sub-ordinate relationships) holds, using the "Is-A" relationship, which is the foundation of the WordNet database [23]. For example, if a *golden retriever* is given as a sub-category of *dog*, we ask the LLM: Is *golden retriever* a *dog*? Similarly, when *companion animal* is given as a super-category of *dog*, we ask the LLM: Is *dog* a *companion animal*? Although exploiting the same model

"knife" "skateboard" "tie" "umbrella"

(a) Misaligned regions



"knife" "skateboard" "tie" "umbrella"

(b) Background regions

Figure S1. Examples of candidate regions for novel classes in COCO [19]. We show regions whose confidence scores predicted using CLIP [25] exceeds 0.9, together with the estimated class labels.

for both generation and verification may seem redundant, this self-validation step serves as an effective filtering mechanism for incorrect entries. This observation coincides with the self-consistency strategy [33], which suggests that LLMs are capable of refining their own outputs to achieve better performance.

Discriminability. While many entries in the LLM-generated hierarchy could provide useful cues for discriminating objects of different classes, some might negatively affect the process by being distractive. This is particularly the case for super-categories, since many distinct objects can share a common, overly broad super-category, making it uninformative. For example, the concept *entity* is a valid super-category shared across all classes, but it provides no useful discriminatory cues. To alleviate this, we discard super-category entries that are shared by a disproportionately large number of object classes, *e.g.*, more than $1/3$ of the total number of novel object classes on OV-COCO [19].

Near-duplicate removal. Lastly, we remove near-duplicates. We compute the cosine similarity between entries in the CLIP embedding space and discard one entry from any pair found to be too close. This process removes, *e.g.*, simple singular/plural variations of the same concept, such as *home appliance* and *home appliances*.

We visualize in Fig. S4 our super- and sub-categories of novel object classes in OV-COCO [19] obtained using GPT-OSS-120b [1] as a LLM where K is initially set to 30 and 10 for sub- and super-categories, respectively.

S2. More discussions

Limitations of CLIP confidence score. Standard VLMs employed in OVD, such as CLIP [25] and ALBEF [15], are pre-trained for image-level tasks, leading to a discrepancy when applied to region-level predictions in OVD. Consequently, their confidence scores do not reliably reflect the presence of novel object classes in candidate regions. To demonstrate this limitation, we show in Fig. S1 the candidate regions whose confidence scores $\hat{p}$ exceeds 0.9, but correspond to misaligned (Fig. S1 (a)) or correspond to the background (Fig. S1 (b)). We can see that directly using the confidence scores $\hat{p}$ in the context of pseudo labeling is suboptimal, resulting in the generated pseudo labels to be dominated by irrelevant regions [31].

Is confidence adjustment in HCC necessary? With the class-wise sub-category scores $\mathbf{z}_{\text{sub}}$ and super-category scores $\mathbf{z}_{\text{sup}}$ in hand, a direct formulation of hierarchical consistency involves verifying whether the class estimations across all levels are consistent. Specifically, we can assign a pseudo class label $\hat{y}_b$ for the region b as follows:

$$\hat{y}_b = \arg\max(\mathbf{p}) \text{ if } \arg\max(\mathbf{p}) = \arg\max(\mathbf{z}_{\text{sub}}) = \arg\max(\mathbf{z}_{\text{sup}}). \tag{S10}$$

This strategy, however, is sub-optimal as it relies solely on the top-ranking indices (*i.e.*, $\arg\max$), thereby discarding the rich statistical information (*e.g.*, likelihood values) encoded in the full distributions of $\mathbf{z}_{\text{sub}}$, $\mathbf{z}_{\text{sup}}$, and $\mathbf{p}$. By contrast, HCC exploits these full distributions to calibrate the confidence scores, allowing for a more robust and fine-grained estimation of object presence. Quantitatively, the strategy defined in Eq. (S10) achieves 35.5 AP_{50}^{N}, which improves upon the baseline (32.2 AP_{50}^{N}) but is significantly outperformed by our HCC technique (37.8 AP_{50}^{N}), confirming the efficacy of our calibration approach.

Reproducibility of MarvelOVD [31]. In Sec. 4.2, we report the performance of MarvelOVD [31] using our re-implementation. Note that although the official code is publicly available[1], we found that it fails to execute due to a fundamental error. We attempted to contact the authors regarding this issue but did not receive a response. Therefore, the results reported in Sec. 4.2 are obtained exclusively from our re-implementation for MarvelOVD [31], which is based on the official code provided by the authors.

S3. More results

More results on hierarchical consistency. We provide additional visualizations of hierarchical consistency in

[1] https://github.com/wkfdb/MarvelOVD

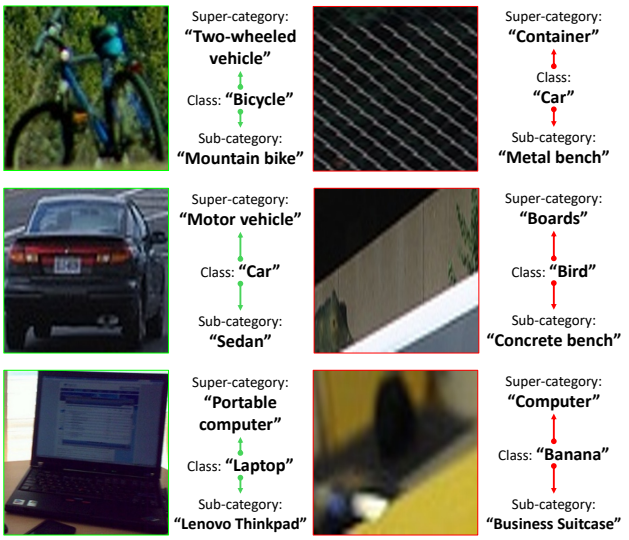


Figure S2. Visualization of hierarchical consistency of candidate regions. We group the candidate regions into well-localized (left) and irrelevant (right) regions based on their IoU with the ground-truth bounding boxes for base object classes. Green and red arrows indicate consistent and inconsistent hierarchical associations, respectively. See text for details.

Fig. S2. We can see that well-localized regions tend to yield hierarchically consistent predictions, whereas irrelevant ones do not, which aligns with the observations in Fig. 1. We further perform quantitative analysis and observe that 88.9% of the well-localized region yield hierarchically consistent predictions, while only 11.2% of the irrelevant regions do. The results demonstrate a significant gap in consistency ratios between the two groups, suggesting that hierarchical consistency could serve as a valuable cue for discriminating valid objects from the background.

Comparison between RPN and LoCLIP. The RPN [26], commonly used in previous works [31, 32, 37] for estimating the objectness of candidate regions, is sub-optimal due to its inherent bias towards base object classes. Consequently, it fails to provide reliable objectness scores for unseen object classes. To address this limitation, we introduce LoCLIP, which employs a parameter-efficient fine-tuning strategy to estimate objectness. We present qualitative comparisons of objectness scores from the RPN and LoCLIP in Fig. S3. The visualizations clearly demonstrate the advantages of LoCLIP over the RPN. For example, for novel object instances shown in the left column (*e.g.*, *bus*, *knife*, *dog* and *cow*), the RPN yields low objectness scores, failing to identify them as foreground. In contrast, LoCLIP successfully assigns high scores to these regions. Conversely, for the irrelevant background regions in the right column, the RPN tends to produce inappropriately high scores, whereas LoCLIP effectively suppresses them, demonstrating its advantage as a reliable objectness score metric.

Table S1. Performance comparison between models trained with pseudo labels obtained using different LLMs on OV-COCO [19].

LLM	AP_{50}^{N}	AP_{50}^{B}
-	32.2	58.3
LLaMa-3-8B [6]	36.2	59.2
Qwen-3-30b [30]	<u>37.4</u>	59.4
GPT-OSS-120b [1]	**37.8**	59.4

Using different LLMs. We primarily use GPT-OSS-120b [1] as a LLM throughout all experiments in the main paper. To further demonstrate the generalization capability of our approach, we employ various LLMs and compare the results. Specifically, we establish hierarchical structure of novel object classes by extracting super-/sub-categories using LLaMa-3.1-8B [6] and Gemma-3-27b [30]. We present the results in Table S1. We can see that all entries in Table S1 outperform a baseline that does not exploit the HCC technique, demonstrating that our work can work well with various LLMs. We point out that while it is possible to construct accurate hierarchies for each dataset manually, doing so requires considerable domain expertise and effort, and such hierarchies would need to be rebuilt for every dataset in use. In contrast, LLMs provide an efficient, scalable way to automatically generate such hierarchies, significantly reducing human-driven efforts. In our work, we use LLMs for this purpose, not as a core contribution of our method, but as a practical means to obtain a reasonable hierarchy that our framework can exploit to improve pseudo label generation for OVD. Our framework assumes a reasonable hierarchical structure, rather than a perfect one. This aligns with results reported in prior works, despite the imperfect hierarchical structure, have successfully used LLM-generated hierarchies to improve performance in image classification [24] and object detection [20].

Comprehensive comparison with the state-of-the-art methods. Tables S2 and S3 present comprehensive comparisons with state-of-the-art methods on OV-COCO [19] and OV-LVIS [8], respectively. For completeness, we include methods that leverage auxiliary sources of supervision and specify additional datasets, and pretrained models used. The results demonstrate that our method outperforms other approaches that do not utilize additional datasets. Ours even outperforms previous works that employ DETR [42] as a detector [16, 36]. Generally, methods exploiting external datasets and pretrained models (*e.g.*, MViT [22] as a proposal generator) outperform ones that do not. We note that in the OVD setting, it is standard practice to

distinguish between these two paradigms, as they operate under fundamentally different assumptions; thus, a direct comparison would be unfair. In this context, exploring how to incorporate such additional datasets into our framework remains an interesting direction for future work.

Visualization of pseudo labels. We visualize in Fig. S5 the pseudo labels generated with and without the HCC and LoCLIP components. Compared to the baseline in the left-most column, we observe that HCC discards irrelevant regions (*e.g.*, redundant boxes labelled as *dog* and incorrect class assignments as *cat* in the top row), while LoCLIP filters out severely misaligned regions (*e.g.*, an excessively large box labelled as *skateboard* in the top row). Similar observations can be made for the *cow* object in the second row. Overall, these two components complement each other, effectively preventing irrelevant regions from being assigned pseudo labels. We also present a failure case in the last row, where the tail region of an airplane is mislabelled as an *airplane*; neither HCC or LoCLIP was able to discard this noisy label. Nonetheless, our method still yields significantly less noisy pseudo labels compared to the baseline.

S4. Limitations and future work

Although our approach is effective and sets a new state of the art, it shares limitations in other OVD methods using pseudo labels [31, 32, 37, 38]. First, we use class names of novel object classes to obtain accurate class label predictions from VLMs, which may not be available in real-world scenarios. Second, particularly on the OV-LVIS dataset [8], our method is outperformed by OVD methods exploiting additional source of supervision [2, 4, 11, 17, 18, 21, 41], *e.g.*, images with corresponding class labels [27] or captions [3, 28]. Exploring how to incorporate these additional datasets into a pseudo labeling framework would be an interesting direction for future work.

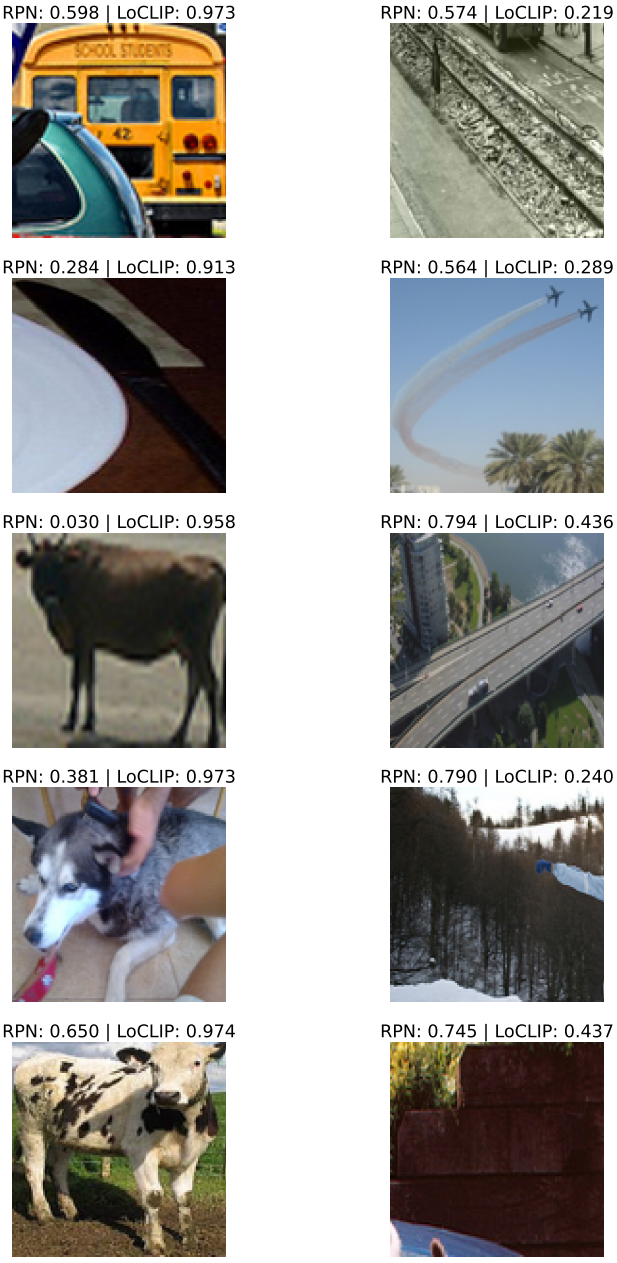
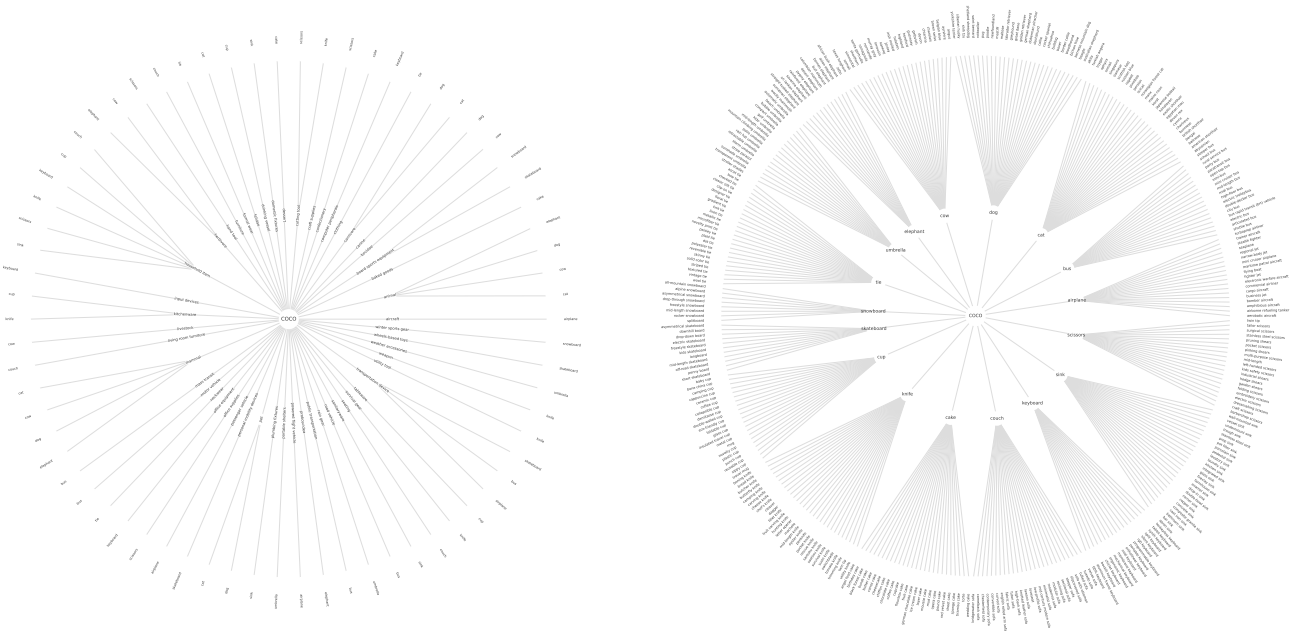


Figure S3. Qualitative comparison of objectness scores estimated by the RPN [26] and LoCLIP. LoCLIP consistently yields higher scores for valid objects and lower scores for background regions compared to the RPN, demonstrating more reliable and unbiased objectness estimations. See text for details.



(a) Super-categories for novel object classes on OV-COCO [19]

(b) Sub-categories for novel object classes on OV-COCO [19]

Figure S4. Super-/sub-categories for novel object classes in OV-COCO [19]. We establish a hierarchical structure between object classes by leveraging GPT-OSS-120b [1] as a LLM. Best viewed by zooming in.

Table S2. Comprehensive comparison with state-of-the-art methods on the OV-COCO dataset [19]. We report the mean and standard deviation across 3 independent runs. † indicates results from our re-implementation.

Method	Backbone	Detector Architecture	Additional Supervision	Additional Pretrained Models	AP_{50}^{N}	AP_{50}^{B}
ViLD [7]	RN50 [9]	FR-CNN [26]			27.6	59.5
F-VLM [14]	RN50 [9]	FR-CNN [26]			28.0	40.2
OV-DETR [36]	RN50 [9]	DETR [42]			29.4	52.7
OADP[1] [32]	RN50 [9]	FR-CNN [26]			31.3	-
VL-PLM [37]	RN50 [9]	FR-CNN [26]			32.3	54.0
DK-DETR [16]	RN50 [9]	DETR [42]			32.3	61.1
RALF [13]	RN50 [9]	FR-CNN [26]			33.4	54.5
BARON [34]	RN50 [9]	FR-CNN [26]			34.0	60.4
MarvelOVD† [31]	RN50 [9]	FR-CNN [26]			35.4	56.5
SAS-Det [38]	RN50 [9]	FR-CNN [26]		RegionCLIP [39]	37.4	58.5
Ours	RN50 [9]	FR-CNN [26]			**38.9**$_{\pm 0.3}$	59.5$_{\pm 0.2}$
Detic [41]	RN50 [9]	FR-CNN [26]	Captions [3]		27.8	47.1
DV-Det [12]	RN50 [9]	FR-CNN [26]	Captions [3]		35.8	57.0
OC-OVD [2]	RN50 [9]	FR-CNN [26]	Captions [3]	MViT [22]	36.6	54.0
RALF [13]	RN50 [9]	FR-CNN [26]	Captions [3]	MViT [22]	41.3	54.3
CLIFF [17]	RN50 [9]	FR-CNN [26]	Captions [3]	MViT [22]	41.3	54.1
BARON [34]	RN50 [9]	FR-CNN [26]	Captions [3]	MViT [22]	**42.7**	54.9
CLIPSelf† [35]	ViT-L/14 [29]	F-ViT [35]			41.3	65.5
Ours	ViT-L/14 [29]	F-ViT [35]			**44.0**$_{\pm 0.2}$	65.8$_{\pm 0.1}$

Table S3. Comprehensive comparison with state-of-the-art methods on the OV-LVIS dataset [19]. We report the mean and standard deviation across 3 independent runs. † indicates results from our re-implementation.

Method	Backbone	Detector Architecture	Additional Supervision	Additional Pretrained Models	AP_{m}^{N}	AP_{m}^{All}
OV-DETR [36]	RN50 [9]	DETR [42]			17.4	26.6
F-VLM [14]	RN50 [9]	Mask R-CNN [10]			18.6	24.2
BARON [34]	RN50 [9]	Mask R-CNN [10]			19.2	26.5
OADP[1] [32]	RN50 [9]	Mask R-CNN [10]			19.9	-
DK-DETR [16]	RN50 [9]	DETR [42]			20.5	30.0
Ours	RN50 [9]	Mask R-CNN [10]			**21.7**$_{\pm 0.4}$	26.0$_{\pm 0.2}$
VL-Det [18]	RN50 [9]	CenterNet2 [40]	CC3M [28]		21.7	30.1
DV-Det [12]	RN50 [9]	CenterNet2 [40]	CC3M [28]		23.1	31.2
CLIFF [17]	RN50 [9]	CenterNet2 [40]	IN-L [41]	MViT [22]	24.5	28.2
Detic [41]	RN50 [9]	CenterNet2 [40]	IN-L [41]		24.6	32.4
OC-OVD [2]	RN50 [9]	CenterNet2 [40]	IN-L [41]	MViT [22]	**25.2**	32.9
CLIPSelf† [35]	ViT-B/16 [29]	F-ViT [35]			25.1	24.5
Ours	ViT-B/16 [29]	F-ViT [35]			**25.5**$_{\pm 0.2}$	24.7$_{\pm 0.1}$

[1] The results for OADP [32], reported in the original publication, were obtained using LSJ data augmentation [5], which is inconsistent with the experimental setting described in the paper. We thus report the results without the LSJ technique, obtained from the official repository.

(a) HCC ✗, LoCLIP ✗ (b) HCC ✓, LoCLIP ✗ (c) HCC ✗, LoCLIP ✓ (d) HCC ✓, LoCLIP ✓

Figure S5. Visualizations of pseudo labels on OV-COCO [19] with and without the HCC and LoCLIP components. Green and orange boxes represent base and novel object classes, respectively. The usage of each component is marked by ✓ and ✗ . The bottom row shows a failure case. Best viewed in color.

References

[1] Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025. 1, 3, 4, 6

[2] Hanoona Bangalath, Muhammad Maaz, Muhammad Uzair Khattak, Salman H Khan, and Fahad Shahbaz Khan. Bridging the gap between object and image-level representations for open-vocabulary detection. In *NeurIPS*, 2022. 5, 7

[3] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 5, 7

[4] Mingfei Gao, Chen Xing, Juan Carlos Niebles, Junnan Li, Ran Xu, Wenhao Liu, and Caiming Xiong. Open vocabulary object detection with pseudo bounding-box labels. In *ECCV*, 2022. 5

[5] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *CVPR*, 2021. 7

[6] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 4

[7] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. In *ICLR*, 2022. 7

[8] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019. 2, 4, 5

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 7

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. 7

[11] Jiaxing Huang, Jingyi Zhang, Kai Jiang, and Shijian Lu. Open-vocabulary object detection via language hierarchy. *NeurIPS*, 2024. 5

[12] Sheng Jin, Xueying Jiang, Jiaxing Huang, Lewei Lu, and Shijian Lu. LLMs meet VLMs: Boost open vocabulary object detection with fine-grained descriptors. In *ICLR*, 2024. 7

[13] Jooyeon Kim, Eulrang Cho, Sehyung Kim, and Hyunwoo J Kim. Retrieval-augmented open-vocabulary object detection. In *CVPR*, 2024. 7

[14] Weicheng Kuo, Yin Cui, Xiuye Gu, AJ Piergiovanni, and Anelia Angelova. F-VLM: Open-vocabulary object detection upon frozen vision and language models. In *ICLR*, 2023. 7

[15] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *NeurIPS*, 2021. 3

[16] Liangqi Li, Jiaxu Miao, Dahu Shi, Wenming Tan, Ye Ren, Yi Yang, and Shiliang Pu. Distilling detr with visual-linguistic knowledge for open-vocabulary object detection. In *ICCV*, 2023. 4, 7

[17] Wuyang Li, Xinyu Liu, Jiayi Ma, and Yixuan Yuan. CLIFF: Continual latent diffusion for open-vocabulary object detection. In *ECCV*, 2024. 5, 7

[18] Chuang Lin, Peize Sun, Yi Jiang, Ping Luo, Lizhen Qu, Gholamreza Haffari, Zehuan Yuan, and Jianfei Cai. Learning object-language alignments for open-vocabulary object detection. In *ICLR*, 2023. 5, 7

[19] 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*, 2014. 1, 2, 3, 4, 6, 7, 8

[20] Mingxuan Liu, Tyler L Hayes, Elisa Ricci, Gabriela Csurka, and Riccardo Volpi. SHiNe: Semantic hierarchy nexus for open-vocabulary object detection. In *CVPR*, 2024. 2, 4

[21] Chuofan Ma, Yi Jiang, Xin Wen, Zehuan Yuan, and Xiaojuan Qi. CoDet: Co-occurrence guided region-word alignment for open-vocabulary object detection. In *NeurIPS*, 2023. 5

[22] Muhammad Maaz, Hanoona Rasheed, Salman Khan, Fahad Shahbaz Khan, Rao Muhammad Anwer, and Ming-Hsuan Yang. Class-agnostic object detection with multi-modal transformer. In *ECCV*, 2022. 4, 7

[23] George A Miller. WordNet: a lexical database for english. *ACM*, 1995. 2

[24] Zachary Novack, Julian McAuley, Zachary Chase Lipton, and Saurabh Garg. CHiLS: Zero-shot image classification with hierarchical label sets. In *ICML*, 2023. 2, 4

[25] 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*, 2021. 3

[26] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *TPAMI*, 2016. 1, 2, 4, 5, 7

[27] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. ImageNet-21K pretraining for the masses. In *NeurIPS*, 2021. 5

[28] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018. 5, 7

[29] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. EVA-CLIP: Improved training techniques for CLIP at scale. *arXiv preprint arXiv:2303.15389*, 2023. 7

[30] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025. 4

[31] Kuo Wang, Lechao Cheng, Weikai Chen, Pingping Zhang, Liang Lin, Fan Zhou, and Guanbin Li. Marvelovd: Marrying object recognition and vision-language models for robust open-vocabulary object detection. In *ECCV*, 2024. 2, 3, 4, 5, 7

[32] Luting Wang, Yi Liu, Penghui Du, Zihan Ding, Yue Liao, Qiaosong Qi, Biaolong Chen, and Si Liu. Object-aware distillation pyramid for open-vocabulary object detection. In *CVPR*, 2023. 2, 4, 5, 7

[33] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *ICLR*, 2023. 3

[34] Size Wu, Wenwei Zhang, Sheng Jin, Wentao Liu, and Chen Change Loy. Aligning bag of regions for open-vocabulary object detection. In *CVPR*, 2023. 7

[35] Size Wu, Wenwei Zhang, Lumin Xu, Sheng Jin, Xiangtai Li, Wentao Liu, and Chen Change Loy. ClipSelf: Vision transformer distills itself for open-vocabulary dense prediction. In *ICLR*, 2024. 7

[36] Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy. Open-vocabulary DETR with conditional matching. In *ECCV*, 2022. 4, 7

[37] Shiyu Zhao, Zhixing Zhang, Samuel Schulter, Long Zhao, BG Vijay Kumar, Anastasis Stathopoulos, Manmohan Chandraker, and Dimitris N Metaxas. Exploiting unlabeled data with vision and language models for object detection. In *ECCV*, 2022. 2, 4, 5, 7

[38] Shiyu Zhao, Samuel Schulter, Long Zhao, Zhixing Zhang, Yumin Suh, Manmohan Chandraker, Dimitris N Metaxas, et al. Taming self-training for open-vocabulary object detection. In *CVPR*, 2024. 2, 5, 7

[39] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, et al. RegionCLIP: Region-based language-image pretraining. In *CVPR*, 2022. 7

[40] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Probabilistic two-stage detection. *arXiv preprint arXiv:2103.07461*, 2021. 7

[41] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *ECCV*, 2022. 5, 7

[42] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *ICLR*, 2021. 4, 7