Open-Vocabulary Object Detection via Language Hierarchy

Jiaxing Huang, Jingyi Zhang, Kai Jiang, Shijian Lu[*]
College of Computing and Data Science
Nanyang Technological University, Singapore

Abstract

Recent studies on generalizable object detection have attracted increasing attention with additional weak supervision from large-scale datasets with image-level labels. However, weakly-supervised detection learning often suffers from image-to-box label mismatch, i.e., image-level labels do not convey precise object information. We design Language Hierarchical Self-training (LHST) that introduces language hierarchy into weakly-supervised detector training for learning more generalizable detectors. LHST expands the image-level labels with language hierarchy and enables co-regularization between the expanded labels and self-training. Specifically, the expanded labels regularize self-training by providing richer supervision and mitigating the image-to-box label mismatch, while self-training allows assessing and selecting the expanded labels according to the predicted reliability. In addition, we design language hierarchical prompt generation that introduces language hierarchy into prompt generation which helps bridge the vocabulary gaps between training and testing. Extensive experiments show that the proposed techniques achieve superior generalization performance consistently across 14 widely studied object detection datasets.

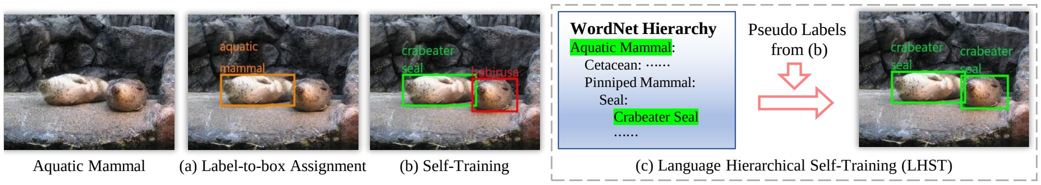
1 Introduction

Object detection aims to locate and identify objects in images by providing basic visual information of "where and what objects are". Thanks to the recent advances of deep neural networks, it has achieved great success with various applications in autonomous driving [1, 2, 3, 4], intelligent surveillance [5, 6, 7, 8], wildlife tracking [9, 10, 11], etc. However, learning a generalizable object detector for various downstream tasks that have different data distributions and data vocabularies remains an open research challenge. To this end, weakly-supervised object detection (WSOD) [12, 13, 14, 15], which allows access of large-scale image-level datasets (e.g., ImageNet-21K [16] with 14M images of 21K classes) with super rich data distributions and data vocabularies, has reignited new research interest under the context of learning generalizable detectors.

While exploiting WSOD to learn generalizable detectors, one typical challenge is that the provided image-level labels do not convey precise object information [15] and often mismatch with box-level labels. Recent methods address this challenge by designing various label-to-box assignment strategies that assign the image-level labels to the predicted top-score [13, 14] or max-size [15] object proposals. However, the mismatch problem remains due to the restriction of the raw image-level labels [17]. At the other end, self-training [18, 19, 20] with the detectors pre-trained with [13, 14, 15] can generate box-level pseudo labels without the restriction of image-level labels. It allows learning from more object proposals without the image-to-box label mismatch issue, but it does not benefit much from the provided image-level label supervision.

[*] Corresponding author

38th Conference on Neural Information Processing Systems (NeurIPS 2024).



Aquatic Mammal (a) Label-to-box Assignment (b) Self-Training (c) Language Hierarchical Self-Training (LHST)

Figure 1: Image-level labels in large-scale datasets such as ImageNet-21k [16] often do not convey precise object information [17, 15] which affects while learning generalizable detectors. Recent methods tackle this issue by various label-to-box assignment strategies [12, 13, 14, 15] as in (a) but are heavily restricted by raw image-level labels and still suffer from image-to-box label mismatch [17]. Self-training [18] with the detectors pre-trained with [13, 14, 15] could circumvent the label mismatch issue but the generated pseudo box labels are error-prone due to the lack of proper supervision as in (b). Our proposed LHST introduces language hierarchy to expand the image-level labels and enables co-regularization between the expanded labels and self-training which allows producing more accurate pseudo box labels in (c).

We propose to incorporate image-level supervision with self-training for learning generalizable detectors, aiming to benefit from self-training while effectively making use of image-level weak supervision. We start from a simple observation: the image-to-box label mismatch largely comes from the ambiguity in language hierarchy, e.g., the image-level label *Aquatic Mammal* in Figure 1 can cover different object-level labels such as seals, dolphins, walruses, etc. With the above observations, we design a **Det**ector with **L**anguage **H**ierarchy (DetLH) that combines language hierarchical self-training (LHST) and language hierarchical prompt generation (LHPG) for learning generalizable detectors.

LHST introduces WordNet's language hierarchy [21] to expand the image-level labels and accordingly enables co-regularization between the expanded labels and self-training. Specifically, the expanded labels are not all reliable though they can mitigate the image-to-box label mismatch problem by providing richer supervision. Here self-training can predict reliability scores for the expanded labels for better selection or weightage of the expanded labels. At the other end, self-training with pseudo box labels allows learning from more proposals and can circumvent the image-to-box label mismatch , but the box-level pseudo labels are usually noisy and may lead to learning degradation [15]. Here the expanded labels provide richer and more flexible supervision which can effectively help suppress prediction noises in self-training.

LHPG helps bridge the vocabulary gaps between training and testing by introducing WordNet's language hierarchy into prompt generation process. Specifically, LHPG leverages the CLIP language encoder [22] to measure the embedding distances between test concepts and WordNet synsets, and then generates the prompt for a given test concept from its best matched WordNet synset. In this way, the test prompts generated by LHPG have been standardized by WordNet and are well aligned with our proposed detector that is trained with WordNet information via LHST. In another word, the combination of LHST and LHPG actually leverages WordNet as a standard and intermediate vocabulary that bridges the gaps between training and testing vocabularies, generating better prompts and leading to better detection performance on downstream applications.

The main contributions of this work are threefold. *First*, we propose language hierarchical self-training that incorporates language hierarchy with self-training for weakly-supervised object detection. *Second*, we design language hierarchical prompt generation, which introduces language hierarchy into prompt generation to bridge the vocabulary gaps between detector training and testing. *Third*, extensive experiments show that our DetLH achieves superior generalization performance consistently across 14 detection benchmarks.

2 Related Work

Weakly-supervised object detection (WSOD) aims to train object detectors using image-level supervision. Traditional WSOD methods [23, 24, 25, 26, 27] use image-level annotations only without any box annotations and thus focus on low-level proposal mining techniques [28, 29, 12, 30, 31, 32], leading to unsatisfying localization performance. **Semi-supervised WSOD** [33, 34, 35, 36, 37, 38, 39] has been proposed to further improve the performance, which leverages both box-level and image-level annotated data. With better localization quality, recent methods [13, 14, 15, 40] design various label-to-box assignment strategies, such as assigning image-level labels to max-score anchors [13], max-score proposals [14] or max-size proposals [15]. Our work belongs to semi-supervised WSOD. Different from previous methods, we tackle the image-to-box label mismatch by introducing language hierarchy into self-training.

Large-vocabulary object detection [41, 13, 42, 43, 44] researches on detecting thousands of categories. Most previous papers focus on tackling the long-tail issue [45, 46, 47, 48, 49, 50], e.g., by using equalization losses [51, 52], SeeSaw loss [53], or Federated Loss [54]. Recent semi-supervised WSOD methods [13, 14, 15] and our work circumvent the long-tail problem by leveraging more balanced image-level datasets such as ImageNet-21K.

Open-vocabulary object detection focuses on detecting objects conditioned on arbitrary words (i.e., any category names). A common strategy [55, 56, 57, 58, 59] is to replace the detector's classification layer with the language embeddings of category names. Recent methods [60, 61, 62, 63, 17, 15] leverage the powerful CLIP [22] model by using its text embeddings [60, 61, 62, 63, 17, 15] or conducting knowledge distillation [60, 63, 17]. Similar to Detic [15], our work uses CLIP text embeddings as the classifier and leverages image-level annotated data instead of distilling knowledge from CLIP.

Language hierarchy has been widely studied for visual recognition tasks [64], especially for large-vocabulary visual recognition. Most existing studies [65, 66, 67] focus on image classification tasks, e.g., leveraging language hierarchy for multi-label image classification [65, 66, 67, 68, 69, 70, 71], modelling hierarchical relations among classes [68, 69] or facilitating classification training [70, 71]. Different from previous work, we introduce language hierarchy into self-training for weakly-supervised object detection.

3 Method

This work focuses on learning generalizable object detectors via weakly-supervised detector training [15], which leverages additional large-scale image-level datasets to enlarge the data distributions and data vocabularies in detector training. We first describe the task definition with training and evaluation setups. Then, we present our proposed DetLH which is detailed in two major aspects on Language Hierarchical Self-training (LHST) that introduces language hierarchy into detector training, and Language Hierarchical Prompt Generation (LHPG) that introduces language hierarchy into prompt generation.

3.1 Task Definition

Training setup. The training data consists of two parts: 1) a detection dataset $\mathcal{D}_{det} = \{(x, y_{det})_i\}_{i=1}^{|D_{det}|}$, where x denotes an image while y_{det} stands for the class and bounding box labels for x; 2) an image classification dataset $\mathcal{D}_{cls} = \{(x, y_{cls})_i\}_{i=1}^{|D_{cls}|}$ where y_{cls} denotes the image-level label (i.e., a one-hot vector) for x. Given the two datasets, the goal is to learn a generalizable detection model F by jointly optimizing F over $\mathcal{D}_{det}$ and $\mathcal{D}_{cls}$:

$$Loss = \sum_{(x,y_{det})\in\mathcal{D}_{det}} \mathcal{L}_{det}(F(x), y_{det}) + \sum_{(x,y_{cls})\in\mathcal{D}_{cls}} \mathcal{L}_{weak}(F(x), y_{cls}), \tag{1}$$

where $\mathcal{L}_{det}(\cdot) = \mathcal{L}_{rpn}(\cdot) + \mathcal{L}_{reg}(\cdot) + \mathcal{L}_{cls}(\cdot)$ is the fully-supervised detection loss function while $\mathcal{L}_{rpn}(\cdot)$, $\mathcal{L}_{reg}(\cdot)$, and $\mathcal{L}_{cls}(\cdot)$ denote RPN, Regression, and Classification loss functions, respectively. $\mathcal{L}_{weak}$ is the weakly-supervised loss function to train detectors with image-level labels.

Evaluation setup. As the goal is to learn a generalizable detection model that works well on various unseen downstream tasks, we conduct zero-shot cross-dataset evaluation[2] to assess the generalization performance of the trained detection model. Note, different domain adaptation [72, 73, 74, 75] that generally uses downstream data in training, our setup is similar to domain generalization [76, 77] that does not involve downstream data in training.

[2] zero-shot cross-dataset evaluation here means that the model is evaluated on unseen datasets, which is the same as the one defined in CLIP [22].

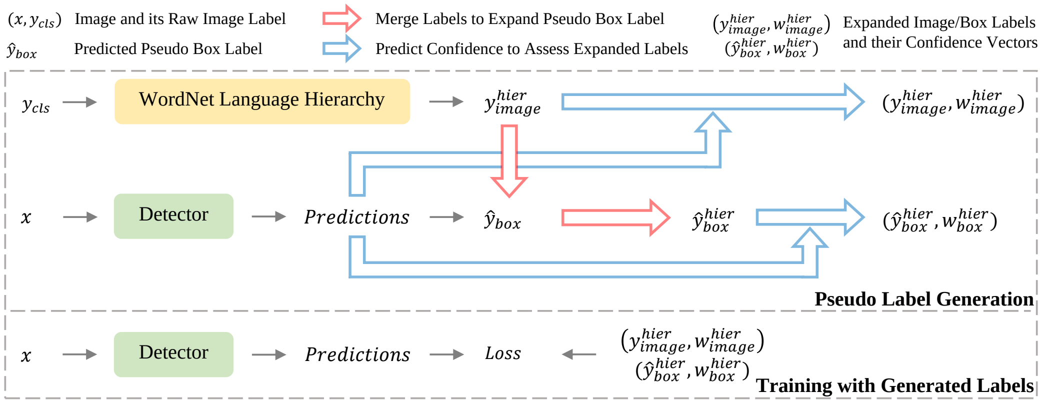

(x, y_{cls}) Image and its Raw Image Label
$\hat{y}_{box}$ Predicted Pseudo Box Label
Merge Labels to Expand Pseudo Box Label
Predict Confidence to Assess Expanded Labels
$(y^{hier}_{image}, w^{hier}_{image})$ $(\hat{y}^{hier}_{box}, w^{hier}_{box})$ Expanded Image/Box Labels and their Confidence Vectors
y_{cls}
WordNet Language Hierarchy
y^{hier}_{image}
$(y^{hier}_{image}, w^{hier}_{image})$
x
Detector
Predictions
$\hat{y}_{box}$
$\hat{y}^{hier}_{box}$
$(\hat{y}^{hier}_{box}, w^{hier}_{box})$
Pseudo Label Generation
x
Detector
Predictions
Loss
$(y^{hier}_{image}, w^{hier}_{image})$
$(\hat{y}^{hier}_{box}, w^{hier}_{box})$
Training with Generated Labels

Figure 2: The proposed language hierarchical self-training consists of two flows including Pseudo Label Generation (top box) and Training with Generated Labels (bottom box). The Pseudo Label Generation flow leverages WordNet to expand the image-level labels, and then merges the expanded image-level labels with the predicted pseudo box labels, such that the expanded image-level labels could provide richer and more flexible supervision (than the limited and rigid raw labels) to regularize the self-training which is prone to errors in pseudo labeling. In addition, as the labels expanded by WordNet (i.e., the expanded logits '1' in y^{hier}_{image} and y^{hier}_{box}) are not all reliable, Pseudo Label Generation predicts reliability scores for the expanded labels to adaptively re-weight them when applying them on different images or pseudo boxes. In Training with Generated Labels, we optimize the detector with the generated image-level and box-level labels, where the image-level training could regularize the training with pseudo box-level labels as pseudo box labels vary along training iterations and are not very stable.

Open-vocabulary Detector. We modify the classification layer of the detector into an open-vocabulary format such that the detector could be tested over unseen datasets. Specifically, we replace the weights of the detector's classification layer with the fixed language embeddings encoded from class names, where the object classification could be achieved by matching the object's embedding and the fixed language embeddings. We adopt the CLIP language embeddings [22] as the classification weights as in [15, 60]. In this way, the modified detector could theoretically detect any target concepts on any target data. As reported in [15], detectors trained solely on detection datasets often exhibits constrained performance due to the small-scale training images and vocabularies. Similar to [15], our proposed DetLH introduces large-scale image-level datasets to enlarge the data distributions and data vocabularies in detector training, leading to more generalizable detectors and better generalization performance on various unseen datasets.

3.2 Language Hierarchical Self-training

The proposed LHST utilizes WordNet's language hierarchy to expand the image-level labels, which enables co-regularization between the expanded image-level labels and self-training as illustrated in Figure 2.

Overview. For *fully supervised detector training* over the detection dataset, we feed box-level annotated samples $(x, y_{det}) \in \mathcal{D}_{det}$ to the detection model F and optimize F with the standard fully supervised detection loss, i.e., the first term of Eq. 1. For *weakly-supervised detector training* over the image-level annotated dataset $(x, y_{cls}) \in \mathcal{D}_{cls}$ shown in Figure 2, we first leverage WordNet's language hierarchy to expand the raw image-level label y_{cls} into y^{hier}_{image} (the hierarchical image-level label), and merge y^{hier}_{image} and the generated pseudo box label $\hat{y}_{box}$ to acquire $\hat{y}^{hier}_{box}$ (the hierarchical box-level pseudo label). Then, we optimize the detector with $(\hat{y}^{hier}_{box}, w^{hier}_{box})$ and $(y^{hier}_{image}, w^{hier}_{image})$, where w^{hier}_{image} and w^{hier}_{box} denote the predicted reliability scores of the expanded logits '1' in y^{hier}_{image} and y^{hier}_{box} and are used to weight the labels in loss calculation.

Expanding image labels with language hierarchy. Given image-level annotated dataset $(x, y_{cls}) \in \mathcal{D}_{cls}$ (y_{cls} is a label vector with length C and C denotes the number of classes), we leverage WordNet's class name hierarchy [21] to expand y_{cls} into y^{hier}_{image} as the following:

$$y_{image}^{hier} = \text{WordNet}(y_{cls}), \tag{2}$$

where the function WordNet($\cdot$) recursively finds all hypernyms and hyponyms of the input (i.e., the class indicated in y_{cls}) and sets their positions in the label vector y_{cls} to be '1' to expand y_{cls} into y_{image}^{hier}. In this way, a single-label annotation could be expanded into a multi-label annotation within the very rich ImageNet-21K vocabulary.

Generating pseudo box labels with predictions. Given the image $x \in \mathcal{D}_{cls}$, we feed x into the detector F to acquire the prediction as following:

$$\{p_n^c\}_{1 \leq n \leq N, 1 \leq c \leq C} = F(x), \tag{3}$$

where p_n is the probability vector of the predicted n-th bounding box after Softmax, and p_n^c denotes the predicted c-th category probability. Note we filter out a prediction if its max confidence score is lower than the threshold t, and N denotes the number of predicted object proposals after filtering, i.e., $max(\{p_n^c\}_{1 \leq c \leq C}) \geq t, \forall n$.

Then the pseudo category label $\hat{y}_{box} = \{\hat{y}_n\}_{1 \leq n \leq N}$ for N boxes in image x is derived by:

$$\arg\max_{\hat{y}_n} \sum_{c=1}^{C} \hat{y}_n^c \log p_n^c, \ s.t. \ \hat{y}_n \in \Delta^C, \forall n, \tag{4}$$

where $\hat{y}_n = (\hat{y}_n^{(1)}, \hat{y}_n^{(2)}, ..., \hat{y}_n^{(C)})$ is the predicted category label, and Δ^C denotes a probability simplex with length C.

Merging image and pseudo box labels. As the predicted pseudo box label $\hat{y}_{box}$ is error-prone, we regularize it with the expanded image-level supervision by merging y_{image}^{hier} and $\hat{y}_{box}$ as the following:

$$\hat{y}_{box}^{hier}(n) = \hat{y}_{box}(n) \vee y_{image}^{hier}, \forall n, \tag{5}$$

where $\vee$ denotes the logical "OR" operator.

Assessing the expanded labels. As the labels expanded by WordNet (i.e., the expanded logits '1' in $\hat{y}_{box}^{hier} = \{\hat{y}_n^c\}_{1 \leq n \leq N, 1 \leq c \leq C}$) are not all reliable, we predict a reliability score w_n^c for the expanded label to adaptively re-weight $y_n^c \in \hat{y}_{box}^{hier}$ when applying it on different pseudo boxes. We measure the reliability of y_n^c with prediction p_n^c, and $w_{box}^{hier} = \{w_n^c\}_{1 \leq n \leq N, 1 \leq c \leq C}$ can be derived by:

$$w_n^c = \begin{cases} p_n^c & \text{if } {y_{image}^{hier}}^{(c)} \neq y_{cls}^{(c)} \\ 1 & \text{otherwise,} \end{cases} \tag{6}$$

where ${y_{image}^{hier}}^{(c)} \neq y_{cls}^{(c)}$ returns True if the c-th label logit in y_{image}^{hier} is expanded by WordNet, which also applies to $\hat{y}_{box}^{hier}$ as $\hat{y}_{box}^{hier}$ is expanded by mergeing it with y_{image}^{hier}.

Given the prediction $\{p_n^c\}_{1 \leq n \leq N, 1 \leq c \leq C}$, the merged pseudo box label $\hat{y}_{box}^{hier} = \{\hat{y}_n^c\}_{1 \leq n \leq N, 1 \leq c \leq C}$ and its reliability score $w_{box}^{hier} = \{w_n^c\}_{1 \leq n \leq N, 1 \leq c \leq C}$, we optimize the detector F as the following:

$$\mathcal{L}_{box}(F(x)) = \sum_{n}^{N} \sum_{c}^{C} (\text{BCE}(p_n^c, y_n^c) \times w_n^c), \tag{7}$$

where BCE($\cdot$) denotes the binary cross-entropy loss.

In addition, training with the predicted pseudo box labels is not very stable as pseudo box labels vary along training process. Thus, we regularize the training of $\mathcal{L}_{box}(F(x))$ with an image-level loss defined as the following:

$$\mathcal{L}_{image}(F(x)) = \sum_{c}^{C} (\text{BCE}(p_{image}^c, {y_{image}^{hier}}^{(c)}) \times w_{image}^c), \tag{8}$$

where $p_{image} = \{p_{image}^c\}_{1 \leq c \leq C}$ denotes the category probability predicted for the image-level proposal. $w_{image}^{hier} = \{w_{image}^c\}_{1 \leq c \leq C}$ denotes the reliability score for the expanded logits "1" in

y^{hier}_{image}. Similar to Eq. 6, $w^c_{image} = p^c_{image}$ if ${y^{hier}_{image}}^{(c)} \neq y^{(c)}_{cls}$, otherwise $w^c_{image} = 1$. Besides, $\text{BCE}(\cdot)$ denotes the binary cross-entropy loss.

Training objective. The overall training objective of Language Hierarchical Self-training is defined as:

$$\mathcal{L}_{lhst} = \sum_{(x,y_{det})\in\mathcal{D}_{det}} \mathcal{L}_{det}(F(x), y_{det}) + \sum_{(x,y_{cls})\in\mathcal{D}_{cls}} (\mathcal{L}_{box}(F(x)) + \mathcal{L}_{image}(F(x))) \tag{9}$$

Language Hierarchical Prompt Generation. As the goal is to learn a generalizable detection model that works well on various downstream tasks, one typical challenge is the vocabulary gap between detector training datasets (i.e., LVIS and ImageNet-21K) and detector testing datasets (e.g., object365 or customized data). A common solution of tackling the vocabulary gaps is to conduct prompt learning [78] to generate proper category prompts. However, prompt learning generally requires labeled target images for additional training.

In this work, we tackle the vocabulary gaps by generating prompts with the help of WordNet, which introduces little computation overhead and does not require labeled target images and additional training. To this end, we design language hierarchical prompt generation (LHPG) that works by incorporating WordNet information into prompt generation process. Specifically, LHPG leverages CLIP language encoder [22] to measure the embedding distances between test concepts and WordNet synsets, and then generates the prompt for a given test concept from its best matched WordNet synset: $V^{\text{WordNet}}_{test} = \text{CLIP}(V_{test}, \text{WordNet})$, where V_{test} denotes test vocabulary, WordNet denotes WordNet synsets, CLIP denotes CLIP language encoder and $V^{\text{WordNet}}_{test}$ stands for the best matched WordNet synsets for the classes in V_{test}. Then, we generate test prompts from $V^{\text{WordNet}}_{test}$. As compared with V_{test}, our $V^{\text{WordNet}}_{test}$ has been standardized by WordNet and is well aligned with our proposed detector that is trained with WordNet information via LHST. In another word, the combination of LHST and LHPG makes use of WordNet as a standard and intermediate vocabulary that bridges the gaps between training and testing vocabularies, generating better prompts and leading to better detection performance on downstream applications.

4 Experiments

We evaluate our DetLH on 14 widely adopted detection benchmarks. We follow the zero-shot cross-dataset object detection setting proposed in [17, 15]. More details like **Dataset** and **Implementation Details** are provided in the appendix.

Table 1: **Zero-shot cross-dataset object detection for common objects.** All detectors are trained over the training datasets (LVIS and ImageNet-21K) and evaluated over target datasets (i.e., Object365 and Pascal VOC with objects from common classes and scenarios) without finetuning. "Dataset-specific oracles" denote the detectors that are fully supervised which are trained by using the training data of respective datasets.

Method	Object365 [79]						Pascal VOC [80]					
	AP	AP50	AP75	APs	APm	APl	AP	AP50	AP75	APs	APm	APl
WSDDN [12]	21.0	29.1	22.7	8.7	20.9	31.2	61.6	82.7	67.5	24.8	50.9	73.5
YOLO9000 [13]	21.0	28.5	22.6	8.6	20.7	30.9	62.6	83.6	68.7	23.7	52.0	73.9
DLWL [14]	21.3	29.1	23.0	8.8	21.0	31.5	62.4	83.4	68.3	23.8	51.2	73.8
Detic [15]	21.6	29.4	23.4	9.0	21.4	31.9	62.4	83.3	68.5	23.7	51.8	73.9
DetLH (Ours)	**23.6**	**32.5**	**25.5**	**9.8**	**23.5**	**35.0**	**64.4**	**86.1**	**70.8**	**25.3**	**54.1**	**75.3**
Dataset-specific oracles	31.2	-	-	-	-	-	54.4	79.7	59.1	19.0	40.8	64.5

4.1 Comparison with the state-of-the-art

We conduct extensive experiments to benchmark our proposed DetLH with state-of-the-art methods. We evaluate them on 14 widely studied object detection datasets to assess their zero-shot cross-dataset generalization ability. Tables 1- 5 report zero-shot cross-dataset detection results for common objects, autonomous driving, intelligent surveillance, and wildlife detection, respectively. More details are to be described in the following paragraphs.

Table 2: **Zero-shot cross-dataset object detection for autonomous driving.** All detectors are trained over the training datasets (LVIS and ImageNet-21K) and evaluated over autonomous driving datasets (i.e., Cityscapes, Vistas and SODA10M) without finetuning.

Method	Cityscapes [1]			Vistas [2]			SODA10M [3]			Average		
	AP	AP50	AP75	AP	AP50	AP75	AP	AP50	AP75	AP	AP50	AP75
WSDDN [12]	28.2	45.4	27.1	22.3	34.0	23.3	17.4	28.9	17.1	22.6	36.1	22.4
YOLO9000 [13]	28.8	46.2	27.4	22.5	34.6	23.4	18.3	30.4	18.0	23.2	37.0	22.9
DLWL [14]	28.6	45.6	28.1	22.5	34.7	23.2	18.3	30.4	18.0	23.1	36.9	23.0
Detic [15]	29.6	47.1	28.4	23.0	35.6	23.6	18.8	30.9	18.5	23.8	37.9	23.5
DetLH (Ours)	**31.2**	**50.3**	**29.1**	**26.5**	**44.0**	**25.8**	**25.1**	**38.4**	**26.1**	**27.6**	**44.2**	**27.0**
Dataset-specific oracles	43.0	69.0	42.6	28.1	45.8	28.5	44.7	68.2	47.3	38.6	61.0	39.5

Table 3: **Zero-shot cross-dataset object detection under different weather and time-of-day conditions (using metric AP50).** All detectors are trained over the training datasets (LVIS and ImageNet-21K) and evaluated over BDD100K and DAWN datasets without finetuning.

Method	BDD100K-weather [81]						BDD100K-time-of-day [81]				DAWN [82]			Avg
	rainy	snowy	overcast	cloudy	foggy	undefined	daytime	dawn&dusk	night	undefined	fog	sand	snow	
WSDDN [12]	35.0	33.0	38.3	41.7	26.7	46.0	39.1	35.5	27.9	50.2	62.6	55.0	65.6	42.8
YOLO9000 [13]	34.4	33.6	39.5	41.8	31.0	45.4	39.6	35.9	28.8	46.6	60.6	53.9	64.4	42.7
DLWL [14]	34.8	33.4	38.8	43.8	**40.2**	45.2	40.1	35.1	28.7	45.0	62.1	56.1	63.7	43.6
Detic [15]	34.3	33.2	39.5	41.9	27.9	45.4	39.2	35.5	28.8	48.2	52.3	54.1	56.1	41.3
DetLH (Ours)	**40.2**	**37.5**	**48.2**	**49.3**	37.1	**49.9**	**45.7**	**40.0**	**34.2**	**53.0**	**63.2**	**57.6**	**67.3**	**47.9**
Dataset-specific oracles	52.0	52.5	56.3	56.3	21.3	65.4	57.0	50.4	48.6	27.7	56.7	48.4	26.4	47.6

Table 4: **Zero-shot cross-dataset object detection for intelligent surveillance.** All detectors are trained over the training datasets (LVIS and ImageNet-21K) and evaluated over surveillance datasets MIO-TCD, BAAI-VANJEE, DETRAC and UAVDT without finetuning.

Method	MIO-TCD [5]			BAAI-VANJEE [6]			DETRAC [7]			UAVDT [8]			Average		
	AP	AP50	AP75	AP	AP50	AP75	AP	AP50	AP75	AP	AP50	AP75	AP	AP50	AP75
WSDDN [12]	11.3	17.6	11.6	13.1	19.6	13.3	25.6	35.3	31.1	17.1	31.9	16.0	16.7	26.1	18.0
YOLO9000 [13]	12.7	19.7	13.0	13.1	19.3	13.0	29.1	39.4	35.3	18.6	33.9	17.7	18.3	28.1	19.7
DLWL [14]	12.9	20.1	12.9	13.5	20.0	13.6	27.8	38.0	33.6	16.6	31.1	15.1	16.7	26.1	18.0
Detic [15]	13.4	20.6	13.9	16.9	23.6	17.6	28.7	39.2	34.8	18.6	34.2	17.6	19.4	29.4	21.0
DetLH (Ours)	**15.8**	**24.5**	**16.0**	**17.9**	**25.1**	**18.5**	**32.7**	**44.0**	**39.7**	**20.1**	**36.6**	**19.3**	**21.6**	**32.6**	**23.4**
Dataset-specific oracles	45.2	63.1	50.8	40.6	58.6	43.7	53.1	70.6	63.5	33.8	60.4	35.2	43.2	63.2	48.3

Object detection for common objects. Table 1 shows that DetLH outperforms state-of-the-art methods clearly on common object datasets Object365 and Pascal VOC. In addition, we can observe that DetLH even brings significant gains above the *dataset-specific oracle* (i.e., the model that is fully trained on the target training data) on Pascal VOC (i.e., a small-scale dataset), showing the advantages of leveraging large-scale training data.

Object detection for autonomous driving. As shown in Table 2, our DetLH outperforms state-of-the-art methods by large margins on various autonomous driving datasets, showing that DetLH still works effectively while facing large variations in camera views from autonomous driving scenarios to the base-dataset scenarios (LVIS and ImageNet-21K), e.g., autonomous driving images are captured under very different camera views. In addition, the experimental results in Table 3 show that our DetLH brings significant performance gains against state-of-the-art methods when encountering various weather and time-of-day conditions, which demonstrates the effectiveness of DetLH while detecting objects under large noises [83], e.g., the images captured under different weather and time-of-day conditions may have very different styles and image quality.

Object detection for intelligent surveillance. From Table 4, we can observe that our DetLH outperforms state-of-the-art methods by clear margins on various intelligent surveillance datasets, indicating that DetLH is also tolerant to large changes in the camera lens and angles which often happen to intelligent-surveillance images that are captured under very different camera lens and angles (e.g., surveillance cameras are often with the wide-angle lens and used in high angle views).

Object detection for Wildlife. The experimental results in Table 5 show that our DetLH performs well on various wildlife detection datasets, showing that DetLH works effectively for detecting fine-grained categories that exist widely in wildlife detection datasets. The significant performance

Table 5: **Zero-shot cross-dataset object detection for Wildlife Detection.** All detectors are trained over the training datasets (LVIS and ImageNet-21K) and evaluated over wildlife datasets (i.e., Arthropod Detection, AfricanWildlife and Animals Detection) without finetuning.

Method	Arthropod Detection [9]			AfricanWildlife [10]			Animals Detection [11]			Average		
	AP	AP50	AP75	AP	AP50	AP75	AP	AP50	AP75	AP	AP50	AP75
WSDDN [12]	18.1	26.3	18.8	**76.7**	**88.2**	**84.0**	36.0	41.7	37.5	43.6	52.0	46.7
YOLO9000 [13]	22.6	33.2	22.5	75.9	87.5	83.2	39.0	45.4	40.8	45.8	55.3	48.8
DLWL [14]	25.3	34.7	26.3	74.7	86.2	81.3	41.7	48.1	43.7	47.2	56.3	50.4
Detic [15]	27.4	36.7	29.2	68.9	80.9	76.4	41.1	47.7	42.9	45.8	55.1	49.5
DetLH (Ours)	**36.2**	**49.0**	**38.3**	74.8	87.2	81.8	**44.3**	**51.2**	**46.3**	**51.8**	**62.5**	**55.5**
Dataset-specific oracles	75.1	86.3	79.9	82.7	90.9	89.1	64.4	74.6	69.4	74.1	83.9	79.5

gains largely come from the introduction of language hierarchy into detector training and prompt generation, which helps model the hierarchical relations among parent and fine-grained subcategories and thus leads to better fine-grained object detection.

The superior detection performance of our DetLH is largely attributed to our two core designs, i.e., LHST and LHPG. LHST enables effective usage of large-scale image-level annotated images and significantly enlarges the data distribution and the data vocabulary in detector training, yielding robust performance under large cross-dataset gaps in data distribution and vocabulary. LHPG ingeniously helps mitigate the vocabulary gaps between detector training and testing. It improves the overall confidence of detection and benefits the detection as large data distribution gaps (or large data vocabulary gaps) often lead to low-confidence predictions and poor detection results.

4.2 Ablation Studies

We perform ablation studies with Swin-B [84] based CenterNet2 [54] over the large-scale Object365 dataset as shown in Table 6. As the core of our proposed DetLH, we examine how our designed LHST and LHPG contribute to the overall performance of zero-shot cross-dataset object detection. As shown in Table 6, the *baseline* (Box-Supervised [15]) does not perform well as it uses box-level training data only. It can be observed that LHST outperforms the baseline clearly, showing that LHST can effectively leverage the large-scale image-level annotated dataset to significantly enlarge the data distribution and data vocabulary involved in detector training, leading to much better zero-shot cross-dataset detection performance. In addition, LHPG brings clear performance improvements in zero-shot cross-dataset detection by introducing language hierarchy into prompt generation, demonstrating the effectiveness of LHPG in mitigating the vocabulary gaps between training and testing. Moreover, the inclusion of both LHST and LHPG in the proposed DetLH performs clearly the best, indicating the complementary property of our two designs.

Table 6: **Ablation studies of our DetLH** with Language Hierarchical Self-training (LHST) and Language Hierarchical Prompt Generation (LHPG). The experiments are conducted with Swin-B based CenterNet2 [15] and the detectors are evaluated on Object365 in zero-shot cross-dataset object detection setup.

Method	LHST	LHPG	AP50
Box-Supervised [15]			26.5
	✓		31.3
		✓	31.0
DetLH (Ours)	✓	✓	**32.5**

4.3 Discussion

Table 7: **Zero-shot cross-dataset object detection on various datasets.** Results are averaged on 14 widely studied datasets.

Method	Averaged over 14 detection datasets					
	AP	AP50	AP75	APs	APm	APl
WSDDN [12]	29.9	42.5	31.4	14.8	25.9	44.2
YOLO9000 [13]	30.9	43.8	32.4	14.1	25.8	45.1
DLWL [14]	31.0	44.0	32.5	15.4	26.3	45.3
Detic [15]	31.0	44.0	32.8	14.6	27.5	45.5
DetLH (Ours)	**34.6**	**49.3**	**36.4**	**16.0**	**28.4**	**49.5**

Generalization across various detection tasks: We study the generalization of our DetLH by conducting zero-shot cross-dataset object detection on 14 widely studied object detection datasets. Tables 1- 5 show that DetLH achieves superior performance consistently across all the detection applications. Besides, Table 7 summarizes the detection results averaged on 14 datasets, showing that DetLH clearly outperforms the state-of-the-art methods.

Generalization across various network architectures: We study the generalization of the proposed DetLH from the perspective of network architectures. Specifically, we perform extensive evaluations with four representative network architectures, including one Transformer-based (i.e., Swin-B) and three CNN-based (i.e., ConvNeXt-T, ResNet-50 and ResNet-18). Experimental results in Table 8 show that the proposed DetLH outperforms the state-of-the-art method consistently over different network architectures.

Table 8: **Zero-shot cross-dataset object detection with different network architectures.** All networks architectures are trained over the training datasets (LVIS and ImageNet-21K) and evaluated over Object365 without finetuning.

Method	Architecture	Object365					
		AP	AP50	AP75	APs	APm	APl
Detic [15]	Swin-B [84]	21.6	29.4	23.4	9.0	21.4	31.9
DetLH (Ours)		23.6	32.5	25.5	9.8	23.5	35.0
Detic [15]	ConvNeXt-T [85]	16.9	23.5	18.1	6.8	16.6	24.9
DetLH (Ours)		18.9	26.8	20.2	7.6	18.8	28.2
Detic [15]	ResNet-50 [86]	16.2	22.8	17.5	6.3	16.2	24.1
DetLH (Ours)		17.7	25.5	19.0	6.9	17.9	26.4
Detic [15]	ResNet-18 [86]	10.8	15.5	11.6	3.9	10.2	16.2
DetLH (Ours)		11.8	17.3	12.5	4.3	11.4	17.7

Parameter Studies for Language Hierarchical Self- training (LHST). In generating pseudo box labels in LHST, we filter out a prediction if its max confidence score is lower than the threshold t. We study the threshold t by changing it from 0.65 to 0.85 with a step of 0.05. Table 12 reports the experimental results on zero-shot transfer object detection over object365 dataset. We can observe that the detection performance is not sensitive to the threshold t.

Table 9: **Parameter Studies for Language Hierarchical Self- training (LHST)** on zero-shot transfer object detection over object365 dataset. We study the thresholding parameter t used in generating pseudo box labels in LHST.

Threshold t	0.65	0.70	0.75	0.80	0.85
AP50	31.1	31.3	31.3	31.3	31.2

Due to the space limit, we provide more DetLH discussions and visualizations in the appendix.

5 Conclusion

This paper presents DetLH, a Detector with Language Hierarchy that combines language hierarchical self-training (LHST) and language hierarchical prompt generation (LHPG) for learning generalizable object detectors. LHST introduces WordNet's language hierarchy to expand the image-level labels and accordingly enables co-regularization between the expanded labels and self-training. LHPG helps mitigate the vocabulary gaps between training and testing by introducing WordNet's language hierarchy into prompt generation. Extensive experiments over multiple object detection tasks show that our DetLH achieves superior performance as compared with state-of-the-art methods. In addition, we demonstrate that DetLH works well with different network architectures such as Swin-B, ConvNeXt-T, ResNet-50, etc. Moving forward, we will explore language hierarchy to further expand the labels in an open-vocabulary manner in addition to the closed ImageNet-21K's vocabulary.

References

[1] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[2] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE international conference on computer vision*, pages 4990–4999, 2017.

[3] Jianhua Han, Xiwen Liang, Hang Xu, Kai Chen, Lanqing Hong, Jiageng Mao, Chaoqiang Ye, Wei Zhang, Zhenguo Li, Xiaodan Liang, et al. Soda10m: a large-scale 2d self/semi-supervised object detection dataset for autonomous driving. *arXiv preprint arXiv:2106.11118*, 2021.

[4] Jingyi Zhang, Jiaxing Huang, Zhipeng Luo, Gongjie Zhang, Xiaoqin Zhang, and Shijian Lu. Da-detr: Domain adaptive detection transformer with information fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23787–23798, 2023.

[5] Zhiming Luo, Frederic Branchaud-Charron, Carl Lemaire, Janusz Konrad, Shaozi Li, Akshaya Mishra, Andrew Achkar, Justin Eichel, and Pierre-Marc Jodoin. Mio-tcd: A new benchmark dataset for vehicle classification and localization. *IEEE Transactions on Image Processing*, 27(10):5129–5141, 2018.

[6] Deng Yongqiang, Wang Dengjiang, Cao Gang, Ma Bing, Guan Xijia, Wang Yajun, Liu Jianchao, Fang Yanming, and Li Juanjuan. Baai-vanjee roadside dataset: Towards the connected automated vehicle highway technologies in challenging environments of china. *arXiv preprint arXiv:2105.14370*, 2021.

[7] Longyin Wen, Dawei Du, Zhaowei Cai, Zhen Lei, Ming-Ching Chang, Honggang Qi, Jongwoo Lim, Ming-Hsuan Yang, and Siwei Lyu. Ua-detrac: A new benchmark and protocol for multi-object detection and tracking. *Computer Vision and Image Understanding*, 193:102907, 2020.

[8] Dawei Du, Yuankai Qi, Hongyang Yu, Yifan Yang, Kaiwen Duan, Guorong Li, Weigang Zhang, Qingming Huang, and Qi Tian. The unmanned aerial vehicle benchmark: Object detection and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 370–386, 2018.

[9] Geir Drange. Arthropod taxonomy orders object detection dataset, 2020.

[10] Wichayas YoLov5. African animals dataset. `https://universe.roboflow.com/wichayas-yolov5/african_animals`, may 2022. visited on 2023-03-01.

[11] Kaggle. Animal detection dataset. `https://universe.roboflow.com/kaggle/animal-detection-7vafe`, apr 2022. visited on 2023-03-01.

[12] Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2846–2854, 2016.

[13] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[14] Vignesh Ramanathan, Rui Wang, and Dhruv Mahajan. Dlwl: Improving detection for lowshot classes with weakly labelled data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9342–9352, 2020.

[15] Zhou Xingyi. Detecting twenty-thousand classes using image-level supervision. *arXiv preprint arXiv:2201.02605*, 2022.

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[17] Hanoona Rasheed, Muhammad Maaz, Muhammad Uzair Khattak, Salman Khan, and Fahad Shahbaz Khan. Bridging the gap between object and image-level representations for open-vocabulary detection. *arXiv preprint arXiv:2207.03482*, 2022.

[18] Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection. *arXiv preprint arXiv:2005.04757*, 2020.

[19] Jingyi Zhang, Jiaxing Huang, Zichen Tian, and Shijian Lu. Spectral unsupervised domain adaptation for visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9829–9840, 2022.

[20] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. Cross-view regularization for domain adaptive panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10133–10144, 2021.

[21] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[22] 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 *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[23] Xiaoyan Li, Meina Kan, Shiguang Shan, and Xilin Chen. Weakly supervised object detection with segmentation collaboration. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9735–9744, 2019.

[24] Yunhang Shen, Rongrong Ji, Yan Wang, Zhiwei Chen, Feng Zheng, Feiyue Huang, and Yunsheng Wu. Enabling deep residual networks for weakly supervised object detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*, pages 118–136. Springer, 2020.

[25] Yunhang Shen, Rongrong Ji, Yan Wang, Yongjian Wu, and Liujuan Cao. Cyclic guidance for weakly supervised joint detection and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 697–707, 2019.

[26] Fang Wan, Chang Liu, Wei Ke, Xiangyang Ji, Jianbin Jiao, and Qixiang Ye. C-mil: Continuation multiple instance learning for weakly supervised object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2199–2208, 2019.

[27] Ke Yang, Dongsheng Li, and Yong Dou. Towards precise end-to-end weakly supervised object detection network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8372–8381, 2019.

[28] Pablo Arbeláez, Jordi Pont-Tuset, Jonathan T Barron, Ferran Marques, and Jitendra Malik. Multiscale combinatorial grouping. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 328–335, 2014.

[29] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104:154–171, 2013.

[30] Peng Tang, Xinggang Wang, Xiang Bai, and Wenyu Liu. Multiple instance detection network with online instance classifier refinement. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2843–2851, 2017.

[31] Peng Tang, Xinggang Wang, Song Bai, Wei Shen, Xiang Bai, Wenyu Liu, and Alan Yuille. Pcl: Proposal cluster learning for weakly supervised object detection. *IEEE transactions on pattern analysis and machine intelligence*, 42(1):176–191, 2018.

[32] Zeyi Huang, Yang Zou, BVK Kumar, and Dong Huang. Comprehensive attention self-distillation for weakly-supervised object detection. *Advances in neural information processing systems*, 33:16797–16807, 2020.

[33] Bowen Dong, Zitong Huang, Yuelin Guo, Qilong Wang, Zhenxing Niu, and Wangmeng Zuo. Boosting weakly supervised object detection via learning bounding box adjusters. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2876–2885, 2021.

[34] Shijie Fang, Yuhang Cao, Xinjiang Wang, Kai Chen, Dahua Lin, and Wayne Zhang. Wssod: A new pipeline for weakly-and semi-supervised object detection. *arXiv preprint arXiv:2105.11293*, 2021.

[35] Yan Li, Junge Zhang, Kaiqi Huang, and Jianguo Zhang. Mixed supervised object detection with robust objectness transfer. *IEEE transactions on pattern analysis and machine intelligence*, 41(3):639–653, 2018.

[36] Yan Liu, Zhijie Zhang, Li Niu, Junjie Chen, and Liqing Zhang. Mixed supervised object detection by transferring mask prior and semantic similarity. *Advances in Neural Information Processing Systems*, 34:3978–3990, 2021.

[37] Jasper Uijlings, Stefan Popov, and Vittorio Ferrari. Revisiting knowledge transfer for training object class detectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1101–1110, 2018.

[38] Ziang Yan, Jian Liang, Weishen Pan, Jin Li, and Changshui Zhang. Weakly-and semi-supervised object detection with expectation-maximization algorithm. *arXiv preprint arXiv:1702.08740*, 2017.

[39] Yuanyi Zhong, Jianfeng Wang, Jian Peng, and Lei Zhang. Boosting weakly supervised object detection with progressive knowledge transfer. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI*, pages 615–631. Springer, 2020.

[40] Cheng Zhang, Tai-Yu Pan, Yandong Li, Hexiang Hu, Dong Xuan, Soravit Changpinyo, Boqing Gong, and Wei-Lun Chao. Mosaicos: a simple and effective use of object-centric images for long-tailed object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 417–427, 2021.

[41] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5356–5364, 2019.

[42] Bharat Singh, Hengduo Li, Abhishek Sharma, and Larry S Davis. R-fcn-3000 at 30fps: Decoupling detection and classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1081–1090, 2018.

[43] Hao Yang, Hao Wu, and Hao Chen. Detecting 11k classes: Large scale object detection without fine-grained bounding boxes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9805–9813, 2019.

[44] Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[45] Nadine Chang, Zhiding Yu, Yu-Xiong Wang, Animashree Anandkumar, Sanja Fidler, and Jose M Alvarez. Image-level or object-level? a tale of two resampling strategies for long-tailed detection. In *International conference on machine learning*, pages 1463–1472. PMLR, 2021.

[46] Chengjian Feng, Yujie Zhong, and Weilin Huang. Exploring classification equilibrium in long-tailed object detection. In *Proceedings of the IEEE/CVF International conference on computer vision*, pages 3417–3426, 2021.

[47] Yu Li, Tao Wang, Bingyi Kang, Sheng Tang, Chunfeng Wang, Jintao Li, and Jiashi Feng. Overcoming classifier imbalance for long-tail object detection with balanced group softmax. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10991–11000, 2020.

[48] Tai-Yu Pan, Cheng Zhang, Yandong Li, Hexiang Hu, Dong Xuan, Soravit Changpinyo, Boqing Gong, and Wei-Lun Chao. On model calibration for long-tailed object detection and instance segmentation. *Advances in Neural Information Processing Systems*, 34:2529–2542, 2021.

[49] Jialian Wu, Liangchen Song, Tiancai Wang, Qian Zhang, and Junsong Yuan. Forest r-cnn: Large-vocabulary long-tailed object detection and instance segmentation. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1570–1578, 2020.

[50] Songyang Zhang, Zeming Li, Shipeng Yan, Xuming He, and Jian Sun. Distribution alignment: A unified framework for long-tail visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2361–2370, 2021.

[51] Jingru Tan, Xin Lu, Gang Zhang, Changqing Yin, and Quanquan Li. Equalization loss v2: A new gradient balance approach for long-tailed object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1685–1694, 2021.

[52] Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Equalization loss for long-tailed object recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11662–11671, 2020.

[53] Jiaqi Wang, Wenwei Zhang, Yuhang Zang, Yuhang Cao, Jiangmiao Pang, Tao Gong, Kai Chen, Ziwei Liu, Chen Change Loy, and Dahua Lin. Seesaw loss for long-tailed instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9695–9704, 2021.

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

[55] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. Zero-shot object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 384–400, 2018.

[56] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[57] Shafin Rahman, Salman Khan, and Nick Barnes. Improved visual-semantic alignment for zero-shot object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11932–11939, 2020.

[58] Zhihui Li, Lina Yao, Xiaoqin Zhang, Xianzhi Wang, Salil Kanhere, and Huaxiang Zhang. Zero-shot object detection with textual descriptions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8690–8697, 2019.

[59] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-vocabulary object detection using captions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14393–14402, 2021.

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

[61] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Open-vocabulary image segmentation. *arXiv preprint arXiv:2112.12143*, 2021.

[62] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. *arXiv preprint arXiv:2201.03546*, 2022.

[63] Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy. Open-vocabulary detr with conditional matching. *arXiv preprint arXiv:2203.11876*, 2022.

[64] Carlos N Silla and Alex A Freitas. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22:31–72, 2011.

[65] Ivica Dimitrovski, Dragi Kocev, Suzana Loskovska, and Sašo Džeroski. Hierarchical annotation of medical images. *Pattern Recognition*, 44(10-11):2436–2449, 2011.

[66] Hui Liu, Danqing Zhang, Bing Yin, and Xiaodan Zhu. Improving pretrained models for zero-shot multi-label text classification through reinforced label hierarchy reasoning. *arXiv preprint arXiv:2104.01666*, 2021.

[67] Ilias Chalkidis, Manos Fergadiotis, Sotiris Kotitsas, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. An empirical study on large-scale multi-label text classification including few and zero-shot labels. *arXiv preprint arXiv:2010.01653*, 2020.

[68] Shiming Chen, Guosen Xie, Yang Liu, Qinmu Peng, Baigui Sun, Hao Li, Xinge You, and Ling Shao. Hsva: Hierarchical semantic-visual adaptation for zero-shot learning. *Advances in Neural Information Processing Systems*, 34:16622–16634, 2021.

[69] Thomas Mensink, Efstratios Gavves, and Cees GM Snoek. Costa: Co-occurrence statistics for zero-shot classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2441–2448, 2014.

[70] Kai Yi, Xiaoqian Shen, Yunhao Gou, and Mohamed Elhoseiny. Exploring hierarchical graph representation for large-scale zero-shot image classification. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XX*, pages 116–132. Springer, 2022.

[71] Zhong Cao, Jiang Lu, Sen Cui, and Changshui Zhang. Zero-shot handwritten chinese character recognition with hierarchical decomposition embedding. *Pattern Recognition*, 107:107488, 2020.

[72] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

[73] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. Model adaptation: Historical contrastive learning for unsupervised domain adaptation without source data. *Advances in Neural Information Processing Systems*, 34, 2021.

[74] Pan Zhang, Bo Zhang, Ting Zhang, Dong Chen, Yong Wang, and Fang Wen. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12414–12424, 2021.

[75] Jiaxing Huang, Dayan Guan, Aoran Xiao, Shijian Lu, and Ling Shao. Category contrast for unsupervised domain adaptation in visual tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1203–1214, 2022.

[76] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. Fsdr: Frequency space domain randomization for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6891–6902, 2021.

[77] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4396–4415, 2022.

[78] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16816–16825, 2022.

[79] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439, 2019.

[80] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–308, 2009.

[81] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020.

[82] Mourad A Kenk and Mahmoud Hassaballah. Dawn: vehicle detection in adverse weather nature dataset. *arXiv preprint arXiv:2008.05402*, 2020.

[83] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. Rda: Robust domain adaptation via fourier adversarial attacking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8988–8999, 2021.

[84] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[85] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022.

[86] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[87] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[88] Siddhesh Khandelwal, Raghav Goyal, and Leonid Sigal. Unit: Unified knowledge transfer for any-shot object detection and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5951–5961, 2021.

[89] Yunqiu Xu, Chunluan Zhou, Xin Yu, and Yi Yang. Cyclic self-training with proposal weight modulation for cross-supervised object detection. *IEEE Transactions on Image Processing*, 32:1992–2002, 2023.

[90] Yunqiu Xu, Yifan Sun, Zongxin Yang, Jiaxu Miao, and Yi Yang. H2fa r-cnn: Holistic and hierarchical feature alignment for cross-domain weakly supervised object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14329–14339, 2022.

[91] Krishna Kumar Singh, Santosh Divvala, Ali Farhadi, and Yong Jae Lee. Dock: Detecting objects by transferring common-sense knowledge. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 492–508, 2018.

[92] 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*, 2022.

[93] Lewei Yao, Jianhua Han, Youpeng Wen, Xiaodan Liang, Dan Xu, Wei Zhang, Zhenguo Li, Chunjing Xu, and Hang Xu. Detclip: Dictionary-enriched visual-concept paralleled pre-training for open-world detection. *arXiv preprint arXiv:2209.09407*, 2022.

[94] Yutong Lin, Chen Li, Yue Cao, Zheng Zhang, Jianfeng Wang, Lijuan Wang, Zicheng Liu, and Han Hu. A simple approach and benchmark for 21,000-category object detection. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XI*, pages 1–18. Springer, 2022.

[95] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021.

[96] Haotian Zhang, Pengchuan Zhang, Xiaowei Hu, Yen-Chun Chen, Liunian Li, Xiyang Dai, Lijuan Wang, Lu Yuan, Jenq-Neng Hwang, and Jianfeng Gao. Glipv2: Unifying localization and vision-language understanding. *Advances in Neural Information Processing Systems*, 35:36067–36080, 2022.

[97] Matthias Minderer. Simple open-vocabulary object detection with vision transformers. *ECCV*, 2022.

[98] Tiancheng Zhao, Peng Liu, Xiaopeng Lu, and Kyusong Lee. Omdet: Language-aware object detection with large-scale vision-language multi-dataset pre-training. *arXiv preprint arXiv:2209.05946*, 2022.

[99] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Simple multi-dataset detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7571–7580, 2022.

[100] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021.

[101] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses. *arXiv preprint arXiv:2104.10972*, 2021.

[102] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, 128(7):1956–1981, 2020.

Appendix

A Dataset and Implementation Details

A.1 Implementation Details

As in [17, 15], we adopt the CenterNet2 [54] with Swin-B [84] backbone in all the experiments (except for Table 8 where different backbone architectures were used, e.g., ConvNeXt-T, ResNet-50 and ResNet-18). We employ SGD [87] as the optimizer and adopt the cosine learning rate scheduler with a warm-up of 1000 iterations [15]. We set the input sizes of box-level annotated images (i.e., LVIS) and image-level annotated images (i.e., ImageNet-21K) as 896×896 and 448×448, respectively. As mentioned in Section 3.1, we employ the CLIP text embeddings [22] as the classifier instead of using the original one in [54]. During training, we sample box-level and image-level mini-batches in a $1 : 16$ ratio. We set the confidence threshold t (in pseudo box label generation in Eq. 3) as 0.75 in all experiments except in parameter analysis. Note we pre-train the detector over the training datasets (i.e., training on LVIS with the conventional detection loss and on ImageNet-21K with the conventional image classification loss) such that it can generate pseudo box-level labels of 21K classes for self-training.

As described in the main text, we train our detector over two training datasets LVIS and ImageNet-21K, and evaluate the trained detector over 14 evaluation datasets as listed without fine-tuning.

A.2 Training Dataset

LVIS [41] is a large vocabulary dataset designed for long-tailed instance segmentation, which contains 100K images and 1203 categories. LVIS provides high-quality instance-wise annotations, including instance masks, class labels and bounding boxes.

ImageNet-21K [16] is a large and diverse dataset over 14M images across more than 21K categories. All categories in ImageNet-21K are defined by WordNet Synsets with clear and accurate definitions and certain language hierarchy.

A.3 Evaluation Dataset

Object365 [79] is a large-scale object detection dataset designed for object detection in the wild. This dataset contains 638K images across 365 categories, including 600K images for training and 38K images for validation.

Pascal VOC [80] is a real-world dataset with two sub-datasets, $i.e.$, PASCAL VOC 2007 and PASCAL VOC 2012. PASCAL VOC 2007 contains 2,501 training images and 2,510 validation images, and PASCAL VOC 2012 contains 5,717 training images and 5,823 validation images. This dataset provides bounding box annotations with 20 categories.

Cityscapes [1] is a dataset designed for the understanding of street scenes. The images in Cityscapes are captured under normal weather conditions from 50 cities, including 2,975 training images and 500 validation images with pixel-wise instance annotations of 8 categories.

Vista [2] is a street-level autonomous driving dataset. This dataset contains high-resolution images that cover diverse urban scenes from around the world, including 18K training images and 2K validation images with pixel-wise instance annotations.

SODA10M [3] is a large-scale object detection dataset for autonomous driving, which contains 10M unlabeled images and 20K images with bounding box annotations of 6 object categories. The images in this dataset are collected within 27833 driving hours covering a variety time periods and locations across 32 different cities.

BDD100k [81] is a large-scale driving video dataset that contains diverse driving scenarios, including different weather conditions ($i.e.$, clear, cloudy, overcast, rainy, snowy and foggy) and times of day ($i.e.$, dawn, daytime and night). This dataset contains 100K videos, including 70K training videos and 10K validation videos with bounding box annotations of 10 categories.

Arthropod Detection [9] is a detection dataset for arthropods taxonomy orders identification. The images are collected from a variety of agricultural settings ($e.g.$, fields, greenhouses, warehouses), including over 12K images with bounding box annotations of 7 categories.

AfricanWildlife [10] is a detection dataset which contains images of African wildlife with bounding box annotations. This dataset contains 4 different categories of African wildlife including buffalo, elephant, rhino, zebra and each category contains 376 images.

Animals Detection [11] is a public dataset of various animals. This dataset contains animal images with bounding boxes of 80 different animal categories, including 6.8K training images and 1.9K validation images.

DAWN [82]) is a vehicle detection dataset that focuses on diverse traffic environment. This dataset contains 1K images from real-traffic environment, including fog, snow, rain and sandstorms. The images are annotated with object bounding boxes of 6 categories.

MIO-TCD [5] is an intelligent surveillance dataset for motorized traffic analysis, which contains 137,743 images captured in various times of the day and different periods of the year, and from different viewing angels. This dataset provides bounding box annotations with 11 categories.

BAAI-VANJEE [6] is an intelligent surveillance dataset which contains 5K images captured by VANJEE smart base station placed about 4.5m high. The images in this dataset vary in weather and traffic conditions, which are annotated with bounding box annotations of 12 categories.

DETRAC [7] is an intelligent surveillance dataset that contains over 14K images captured by a Canon EOS 550D camera at 24 different locations, covering various traffic patterns and conditions including urban highway, traffic crossings and T-junctions. The images in this dataset are annotated with bounding box annotations of 4 categories, including car, bus, van, and others.

UAVDT [8]) is a unmanned aerial vehicle detection dataset, which contains about 80K frames from 10 hours videos. The images in this dataset are captured by a unmanned aerial vehicle across various weather conditions ($i.e.$, daylight, night and fog) and multiple camera views ($i.e.$, front-view, side-view and bird-view). This dataset provides bounding box annotations with three categories including car, truck and bus.

B Additional Discussion

Strategy studies for Language Hierarchical Self-training. Our proposed language hierarchical self-training (LHST) introduces WordNet's language hierarchy [21] to expand the image-level labels and accordingly enables co-regularization between the expanded labels and self-training. We examine the superiority of the proposed LHST by comparing it with “Self-training" [18] and “Direct WordNet Hierarchy Labeling" [21]. “Self-training" is the standard self-training algorithm as in [18] while “Direct WordNet Hierarchy Labeling" denotes directly using the expanded image-level labels (by WordNet) for weakly-supervised detection training. Table 10 reports the experimental results, which show that either “Self-training" [18] or “Direct WordNet Hierarchy Labeling" [21] does not perform well. The reasons are: 1) the box-level pseudo labels in “Self-training" are usually error-prone, making the self-training process unstable and barely improving the performance; 2) the expanded image-level labels in ‘Direct WordNet Hierarchy Labeling" are not all reliable, training with which leads to unsatisfying performance. Besides, the combination of ‘Self-training" and “Direct WordNet Hierarchy Labeling" still works not very well largely because the direct combination of them does not well address their own drawbacks and limitations. On the other hand, our proposed LHST performs better clearly, as shown in the last row of Table 10. The superior performance of LHST is largely attributed the co-regularization design, which employs self-training to assess and re-weight the expanded labels according to the predicted reliability while enabling the expanded (and re-weighted) labels to regularize self-training by providing richer and flexible supervision (the flexible supervision is achieved by the adaptive re-weighting operation).

Table 10: **Strategy studies for Language Hierarchical Self-training** on zero-shot cross-dataset object detection over object365 dataset.

Analysis for Language Hierarchical Self- training	AP50
Detic [15]	29.4
Self-training [21]	29.4
Direct WordNet Hierarchy Labeling [21]	29.7
Self-training + Direct WordNet Hierarchy Labeling	29.9
Language Hierarchical Self- training (Ours)	31.3

Ablation studies of Language Hierarchical Self-training. As mentioned in the main text, our proposed language hierarchical self-training (LHST) consists of box-leve LHST (i.e., $\mathcal{L}_{box}(F(x))$ in Eq.7 in the main text) and image-level LHST (i.e., $\mathcal{L}_{image}(F(x))$ in Eq.8 in the main text), where image-level LHST could regularize box-level LHST. Here we conduct experiments to investigate this. The experimental results in Table 11 show that Box-level LHST brings clear performance improvements while including Image-level LHST further improves the detection performance, largely becuase Image-level LHST provides stable supervision to regularize Box-level LHST, i.e., Image-level LHST is much more stable as it uses the image-level proposal while the pseudo box labels in Box-level LHST (e.g., the location of pseudo boxes) vary along training iterations and are not very stable.

Table 11: **Ablation studies of Language Hierarchical Self-training.** The experiment setup is zero-shot cross-dataset object detection over Object365 dataset.

Method	Language Hierarchical Self-training (LHST)		AP50
	Box-level LHST	Image-level LHST	
Box-Supervised [15]			26.5
Detic [15]			29.4
	✓		30.5
	✓	✓	31.3

Parameter Studies for Language Hierarchical Self- training (LHST). In generating pseudo box labels in LHST, we filter out a prediction if its max confidence score is lower than the threshold t. We study the threshold t by changing it from 0.65 to 0.85 with a step of 0.05. Table 12 reports the experimental results on zero-shot transfer object detection over object365 dataset. We can observe that the detection performance is not sensitive to the threshold t.

Table 12: **Parameter Studies for Language Hierarchical Self- training (LHST)** on zero-shot transfer object detection over object365 dataset. We study the thresholding parameter t used in generating pseudo box labels in LHST.

Threshold t	0.65	0.70	0.75	0.80	0.85
AP50	31.1	31.3	31.3	31.3	31.2

Analysis of discrepancies of the taxonomies of image vs. box categories. We analyze the mismatch between image-level and box-level categories and how much it could affect the detection performance. As Table 13 shows, the mismatch between image-level and box-level categories varies across datasets, and the proposed DetLH improves more with the increase of mismatch levels.

Table 13: **Analysis of discrepancies of the taxonomies of image vs. box categories.**

ImageNet-21K	Mismatch Ratio	Baseline (AP50)	DetLH (AP50)	Δ
v.s. Cityscapes	0.13	47.1	50.3	+3.2
v.s. DETRAC	0.25	39.2	44.0	+4.8
v.s. MIO-TCD	0.27	20.6	24.5	+3.9
v.s. African Wildlife	0.50	80.9	87.2	+6.3
v.s. Vistas	0.67	35.6	44.0	+8.4
v.s. Arthropod Detection	0.86	36.7	49.0	+12.3

How effective DetLH deals with noisy labels. We conduct ablation studies to analyze how effective DetLH deals with noisy labels. Specifically, we compare DetLH with and without using reliability scores (the latter means uniform category weights) over Object365. As Table 14 shows, including the adaptive weighting mechanism (i.e., reliability scores) helps mitigate the label noises effectively.

Table 14: **How effective DetLH deals with noisy labels.**

Method	AP50
DetLH without reliability score	31.8
DetLH	32.5

The impact of using proxy vocabulary. We conduct experiments on Object365 dataset to compare LHPG with CLIP embeddings only and LHPG with both CLIP embeddings and proxy vocabulary. As Table 15 shows, using a proxy vocabulary performs clearly better, demonstrating its effectiveness in narrowing the distribution gap between training and test labels.

Table 15: **The impact of using proxy vocabulary.**

Method	AP50
Baseline	29.4
LHPG (CLIP embeddings only)	30.0
LHPG (CLIP embeddings + proxy vocabulary)	31.0

Comparisons with other semi-supervised WSOD methods. We conduct experiments on Object365 dataset to compare our DetLH and [88, 89, 90, 91]. As Table 16 shows, DetLH clearly outperforms [a,b,c,d].

Table 16: **Comparisons with other semi-supervised WSOD methods.**

	Baseline	[88]	[89]	[90]	[91]	DetLH
AP50	29.4	29.5	29.9	29.8	29.6	32.5

C Additional Comparison

Comparison with RKD [17]. We note that RKD [17] explores Region-based Knowledge Distillation to better distill knowledge from the CLIP model for weakly-supervised object detection. In this paragraph, we compare our DetLH (i.e., the self-training based method) with RKD (i.e., the knowledge distillation-based method) on zero-shot cross-dataset object detection over Object365 dataset. Table 17 reports the results, which show that our DetLH clearly outperforms RKD [17], indicating the effectiveness the proposed designs in DetLH for zero-shot cross-dataset object detection.

Table 17: Comparison with RKD [17] on zero-shot cross-dataset object detection over Object365 dataset.

Method	AP
RKD [17]	22.3
DetLH (Ours)	23.6

Discussion and comparison with visual grounding-based detection methods [92, 93]. We note that GLIP [92] and DetCLIP [93] explore extra visual grounding data to train a open-vocabulary detector. In this paragraph, we compare our DetLH (i.e., the WSOD-based method) with [92, 93] (i.e., the visual grounding-based method) from the perspective of detection efficiency. Table 18 reports the results of run time in millisecond (the run times of GLIP [92] and DetCLIP [93] are acquired from [93].). It shows that our DetLH (i.e., the WSOD-based method) runs much more efficient than the visual grounding-based detection methods (i.e., GLIP [92] and DetCLIP [93]), largely because the visual grounding-based detection methods [92, 93] include a text encoder in their networks. Note we did not compare our DetLH (i.e., the WSOD-based method) with the visual grounding-based method [92, 93] from the perspective of detection accuracy becuase these two types of detection methods use very different training data, e.g., [92] and [93] use visual grounding data. In addition, the WSOD techniques are basically complementary to the visual grounding-based detection techniques [92, 93] because the visual grounding data could be used to further improve the WSOD-based detectors [94]. Similar to [94], we leave this as our future research.

Other image-level supervision. We follow Detic [15] to build our proposed DetLH. Therefore, similar to Detic [15], our DetLH can also seamlessly incorporate the free-form caption text as the image-level supervision, i.e., by using the language embeddings of image captions as the detection classifier when training on image-text pair data [15]. In this way, we could further incorporate LAION dataset [95] that includes 400 million image-text pair data into detector training for learning more generalizable object detectors. On the other hand, training over large-scale LAION dataset is computation-intensive and thus we leave this as our future work.

Comparison with other detection methods [92, 96, 97, 93, 98, 99]. We didn't compare with these methods [92, 96, 97, 93, 98, 99] in the main manuscript because they focus on different topics with different objectives,

Table 18: Efficiency comparison of WSOD-based and visual grounding-based detection methods.

Method	Types	Run time (ms)
GLIP [92] w Swin-T	Visual Grounding-based	8333.3
DetCLIP [93] w Swin-T	Visual Grounding-based	434.7
DetLH w Swin-B (ours)	WSOD	46.0

training data, backbones and benchmarks. Instead, we follow Detic [15] as both our DetLH and Detic belong to and are claimed as weakly-supervised object detection (WSOD), aiming to using large-scale images and classes (i.e., ImageNet-21K and LVIS) to train a general detector that can work on any detection scenarios. However, [92, 96, 97, 93, 98, 99] are not for WSOD: GLIP and DetCLIP [92, 93] introduce visual grounding and studies how to use grounding data for detection; OWL-ViT [97] focuses on fine-tuning CLIP with standard detection datasets; OmDet and UniDet [98, 99] focus on training with multiple detection datasets. We still managed to benchmark with [92, 96, 97, 93, 98, 99], i.e., GLIP [92], GLIPv2 [96], OWL-ViT [97], DetCLIP [93], OmDet [98] and UniDet [99]. As our method and [92, 96, 97, 93, 98, 99] use various different datasets in evaluations, the benchmarking below is on the shared one, i.e., Pascal VOC (in AP).

GLIP-L	GLIPv2-B	OWL-ViT	DetCLIP	OmDet	UniDet	Ours
61.7	62.8	60.3	56.7	60.8	60.1	64.4

Note Florence [100] is not included as it is a foundation model that uses a very large backbone (CoSwin-H) and very large customized training data (FLD-900M and FLOD-9M). Besides, we compared our DetLH with GLIP [92] and DetCLIP [93] as in Table 18, which shows DetLH (i.e., the WSOD-based method) runs much more efficient (about 10 times) than the visual grounding-based detection methods (i.e., GLIP and DetCLIP).

Comparison on ODinW [92]. We note that ODinW [92] also benchmarks cross-dataset generalization. We benchmark on ODinW and the results below (averaged on 35 datasets in ODinW) show that our DetLH works effectively on ODinW. Note GLIP obtains higher accuracy because it introduces visual grounding and involves Language Encoder in inference. Without those extra modules, our DetLH runs much faster (over 10 times) than GLIP as discussed in Table 18.

WSDDN	YOLO9000	DLWL	Detic	GLIP	OmDet	Ours
14.9	16.3	17.1	17.3	19.7	16.0	18.2

Open-vocabulary benchmark. We did not benchmark on open-vocabulary LVIS/COCO (both divide a single-dataset vocabulary into base and novel classes to mimic and benchmark vocabulary generalization), because our work aims to leverage large-scale images and classes (i.e., 21K classes) to train a general detector that can work on any detection scenarios. Cross-dataset generalization benchmark fits this objective better and is more general and challenging than open-vocabulary benchmark that tackles base and novel classes within a single dataset.

Comparison with class hierarchy methods [101, 70] on OpenImages [102]. We benchmark with other methods that use class hierarchy, including [101] and [70]. As shown below, our DetLH performs clearly better than hierarchy-aware losses in [101, 70] due to our designed co-regularization as detailed in the manuscript. Note we use OpenImages V7 (Oct 2022) instead of 2019 version.

WSDDN	YOLO9000	DLWL	Detic	[101]	[70]	Ours
14.8	15.8	16.2	16.4	16.5	16.5	17.6

Results of WSDDN, YOLO9000 and DLWL. Note Detic implemented WSDDN, YOLO9000 and DLWL and we directly adopted Detic's implementation in evaluations. The gains in our reported results are different as we evaluated on more challenging datasets and benchmark: 1) The results in our Table 7 in the main manuscript are averaged over 14 datasets while those in Table 1 of Detic are on a single dataset LVIS; 2) Our Table 7 in the main manuscript is cross-dataset generalization benchmark while Table 1 in Detic is open-vocabulary LVIS.

D Additional Qualitative Result and Comparison

We provide qualitative results of zero-shot cross-dataset object detection for various detection tasks. As shown in Figures 3- 7. our DetLH produces good detection results consistently across different detection tasks, showing DetLH still works effectively under large cross-dataset gaps in data distribution and vocabulary.

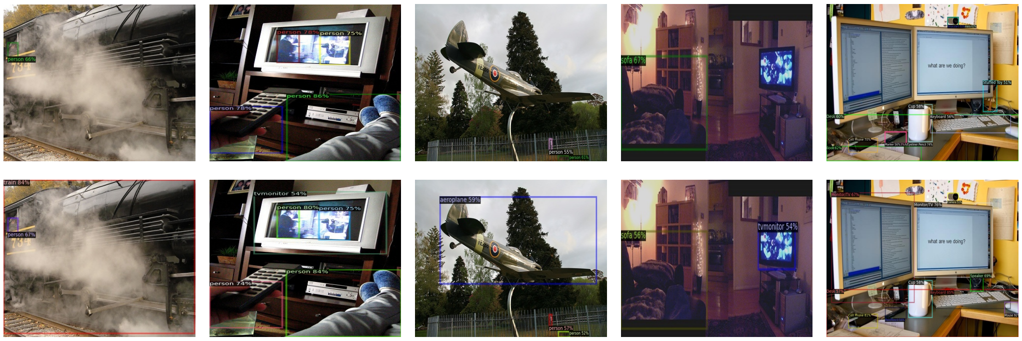


Figure 3: Qualitative results of DetLH over zero-shot cross-dataset object detection for common objects. Zoom in for details. Top: Detic [15]. Bottom: DetLH (Ours).

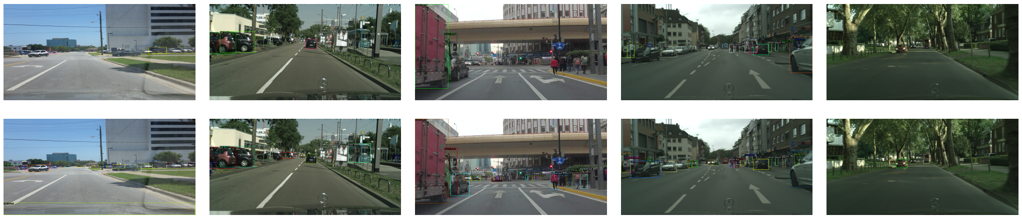


Figure 4: Qualitative results of DetLH over zero-shot cross-dataset object detection for autonomous driving. Zoom in for details. Top: Detic [15]. Bottom: DetLH (Ours).

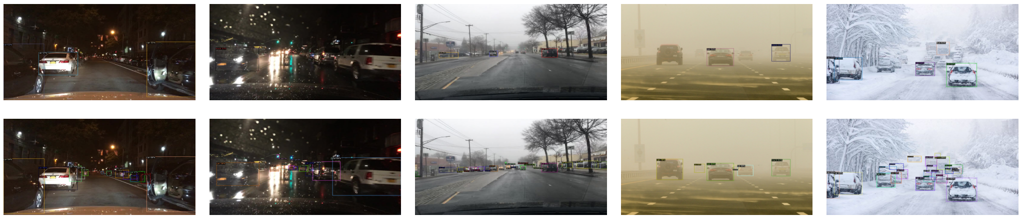


Figure 5: Qualitative results of DetLH over zero-shot cross-dataset object detection under different weather and time-of-day conditions. Zoom in for details. Top: Detic [15]. Bottom: DetLH (Ours).

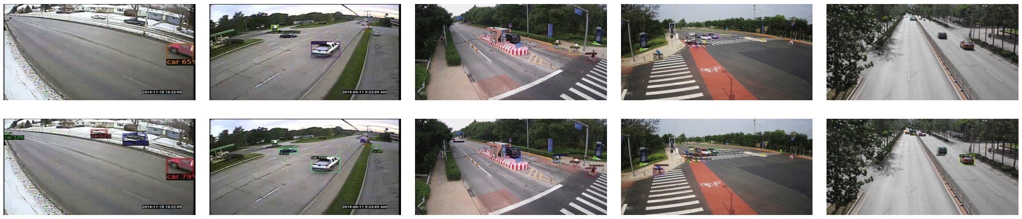


Figure 6: Qualitative results of DetLH over zero-shot cross-dataset object detection for intelligent surveillance. Zoom in for details. Top: Detic [15]. Bottom: DetLH (Ours).

E Broader Impacts and Limitations

Broader Impacts. This work strives for exploiting weakly-supervised object detection (WSOD) to learn generalizable detectors by addressing the image-level label mismatch issue. We propose to incorporate image-level supervision with self-training for learning generalizable detectors, aiming to benefit from self-training while effectively making use of image-level weak supervision. Our proposed technique provides great advantages by avoiding the need of massive object-level annotations and allowing learning effective and generalizable detectors with image-level supervision. It thus makes a very valuable contribution to the computer vision

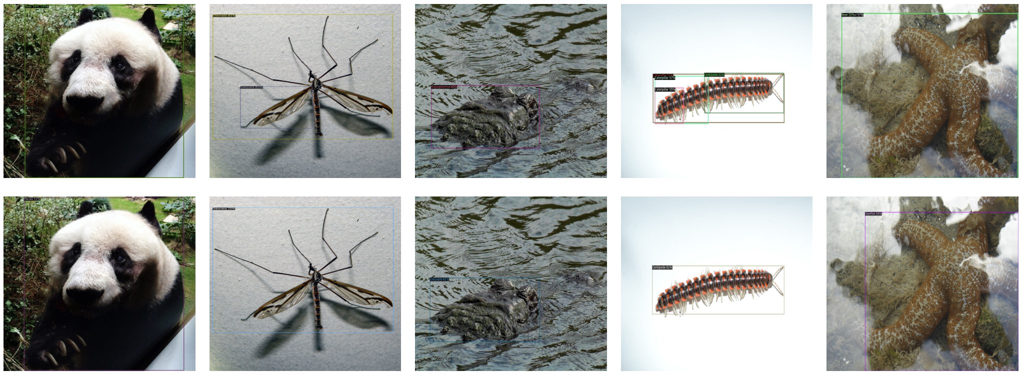


Figure 7: Qualitative results of DetLH over zero-shot cross-dataset object detection for Wildlife Detection. Zoom in for details. Top: Detic [15]. Bottom: DetLH (Ours).

research community by providing a novel and efficient weakly-supervised object detection method. The feature of scaling detectors with image-level labels enables effective and generalizable detectors that could work well in various downstream tasks, broadening the applicability of object detectors significantly.

Limitations. As discussed in Sections 3 and 4 of the main text and Section Dataset and Implementation Details in the appendix, our proposed WSOD method adopts ImageNet-21K with image-level labels to scale up detectors. It avoids the need of massive object-level annotations and allowing learning effective and generalizable detectors with image-level supervision. At the other end, we could further scale up detector training by involving the recent image-text pair data for WSOD training, which may further improve the performance significantly. We will investigate how to involve the recent image-text pair data for WSOD training in our future work.

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 accurately describe the paper's contributions and scope.

   Guidelines:

   - The answer NA 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 NA 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: We discussed the limitations of the work in Section E of the Appendix.

   Guidelines:

   - The answer NA 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: [NA]

   Justification: The paper does not include theoretical results.

   Guidelines:

   - The answer NA 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: We provided detailed instructions for reproducing the main experimental results in Section 3 Method and Section 4 Experiment including the details of the proposed framework, and the datasets, base models and the parameters used for experiments in Section Dataset and Implementation Details in the appendix.

   Guidelines:

   - The answer NA 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]

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provided the detailed implementation details in Section Dataset and Implementation Details in the appendix.

   Guidelines:

   - The answer NA 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: [NA]

   Justification: We conducted the experiments with multiple runs and did not observe clear variance.

   Guidelines:

   - The answer NA 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: We provided sufficient information on the computation resources required for reproduce the experiments in Section Dataset and Implementation Details in the appendix.

   Guidelines:

   - The answer NA 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 conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA 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: [Yes]

    Justification: We discussed the broader impacts of the work in Section E of the Appendix.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA 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: We properly credited the original owners of assets used in the paper and properly respect their license and terms of use.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA 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: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA 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.