

DetCLIPv2: Scalable Open-Vocabulary Object Detection Pre-training via Word-Region Alignment

Lewei Yao^{1,2}, Jianhua Han², Xiaodan Liang^{3†}, Dan Xu¹,
Wei Zhang², Zhenguo Li², Hang Xu^{2†}

¹Hong Kong University of Science and Technology, ²Huawei Noah's Ark Lab

³Shenzhen Campus of Sun Yat-Sen University

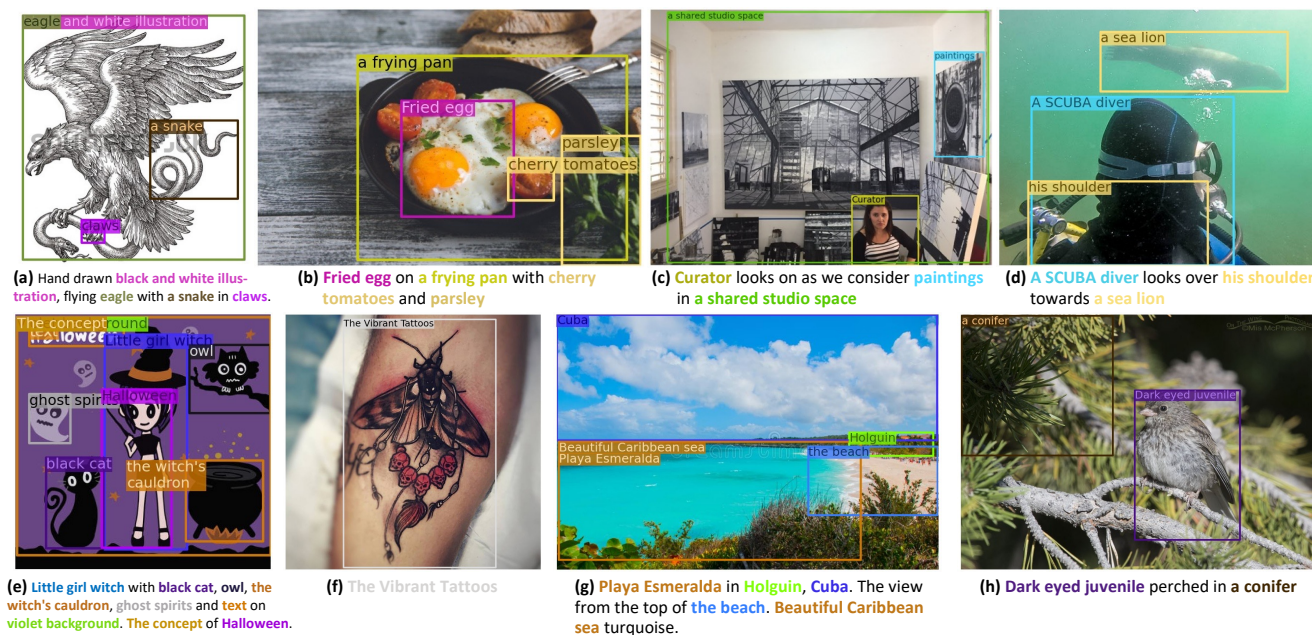


Figure 1. Visualizations of DetCLIPv2 for open-vocabulary word-region alignment. DetCLIPv2 is able to detect broad concepts.

Abstract

This paper presents DetCLIPv2, an efficient and scalable training framework that incorporates large-scale image-text pairs to achieve open-vocabulary object detection (OVD). Unlike previous OVD frameworks that typically rely on a pre-trained vision-language model (e.g., CLIP) or exploit image-text pairs via a pseudo labeling process, DetCLIPv2 directly learns the fine-grained word-region alignment from massive image-text pairs in an end-to-end manner. To accomplish this, we employ a maximum word-region similarity between region proposals and textual words to guide the contrastive objective. To enable the model to gain localization capability while learning broad concepts, DetCLIPv2 is trained with a hybrid supervision from detection, grounding and image-text pair data under a unified data formulation. By jointly training with an alternating scheme and adopting low-resolution input for image-text pairs, DetCLIPv2 exploits image-text pair data efficiently and effectively: DetCLIPv2 utilizes $13\times$ more image-text pairs

than DetCLIP with a similar training time and improves performance. With 13M image-text pairs for pre-training, DetCLIPv2 demonstrates superior open-vocabulary detection performance, e.g., DetCLIPv2 with Swin-T backbone achieves 40.4% zero-shot AP on the LVIS benchmark, which outperforms previous works GLIP/GLIPv2/DetCLIP by 14.4/11.4/4.5% AP, respectively, and even beats its fully-supervised counterpart by a large margin.

1. Introduction

Traditional object detection frameworks [6, 40, 41, 62] are typically trained to predict a set of predefined categories, which fails to meet the demand of many downstream application scenarios that require to detect arbitrary categories (denoted as open-vocabulary detection, OVD). For example, a robust autonomous driving system requires accurate predictions for all classes of objects on the road [29]. Extending traditional object detectors to adapt these scenarios needs tremendous human effort for extra instance-level bounding-box annotations, especially for rare classes. To obtain an open-vocabulary detector without the expensive annotation process, the central question we should ask is:

[†]Corresponding author: xu.hang@huawei.com

liangxd9@mail.sysu.edu.cn,

where does knowledge about unseen categories come from?

Recent works [19, 49, 56] try to achieve open-vocabulary object detection by transferring knowledge from a pre-trained vision-language (VL) model [23, 38, 54]. E.g., ViLD [19] distills the CLIP’s [38] image embeddings of cropped proposals into the proposal features of a detection model. However, these solutions suffer from the domain gap problem: VL models are typically pre-trained with an image-level supervision using a fixed resolution input, which are not capable of recognizing objects with various scales in the detection task, especially for small objects.

Another line of work resorts to exploiting massive image-text pairs crawled from the Internet. To utilize the image-text pair data without instance-level annotation, approaches [16, 17, 22, 30, 53, 59] generate pseudo-bounding-box labels following a self-training paradigm [45] or based on a pre-trained VL model [38]. However, their final performance is restricted by the quality of pseudo-labels provided by a detector trained with limited human-annotated concepts or a VL model suffering from the aforementioned domain gap problem. Besides, using high-resolution inputs similar to detection data for massive image-text pairs will impose a huge computational burden on training, preventing us from further scaling up image-text pairs.

To address the above issues, we present DetCLIPv2, an end-to-end open-vocabulary detection pre-training framework that effectively incorporates large-scale image-text pairs. DetCLIPv2 simultaneously learns localization capability and knowledge of broad concepts without relying on a teacher model to provide pseudo labels. Specifically, we perform joint training with heterogeneous data from multiple sources, including detection [43], grounding [25] and image-text pairs [7, 44], under a unified data formulation. To enable image-text pairs without instance-level annotations to facilitate learning of detection, inspired by [54], we employ an optimal matching-based set similarity between visual regions and textual concepts to guide the contrastive learning. By alternating different types of data for training, we enable a “flywheel effect”: learning from detection data provides accurate localization, which helps extract representative regions for contrastive learning, while contrastive learning from image-text pairs helps recognize broader concepts, which further improves the localization of unseen categories. As the training goes on, the detector learns to locate and recognize increasingly rich concepts.

Furthermore, to relief the computation burden brought by large-scale image-text pairs, we adopt a low-resolution input for image-text pair data, which significantly improves the training efficiency. This is a reasonable design since the caption of image-text pair data typically describes only the main objects appearing in the image, which alleviates the necessity of high-resolution training.

Benefiting from the effective designs, DetCLIPv2

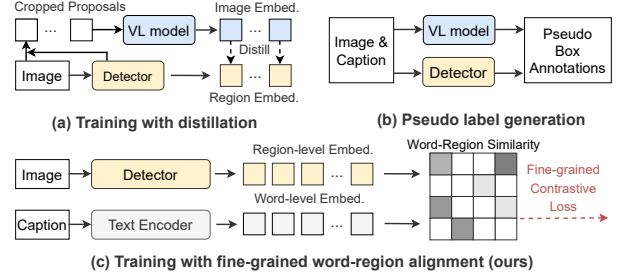


Figure 2. **Different OVD training paradigms.** (a) Distilling knowledge from a pre-trained VL model [19]. (b) Exploiting image-text pairs via pseudo labeling [30]. (c) Our end-to-end joint training eliminates complex multi-stage training schemes, allowing for mutual benefits in learning from different types of data.

demonstrates superior open-vocabulary detection performance and promising scaling behavior. E.g., compared to the prior work DetCLIP [53], DetCLIPv2 is able to exploit $13\times$ more image-text pairs while requiring only a similar training time. Using the vanilla ATSS [58] as the detector, DetCLIPv2 with Swin-T backbone achieves 40.4% *zero-shot* AP on the LVIS [20] benchmark, surpassing previous works GLIP [30]/GLIPv2 [57]/DetCLIP [53] by 14.4/11.4/4.5% AP, respectively. DetCLIPv2 also exhibits great generalization when transferring to downstream tasks, e.g., it achieves SoTA fine-tuning performance on LVIS and ODinW13 [30]. We present a possibility of achieving open-world detection by incorporating large-scale image-text pairs and hope it will enlighten the community to explore a similar successful trajectory to CLIP [38].

2. Related Work

Vision-Language Pre-training (VLP). Conventional vision-language models are designed to serve a specific task, e.g., VQA [2, 18, 27, 31] and image captioning [1, 33, 50, 55], etc. Recently, there has been a trend to develop generic vision-language representation learning systems by exploiting large-scale low-cost image-text pairs. For example, CLIP [38] and ALIGN [23] perform cross-modal contrastive learning on millions of image-text pairs and achieve impressive zero-shot image classification performance. The most relevant work to our approach is FILIP [54], which proposes a cross-modal late interaction mechanism based on a word-patch similarity to better facilitate image-text alignment. However, it is non-trivial to leverage the idea to construct an open-vocabulary detection system, for which our approach provides a solution.

Open-vocabulary object detection (OVD) emerges recently as a more general and practical paradigm to detect objects of unbounded concepts. Inspired by the success of vision-language pre-training, recent works [19, 49, 56, 59] propose to transfer knowledge of a pre-trained VL model (e.g., CLIP [38]) into a detector. Another effective idea is

to use a wider source of training data. E.g., [16, 17, 22, 30, 53] incorporate low-cost image-text pairs to expand domain coverage via a pseudo labeling process. XDETR [5] integrates a standard contrastive learning in VLP [23, 38] to learn image-to-text alignment. Detic [60] turns to solve a large-vocabulary detection problem by directly assigning classification labels to the max-size region proposals. Unlike previous works, our approach targets on building an end-to-end framework that effectively learns word-region alignment from massive image-text pairs without relying on a teacher model.

Semi-supervised Object Detection (SSOD) methods [45, 46, 51, 63] aim to improve object detection systems by exploiting unlabeled data on the basis of some available labeled data. Although effective in improving performance, these methods still assume a closed-domain setting where the categories in unlabeled data should be covered by labeled data. On the other hand, **Weakly-supervised Object Detection (WSOD)** methods [3, 10, 36] aim to establish localization-capable detectors by leveraging image-level labels, which also require a set of pre-defined categories. Differing from methods in these fields, our approach considers a more challenging open-domain setting and targeting on establishing an open-world detector by learning unlimited concepts from massive image-text pairs.

3. The Proposed Approach

An overview framework of the proposed approach is illustrated in Figure 3. To construct a robust open-world object detection system, DetCLIPv2 incorporates data from different sources, i.e., detection, grounding, and image-text pairs, for pre-training. We first introduce a unified paralleled data formulation enabling a training with heterogeneous supervisions (Sec. 3.1). To utilize image-text pairs without instance-level annotations, we introduce a fine-grained contrastive learning that automatically aligns textual words and visual regions (Sec. 3.2). Finally, we introduce the model architecture/training objective (Sec. 3.3) and the joint-training details (Sec. 3.4).

3.1. A Unified Data Formulation

Following DetCLIP [53], we use a *paralleled formulation* to unify the formats of data from different sources. Specifically, we formulate each data sample as a triplet: $(x^I, \{\mathbf{b}_i\}_{i=1}^N, \{t_j\}_{j=1}^M)$, where $x^I \in \mathbb{R}^{3 \times h \times w}$ is the image, $\{\mathbf{b}_i\}_{i=1}^N$ denote a set of bounding box annotations and $T = \{t_j\}_{j=1}^M$ denote a set of concept names, respectively. The triplet is constructed for different types of data differently:

- **Detection.** T is constructed from a sampled category names of the dataset, which consists of categories appearing in the image and additional randomly-sampled negative categories. To explicitly provide the relationships between various concepts, We apply *concept en-*

richment [53] during both training and testing phases, i.e., each t_j is obtained by concatenating its category name with the corresponding definition.

- **Grounding.** We first extract noun phrases (provided in annotations) from the original caption to form a positive concept set $T_{pos} = \{t_j\}_{j=1}^{|pos|}$. To provide enough negative concepts for learning, we further randomly sample a negative concept set $T_{neg} = \{t_j\}_{j=1}^{|neg|}$ that does not contained in the caption (i.e., $T_{pos} \cap T_{neg} = \emptyset$) from a constructed *concept dictionary* [53]. The final category name set is formed by $T = T_{pos} \cup T_{neg}$.
- **Image-text pairs.** As instance-level annotation is not available, we have $\{b_i\}_{i=1}^N = \emptyset$. T consists of the original caption and noun phrases extracted from it¹.

For detection and grounding data, each \mathbf{b}_i is labeled with a concept t_j , which enables the learning of open-vocabulary object detection. We describe it as follows.

Open-vocabulary object detection. As illustrated in Figure 3, we use a dual-stream architecture which consists of an image encoder and a text encoder. The image encoder is an arbitrary-form object detector that takes an image x^I as the input and outputs a set of region proposals $P = \{\mathbf{p}_k\}_{k=1}^K$ (for one-stage detector, K equals to number of anchors), as well as their classification features $\mathbf{f}^P \in \mathbb{R}^{K \times D}$, where D is the feature dimension. For the text side, we treat each concept name t_j as a sentence and forward all $t_j \in T$ to the text encoder *separately* to obtain the sentence embeddings $\mathbf{f}^T \in \mathbb{R}^{M \times D}$. Following previous works [23, 38, 57], to increase the number of negative samples, we collect \mathbf{f}^T across a global batch and remove duplicate concepts contained in different samples in a batch, which gives a gathered text embedding $\mathbf{f}^{T_{batch}} \in \mathbb{R}^{M_B \times D}$, where M_B is the total number of concepts in a global batch after deduplication. Then we calculate the similarity matrix $S \in \mathbb{R}^{K \times M_B}$ between \mathbf{f}^P and $\mathbf{f}^{T_{batch}}$ by

$$S = \mathbf{f}^P (\mathbf{f}^{T_{batch}})^\top \quad (1)$$

When instance-level annotations are available, e.g., for detection and grounding data, we can construct a target matrix $G \in \{0, 1\}^{K \times M_B}$ following a ground-truth assignment process in conventional object detection frameworks [41, 48, 58], then the alignment loss $\mathcal{L}_{align}(S, G)$ (detailed in Sec. 3.3) can be calculated; while for image-text pairs where the instance-level annotation is not available, we elaborate our approach in the Sec. 3.2.

3.2. Learning from Image-text Pairs

Massive image-text pairs crawled from the Internet can provide rich knowledge for the learning of visual-language

¹We use NLP parser provided by Spacy [21] repository.

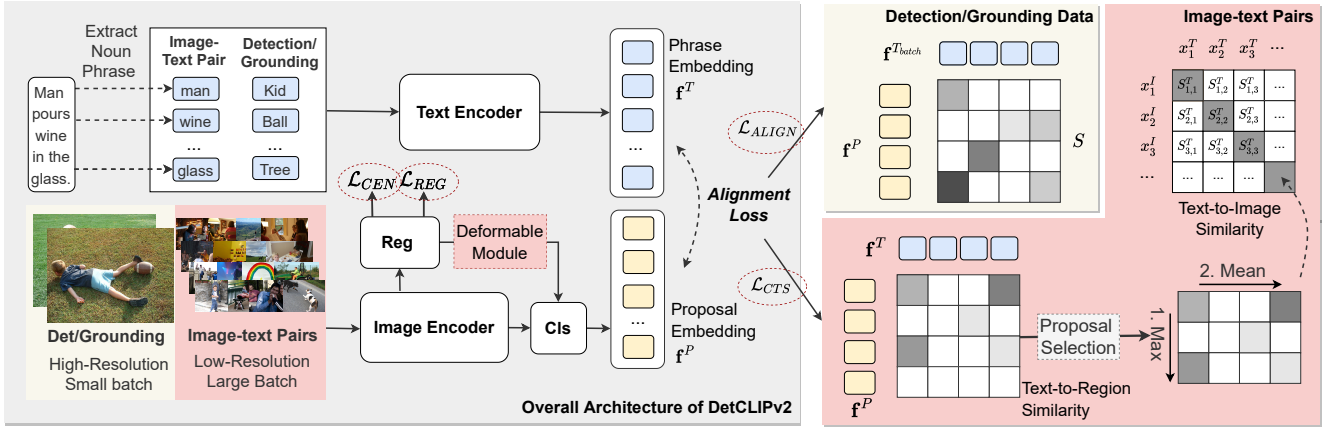


Figure 3. **Overall architecture of DetCLIPv2.** DetCLIPv2 performs a joint training with detection, grounding and image-pair data in an end-to-end manner. The architecture consists of an image encoder to extract region embeddings \mathbf{f}^P from an input image and a text encoder to compute word embeddings \mathbf{f}^T for the input noun phrases. For detection and grounding data, the learning is performed by aligning the word-region similarity matrix S to a target matrix constructed with instance-level annotations. For image-text pairs, we calculate an optimal match-based set similarity between \mathbf{f}^T and \mathbf{f}^P to guide the contrastive learning, enabling the learning of word-region alignment.

models. However, due to the lack of instance-level annotations, it is non-trivial to leverage image-text pairs to improve a dense prediction (e.g, object detection) learning system. Inspired by [54], we introduce a contrastive learning method to learn a fine-grained word-region correspondences without relying on instance-level annotation, which is described as follows.

Word-region alignment similarity. Given an image-text pair (x^I, x^T) , we extract a set of noun phrases $T = \{t_j\}_{j=1}^M$ from x^T and take $(x^I, \{t_j\}_{j=1}^M)$ as the input of the model. The image encoder generates a set of proposals $P = \{\mathbf{p}_k\}_{k=1}^K$ from x^I with their region features $\mathbf{f}^P \in \mathbb{R}^{K \times D}$ and the text encoder extracts text embeddings $\mathbf{f}^T \in \mathbb{R}^{M \times D}$ of $\{t_j\}_{j=1}^M$. Our word-region alignment contrastive learning is constructed based on the set similarity between P and T . Specifically, for j -th concept $t_j \in T$, we find its closest match in P by calculating

$$m_j = \arg \max_{0 < k \leq K} [\mathbf{f}^T]_j^\top [\mathbf{f}^P]_k, \quad (2)$$

where $[\mathbf{f}^P]_k$ is the k -th region feature in \mathbf{f}^P , and similar for $[\mathbf{f}^T]_j$. This operation can be interpreted as, for each concept we find a region that best fits its description. Then we calculate the text-to-image similarity s^T between x^I and x^T by aggregating all word-to-region similarities, i.e.,

$$s^T(x^I, x^T) = \frac{1}{M} \sum_{j=1}^M [\mathbf{f}^T]_j^\top [\mathbf{f}^P]_{m_j} \quad (3)$$

Note that the image-to-text similarity $s^I(x^I, x^T)$ can be calculated in a similar way. However, we exclude this part from our algorithm, since image-text pairs crawled from the Internet suffer from a severe *partial labeling* problem – for the vast majority of data, the text describes only a small fraction of the objects appearing in the image, i.e., most

of region proposals cannot find their corresponding match in the caption texts. Including image-to-text matching can result in a noticeable performance degradation, for which we give an ablation in Sec. 4.2.1.

Another reasonable consideration is that each textual concept should correspond to multiple regions. This design can be modeled by using a softmax-weighted-sum similarity between a textual concept and all visual regions, i.e.,

$$s^T(x^I, x^T) = \frac{1}{M} \sum_{j=1}^M \sum_{k=1}^K \frac{\exp(s_{j,k}/\tau_t)}{\sum_{i=1}^K \exp(s_{j,i}/\tau_t)} s_{j,k} \quad (4)$$

where $s_{j,k} = [\mathbf{f}^T]_j^\top [\mathbf{f}^P]_k$ is the similarity between j -th textual concept and k -th visual region, and τ_t is a temperature hyper-parameter to control sharpness of the softmax-based weights (when $\tau_t \rightarrow 0$, Eq. 4 degrades to Eq. 3). We investigate this design in Sec. 4.2.1.

Image-text contrastive loss. Based on the introduced word-region alignment similarity, a standard contrastive learning between image-text pairs can be performed [38]. Specifically, assume a batch of B image-text pairs $\{(x_i^I, x_i^T)\}_{i=1}^B$, the contrastive loss \mathcal{L}_{cts} is formulated as

$$\mathcal{L}_{cts} = \mathcal{L}_{T \rightarrow I} = -\frac{1}{B} \log \frac{\exp(s^T(x_i^I, x_i^T)/\tau)}{\sum_{j=1}^B \exp(s^T(x_j^I, x_j^T)/\tau)} \quad (5)$$

where $s^T(x_i^I, x_j^T)$ is text-to-image word-region alignment similarity between i -th image x_i^I and j -th text x_j^T , which is given by Eq. 3, and τ is a temperature to scale the logits. As discussed before, we only consider text-to-image contrastive loss. By incorporating the word-region alignment similarity, the contrastive loss helps the model learn fine-grained word-region correspondences automatically.

Proposal selection. Intuitively, we expect to select the most representative regions in an image to calculate similarities

with textual concepts. There are several schemes to accomplish this. For example, many detectors incorporate class-agnostic object scores in their designs, e.g., foreground classification score in RPN [41], centerness in FCOS [48], etc., which can be utilized to generate high-quality region proposals with good generalization [19, 26]. However, these approaches fail to take the textual information into consideration. To select regions valuable for contrastive learning, for each candidate region, we calculate its similarities with all textual concepts within a local batch, and use the maximum similarity as its objectness score. The benefits of this design are two-fold: (1) it selects the regions most relevant to the text description; (2) it selects hard negative concepts that described in other texts which may benefit the contrastive learning. With the objectness score, we select top- k proposals after a NMS operation. Different proposal selection strategies and the optimal k are studied in Sec. 4.2.1.

3.3. Model Architecture and Training Objective

Model architecture. Similar to DetCLIP [53], DetCLIPv2 is built using the vanilla ATSS [58] detector equipped with a transformer-based [38, 39] text encoder. We do not introduce additional heavy modules such as DyHead [11] adopted in [30, 57] and cross-modal fusion adopted in [15, 30, 57].

A special design is that we insert a lightweight deformable convolution [61] at the beginning of the classification head, which uses the features output by the regression head to calculate the spatial offsets and the modulation scalar, and aggregates the features from the backbone output. The motivation is that when training with image-text pairs, there is no supervision signal on the regression branch and therefore no gradient is generated. This design helps the gradient from the classification head to flow back to the regression head, so that the regression head also benefits from training with massive image-text pairs. I.e., learning a better spatial aggregation for backbone features helps regression head acquire better localization ability. We show this neat design provides substantial performance improvement when training with image-text pairs (see Sec. 4.2.2).

Training Objective. The overall objective of DetCLIPv2 can be formulated as

$$\mathcal{L} = \begin{cases} \mathcal{L}_{align} + \alpha\mathcal{L}_{reg} + \beta\mathcal{L}_{center}, & \text{for detection} \\ \mathcal{L}_{align}, & \text{for grounding} \\ \lambda\mathcal{L}_{cts}, & \text{for image-text pairs} \end{cases} \quad (6)$$

where \mathcal{L}_{align} is the alignment loss described in Sec. 3.1; \mathcal{L}_{cts} is the contrastive loss in Eq. 5; \mathcal{L}_{reg} and \mathcal{L}_{center} are regression and centerness losses, respectively; α , β and λ are loss weights. Following ATSS [58], we use focal loss for \mathcal{L}_{align} , GIoU loss [42] for \mathcal{L}_{reg} , and cross-entropy loss for \mathcal{L}_{center} . We remove the localization loss for grounding

Dataset	Type	Volume
Objects365 [43] (O365)	Detection	0.66M
GoldG [25]	Grounding	0.77M
CC15M	Image-text pairs	13M
(CC3M [44]+CC12M [7])		(3M+10M)

Table 1. A summary of training data. CC15M contains only 13M image-text pairs since some urls are invalid.

data due to its inaccurate bounding box annotations.

3.4. Joint Training

DetCLIPv2 performs a joint training with heterogeneous datasets. During training, we group data belonging to the same type for a global batch. At each iteration, we sample one type of data for training. Different data types are trained with different input resolutions and batch sizes. Specifically, we use a high-resolution input with a small batch size for detection and grounding data; while for image-text pairs, a low-resolution input with a large batch size is adopted, which helps increase the number of negative samples in contrastive learning and considerably reduce the training cost of massive image-text pairs.

4. Experimental Results

4.1. Implementation Details

Training Dataset. We use multiple datasets from different sources for training (Table 1). Specifically, for detection data, we use a sampled subset from Objects365v2 [43] dataset (denoted as O365) with 0.66M images; for grounding data, we use GoldG [25] with COCO [32] images removed, which results in a fairer zero-shot evaluation on LVIS [20]. For image-text pairs, we use 2 versions of Conceptual Captions (CC) datasets, i.e., CC3M [44] and CC12M [7] (together denoted as CC15M).

Training details. We use Swin-transformer [34] backbones for image encoder. For text encoder, the maximum token length is set to 16 for efficient training and inference. We initialize the text-encoder with a pretrained FILIP model [54]. 32/64 V100 GPUs are used for training Swin-T/L-based models, respectively. For detection and grounding data, we use input resolution 1333×800 with a batch size of 128/256 for Swin-T/L models (4 per card), respectively; and for image-text pairs, we use input resolution 320×320 with a batch size of 6144 (192/96 per card for Swin-T/L model). We set $\alpha = 2$ and $\beta = 1$ and $\lambda = 0.1$ in Eq. 6. Without otherwise specified, all models are trained with 12 epochs. More training details can refer to Appendix.

Evaluation benchmark. Following GLIP [30] and DetCLIP [53], we evaluate our method’s *zero-shot* performances on LVIS [20] with 1203 categories. *Fixed AP* [12] on LVIS minival5k are reported for ablation and comparison with other methods. To further study the generaliza-

tion ability of our method, we also evaluate with ODinW13 dataset [30, 57], which contains 13 downstream detection tasks with highly varied distributions. We focus on the GLIP protocol [30] rather than the ViLD protocol [19] that splits LVIS into seen/unseen categories, since the former is a stronger and more practical open-world setting that does not make any prior assumptions on downstream tasks while the latter still requires partial LVIS data for training.

4.2. Ablation Studies

4.2.1 Ablations for Image-text Contrastive Learning

We investigate key factors for our image-text contrastive learning to work in Table 2. The experiments are conducted with Swin-T-based model on O365+CC3M datasets.

Proposal selection strategy. Selecting representative regions is critical for image-text contrastive learning. Table 2a studies multiple class-agnostic objectness scores for selecting proposals, which includes foreground classification score [52] (row1), IoU score [24] (row2) and centerness [48] (row3). Except for centerness which is originally designed in ATSS [58], other 2 scores are predicted by plugging in an additional head after the regression branch. We consider 3 additional scores to utilize textual information: (1) sample-wised text similarity (row4), i.e., each region calculates the similarities with the textual concepts of the sample and the maximum similarity is used as the objectness score; (2) batch-wised text similarity (row5), i.e., the similarities are calculated between a region and textual concepts within a local batch, as described in Sec. 3.2; and (3) multiplying the batch-wised text similarity with the centerness score (row6), which is commonly adopted by conventional detectors [48, 58].

Among 3 class-agnostic objectness scores, centerness and IoU scores are superior to classification score, indicating that localization-based objectness scores provide better class-agnostic proposals. The result is consistent with the observations in [26]. Considering only the sample-wised text similarity performs worse than using class-agnostic scores, since the regions selected in this way make it easier to distinguish between positive and negative samples in the contrastive learning, thus reducing the learning efficiency. Batch-wise similarity addresses the problem by considering text similarities with negative samples and achieves the best performance of 31.3 AP. Further integrating centerness score results in a performance drop to 30.9 AP.

Word-region alignment strategy. Table 2b investigates word-region alignment strategies described in Sec. 3.2. Specifically, for fine-grained word-region alignment, 2 matching strategies are studied: (1) 1-to-1 match (row2), i.e., each textual concept is matched with its closest region and (2) 1-to-many match (row3), i.e., each textual concept calculates similarities with all regions, which is then aggregated through a softmax-weighted-sum operation. Besides,

we also study a coarse-grained image-text matching strategy proposed in [60] (row1). Specifically, it directly calculates the similarity between the max-size proposal of image and the entire caption of text. Both fine-grained word-region alignment strategies outperform the coarse-grained image-text alignment. Assigning each textual concept with the closest region reaches the best performance (31.3 AP) and substantially saves the GPU memory compared to the 1-to-many strategy, which allows a larger batch size to boost the contrastive learning.

Number of proposals k . Table 2d investigates the optimal k when selecting proposals. We vary k from 25 to 200. Using a large $k = 200$ results in too many low-quality candidates that slightly decreases the performance. A too small $k = 25$ leads to insufficient region extraction which causes a noticeable performance drop. A modest design with 100 proposals achieves the best performance.

Contrastive loss design. Table 2c performs ablation experiments on different sides of the image-text contrastive loss (Eq. 5). 3 designs are considered: (1). only image-to-text side loss; (2) only text-to-image side loss; and (3) bilateral loss. As discussed in Sec. 3.2, using only image-to-text contrastive loss can lead to a significant performance degradation (29.8 AP) due to the *partial labeling* problem of the image-text pair data. Excluding image-to-text contrastive loss can alleviate the problem and achieving a better performance of 31.3 AP.

Temperature and Loss weight. Table 2e and 2f study the optimal values of temperature τ in Eq. 5 and loss weight λ in Eq. 6, respectively. The default values of $\lambda = 1$, $\tau = 0.07$ commonly adopted in standard constastive learning methods [38, 54] perform poorly in our case. We use $\tau = 0.5$ and $\lambda = 0.1$ as our final setting.

4.2.2 Effectiveness of Deformable Module

Table 3 studies the effectiveness of the proposed deformable module described in Sec. 3.3. The deformable module effectively promotes the weakly supervised learning. Specifically, it presents negative effect when trained with strongly supervised detection data (row1 and 2), while demonstrating substantial performance improvement when incorporating grounding/image-text pair data without localization supervisions (row3 and 4). Besides, the lightweight deformable module introduces negligible computational cost in terms of training time.

4.2.3 Incorporating More Data Helps Learning

Table 4 reports the performance gains when scaling up the training data. With the proposed framework, incorporating more training data from different sources can consistently improve the performance. Compared to training with only Objects365, including CC3M effectively improves the overall AP from 28.6 to 31.3, especially for rare categories (from

#	Strategy	AP (r/c/f)
1	cls	28.4 (26.6/28.2/28.8)
2	IoU	30.1 (30.0/30.1/30.2)
3	centerness	30.2 (28.4/30.6/30.1)
4	text sim (S)	29.6 (24.9/29.5/30.5)
5	text sim (B)	31.3 (29.4/31.7/31.3)
6	+centerness	30.9 (30.2/31.1/30.8)

(a) **Proposal selection strategy.** Batch-wised text similarity generates better proposals for contrastive learning.

Top-k	AP (r/c/f)
25	30.6 (29.9/30.5/30.8)
50	30.8 (30.2/30.6/31.0)
100	31.3 (29.4/31.7/31.3)
200	30.8 (29.4/30.6/31.1)

(d) **Number of proposals.** We use $k = 100$.

#	Strategy	AP (r/c/f)	Memory
1	max-bbox	29.8 (28.5/39.5/30.4)	19.8 GB
2	1-to-1	31.3 (29.4/31.7/31.3)	20.6 GB
3	1-to-many	30.9 (31.3/30.7/31.1)	26.0 GB

(b) **Word-region matching strategy.** Matching each textual concept to the closest region is effective and memory-efficient.

τ	AP (r/c/f)
1	30.1 (28.4/30.3/30.3)
0.5	31.3 (29.4/31.7/31.3)
0.15	30.8 (29.6/31.0/30.9)
0.07	29.2 (27.5/29.0/29.6)

(e) **Temperature τ .** $\tau = 0.5$ works the best.

Design	AP (r/c/f)
text-to-image	31.3 (29.4/31.7/31.3)
image-to-text	29.8 (30.0/29.7/29.9)
bilateral	30.9 (30.0/31.5/30.5)

(c) **Contrastive loss design.** Excluding image-to-text contrastive loss can boost the performance.

λ	AP (r/c/f)
0.03	30.5 (29.6/30.0/31.0)
0.1	31.3 (29.4/31.7/31.3)
0.3	30.9 (29.4/31.5/30.7)
1	28.9 (27.7/28.8/29.3)

(f) **Contrastive loss weight.** We use $\lambda = 0.1$.

Table 2. **Ablation experiments for image-text contrastive learning.** The models are based on the Swin-T backbone and trained with O365+CC3M dataset. We report zero-shot *fixed* AP (%) [12] on LVIS minival5k [25]. r/c/f indicate AP of rare/common/frequent categories, respectively. Designs with higher overall AP (marked in gray) are selected as our final setting.

Pretrain-data	deform	AP (r/c/f)	iter time (s)
O365	✗	28.8 (26.0 / 28.0 / 30.0)	0.925
O365	✓	28.6 (24.2 / 27.1 / 30.6)	1.075
O365+GoldG+CC3M	✗	37.3 (34.1 / 36.9 / 38.2)	0.980
O365+GoldG+CC3M	✓	38.4 (36.7 / 37.9 / 39.1)	1.092

Table 3. **The deformable module** effectively improves the weakly-supervised learning while introducing negligible computational cost. 'iter time' is the training time per iteration.

Pretrain-data	AP (r/c/f)
O365	28.6 (24.2 / 27.1 / 30.6)
O365+CC3M	31.3 (29.4 / 31.7 / 31.3)
O365+GoldG+CC3M	38.4 (36.7 / 37.9 / 39.1)
O365+GoldG+CC15M	40.4 (36.0 / 41.7 / 40.0)

Table 4. **Incorporating more data from different sources** consistently improves the performance.

Model	Pretrain-data	Training time (GPU hours)	Training FPS
GLIP-T [30]†	O365+GoldG	7.4k (3.0k)†	1.6
DetCLIP-T [53]	O365+GoldG+YFCC1M	2.0k	4.1
DetCLIPv2-T	O365+GoldG+CC15M	2.1k	25.7

Table 5. **Training efficiency.** For DetCLIP, we directly use the result reported in the paper; while for GLIP, we calculate the training time based on the FPS provided in the paper. †: 7.4k is calculated based on the official implementation which trains 30 epochs, while 3.0k is obtained by converting it to our setting of 12 epochs.

24.2 to 29.4, +5.2 AP). GoldG helps significantly improve the overall AP to 38.4 thanks to its instance-level annotations. Including CC12M pushes the envelop further, achieving a 40.4 overall AP which already surpasses the performance of the fully-supervised method (see Table 6).

4.2.4 Training Efficiency

We develop DetCLIPv2 with several designs that facilitate training efficiency, including using low-resolution inputs for image-text pairs, limiting the maximum token length of

the text encoder to 16, etc. Table 5 compares the training efficiency of DetCLIPv2 with that of GLIP [30] and DetCLIP [53]. First, both DetCLIP and DetCLIPv2 are more efficient than GLIP due to the lightweight architecture design, as described in Sec. 3.3. Besides, DetCLIPv2 is much faster than DetCLIP: it exploits $13\times$ more image-text pairs than DetCLIP with a similar training time, achieving more than $6\times$ FPS speed up (25.7 FPS v.s. 4.1 FPS). This indicates the great scaling property of our method and allows a possibility of incorporating a larger-scale image-text pairs to build a more powerful open-vocabulary detection system.

4.3. Main Results

4.3.1 Zero-shot Performance on LVIS

To compare with the existing works, We train DetCLIPv2 with the best setting reported in 4.2.1. We vary models' capacity by considering two backbones, i.e., swin-T and swin-L [34], denoted as DetCLIPv2-T/L, respectively. Table 6 reports the comparison with MDETR [25], GLIP [30], GLIPv2 [57], and DetCLIP [53] on zero-shot performance. For better demonstration, we also report the performances of the fully-supervised method on LVIS.

DetCLIPv2 outperforms the existing methods by a large margin. Compared to GLIP/GLIPv2, DetCLIPv2 uses a more lightweight backbone (without heavy DyHead [11] and cross-modal fusion) but still achieves better performances, e.g., DetCLIPv2-T outperforms GLIP-T/GLIPv2-T by 14.4/11.4 AP, respectively. Compared to DetCLIP, DetCLIPv2 achieves 4.5 (40.4 v.s. 35.9) and 6.1 (44.7 v.s. 38.6) AP performance gains for Swin-T- and Swin-L-based models, respectively. Despite using more training data, our total training cost is on par with DetCLIP [53], as reported in Table 5. *Notably, our models beat their fully-supervised counterparts in a zero-shot manner*, e.g., +6.8/0.8 AP for Swin-T- and Swin-L-based models, respectively. Especially, due to the long-tailed property of LVIS, the im-

Method	Detector (Backbone)	Pre-Train Data	LVIS	
			AP	$AP_r / AP_c / AP_f$
MDETR [25]	DETR [6] (RN101)	GoldG+	24.2	20.9 / 24.3 / 24.2
Supervised	ATSS [58] (Swin-T)	LVIS	33.6	19.7 / 32.4 / 37.2
GLIP-T [30]	DyHead [11] (Swin-T)	O365,GoldG,Cap4M	26.0	20.8 / 21.4 / 31.0
GLIPv2-T [57]	DyHead [11] (Swin-T)	O365,GoldG,Cap4M	29.0	- / - / -
DetCLIP-T [53]	ATSS [58] (Swin-T)	O365,GoldG,YFCC1M	35.9	33.2 / 35.7 / 36.4
DetCLIPv2-T (ours)	ATSS [58] (Swin-T)	O365,GoldG,CC15M	40.4	36.0 / 41.7 / 40.0
Supervised	ATSS [58] (Swin-L)	LVIS	43.9	30.6 / 43.6 / 46.6
GLIP-L [30]	DyHead [11] (Swin-L)	O365,GoldG,Cap24M	37.3	28.2 / 34.3 / 41.5
DetCLIP-L [53]	ATSS [58] (Swin-L)	O365,GoldG,YFCC1M	38.6	36.0 / 38.3 / 39.3
DetCLIPv2-L (ours)	ATSS [58] (Swin-L)	O365,GoldG,CC15M	44.7	43.1 / 46.3 / 43.7

Table 6. **Zero-shot performance** on LVIS minival5k [25]. *Fixed AP [12]* is reported. DetCLIPv2 achieves SoTA performance.

Method	LVIS	ODinW13
	AP ($AP_r/AP_c/AP_f$)	average AP
GLIP-T [30]	-	64.9
GLIPv2-T [57]	[†] 50.6 (- / - / -)	66.5
DetCLIPv2-T (ours)	50.7 (44.3/52.4/50.3)	68.0
GLIP-L [30]	-	68.9
GLIPv2-B [57]	[†] 57.3 (- / - / -)	69.4
GLIPv2-H [57]	[†] 59.8 (- / - / -)	70.4
DetCLIPv2-L (ours)	60.1 (58.3/61.7/59.1)	70.4

Table 7. **Fine-tuning performance.** *Fixed AP [12]* on LVIS minival5k [25] and average AP on ODinW13 [30] are reported. Numbers with [†] mean mask annotation are used for training.

provements over rare categories are significant, i.e., more than 10 AP improvements can be observed on both models.

4.3.2 Transfer Results with Fine-tuning

We study the transferability of DetCLIPv2 by fine-tuning it on down-stream tasks. Specifically, we conduct full-shot fine-tuning on LVIS [20] with 1203 categories and ODinW13 [30, 57] containing 13 detection tasks. The results are shown in Table 7. Without using mask annotation for training, DetCLIPv2 slightly outperforms GLIPv2 on LVIS, e.g., 50.7 AP of DetCLIPv2-T v.s. 50.6 AP of GLIPv2-T. On ODinW13, DetCLIPv2-T demonstrates superior performance compared to GLIP-T/GLIPv2-T, outperforming GLIP-T/GLIPv2-T by 3.1/1.5 average AP, respectively; and DetCLIPv2-L with Swin-L backbone achieves the same performance (70.4 average AP) with GLIPv2-H that uses a heavier Swin-H backbone.

4.4. Visualizations and Analyses

Visualization of word-region alignment. Figure 1 visualizes the learning results of word-region alignment on image-text pairs in CC12M [7]. For each textual concepts, we find its best matching with the highest similarity to it, as described in Sec. 3.2. Our approach achieves accurate word-region alignment (on instance-level) with great generalization, which is demonstrated by several aspects: (1) it suc-

Pretrain-data	AR (s/m/l)
O365	44.9 (35.2 / 52.9 / 62.0)
O365+GoldG	57.2 (42.1 / 67.0 / 76.2)
O365+GoldG+CC15M	59.4 (44.5 / 69.4 / 76.8)

Table 8. **Average recall (AR)** across 0.5-0.95 IoU on LVIS. s/m/l denote for small/medium/large objects, respectively.

cesses to recognize concepts that do not covered by detection datasets, e.g., ‘parsley’ in case (b); (2) it works for images with *natural distribution shifts* [47], e.g., the sketch image in case (a) and the cartoon image in case (e); and (3) it is capable of resolving co-reference expressions, e.g., the ‘juvenile’ in case (h) refers to ‘young bird’ and ‘curator’ in case (c) refers to a person. These capabilities are critical for open-world detectors but cannot be reflected well in the commonly adopted evaluation benchmarks like LVIS [20].

Learning from image-text pairs benefits localization. Table 8 provides more evidences showing that learning from image-text pairs also helps localization. Specifically, we evaluate the average recall across 0.5-0.95 IoU on LVIS and compare models trained with different data. Incorporating image-text pairs brings a significant and comprehensive recall improvements for small, medium, and large objects.

5. Conclusion

Learning from massive Internet-crawled data to achieve generic visual/language understanding systems has always been an important topic for both NLP [4, 14, 39] and CV [23, 28, 38] fields. In this paper, we present DetCLIPv2, a unified end-to-end pre-training framework towards open-vocabulary object detection. By employing a best-matching set similarity between regions and words to guide the contrastive objective, we effectively leverage massive image-text pairs to serve the object detection task. Experiments demonstrate DetCLIPv2’s superior open-vocabulary performance and its broad domain coverage. Our method provides a possible way to achieve open-world detection by further scaling up image-text pairs and we leave it to future work.

Acknowledgements We acknowledge the support of MindSpore², CANN (Compute Architecture for Neural Networks) and Ascend AI Processor used for this research.

A. Limitations

Our method provides a possible way to achieve open-world detection by scaling up image-text pairs. However, its localization capability still strongly relies on bounding box annotations provided in detection data. To improve the generalization of localization, designing architectures like [26] for robust open-world region proposals is a promising direction for future work. Furthermore, image-text pairs crawled from the Internet are noisy and suffer from severe incomplete descriptions, which undermines the learning efficiency of word-region alignment and requires further designs like [28] for ameliorating data quality. When further scale up image-text pairs to overwhelm detection data, imbalanced training can potentially hurt the performance, which also calls for a future exploration.

B. More Implementation Details

In this section, we provide more implementation details for both pre-training and fine-tuning experiments.

Pre-training details. We pre-train DetCLIPv2 with AdamW [35] optimizer. The learning rate first warms up linearly to a peak value ($2.8\text{e-}4/4\text{e-}4$ for Swin-B/L based models, respectively) and then decays following a cosine annealing schedule, where the peak lr values are obtained using a square root scaling rule: $lr = base_lr \times \sqrt{\frac{batchsize}{16}}$, where $base_lr = 1\text{e-}4$. We initialize the text encoder with a pre-trained FILIP [54] model and reduce the learning rate of text encoder by a factor of 0.1 to preserve the language knowledge obtained in FILIP’s pre-training. To save the GPU memory cost and allow a large batch size for contrastive learning, we adopt automatic mixed-precision [37] and gradient checkpointing [9] for training. Mmdetection [8] repository is used for implementation. Table 9 summarizes the detailed training settings.

Fine-tuning details. We fine-tune DetCLIPv2 on 2 datasets, i.e., LVIS [20] and ODinW13 [30]. For LVIS, we follow most settings of pre-training except that we use a smaller learning rate and the total epochs are set to 24 (i.e., 2x schedule). Table 11 summarizes the detailed setting of fine-tuning LVIS. For ODinW13, since the number of training samples of different datasets varies a lot, we cannot set the same training epoch for all datasets. To avoid tedious hyper-parameter turning and ensure a sufficient training for all datasets, we adopt a long training schedule with early stop mechanism. Specifically, we assign a maximum training epoch with an auto-step learning rate schedule. We

²<https://www.mindspore.cn/>

Config	Value
GPUs (V100)	32(T)/64(L)
training epochs	12
loss weight	$\alpha = 1, \beta = 2, \lambda = 0.1$
optimizer	AdamW [35]
optimizer momentum	$\beta_1 = 0.9, \beta_2 = 0.999$
lr for image encoder	$2.8\text{e-}4(\text{T})/4\text{e-}4(\text{L})$
lr for text encoder	$2.8\text{e-}5(\text{T})/4\text{e-}5(\text{L})$
weight decay	0.05
warmup iters	1000
learning rate schedule	cosine decay
batch size (det/grounding)	128(T)/256(L)
batch size (image text pairs)	6144
input resolution (det/grounding)	1333×800
input resolution (image-text pairs)	320×320
drop path of visual backbone	0.2
max text token length	16
# of concepts M (det)	150
# of concepts M (grounding)	100
label smooth for contrastive loss	0.1
augmentation	multi-scale training, random flip

Table 9. **Detailed pre-training settings** of DetCLIPv2. T/L in parentheses denote Swin-T/L models, respectively. Det/grounding mean detection and grounding data, respectively.

Config	Value
GPUs (V100)	16
training epochs	24
optimizer	AdamW [35]
optimizer momentum	$\beta_1 = 0.9, \beta_2 = 0.999$
lr for image encoder	$4\text{e-}5$
lr for text encoder	$4\text{e-}6$
weight decay	0.05
warmup iters	1000
learning rate schedule	cosine decay
batch size	64
input resolution	1333×800
drop path of visual backbone	0.2
# of concepts M	150
augmentation	multi-scale training, random flip

Table 10. **Detailed fine-tuning settings** for LVIS [20].

monitor the performance and decay the learning rate by 0.1 when the performance reaches a plateau for a tolerance of t_1 epochs. If the learning rate reaches a given minimum value and there is no performance improvement for t_2 epochs, the training exits. We use the same learning rate configuration for all datasets and *do not* search optimal hyper-parameters for each dataset separately.

Config	Value
GPUs (V100)	8
maximum training epochs	250
optimizer	AdamW [35]
optimizer momentum	$\beta_1 = 0.9, \beta_2 = 0.999$
lr for image encoder	$4e-5$
lr for text encoder	$4e-7$
weight decay	0.05
warmup iters	500
learning rate schedule	auto-step decay
lr decay tolerance t_1 (epochs)	5
training exit tolerance t_2 (epochs)	8
minimum lr to stop decay	$1e-8$
batch size	32
input resolution	1333×800
drop path of visual backbone	0.2
augmentation	multi-scale training, random flip

Table 11. **Detailed fine-tuning settings** for ODinW13 [30].

Input res	GPU Memory	Training time (GPU hours)	AP
224×224	14.1 GB	697.6	30.5
256×256	16.0 GB	729.6	30.5
320×320	20.7 GB	793.6	31.3
384×384	24.2 GB	876.8	31.5

Table 12. **Input resolution change of image-text pairs.** We use Swin-T-based model trained with O365+CC3M. Zero-shot AP on LVIS minival5k is reported. Using resolution of 320×320 (marked in gray) achieves the best trade-off between computational cost and model performance.

C. More Experimental Results

Effect of input resolution of image-text pairs. Reducing the input resolution of massive image-text pairs can significantly boost the training efficiency while may lead to performance degradation. Table 12 studies the effect of input resolution change of image-text pairs, where we conduct experiments with the Swin-T-based model on O365+CC3M and vary the resolution of CC3M data from 224×224 to 384×384 . Increasing the resolution from 256 to 320 leads to an obvious performance improvement (from 30.5 AP to 31.5 AP). However, further increasing it to 384 only brings limited performance gains and introduces considerable memory and training time overhead. Therefore, we choose 320×320 as our final setting.

Incorporating classification dataset. Our framework can be viewed as a more general design for weakly-supervised (WSOD) approaches, which eliminates the limit of pre-defined categories in traditional WSOD methods. By formulating classification data as a special type of image-text

#	Backbone	Pretrain-data	AP (r/c/f)
1	Swin-T	O365	28.6 (24.2/27.1/30.6)
2	Swin-T	O365+IN1k	30.4 (32.2 /29.4/30.9)
3	Swin-T	O365+CC3M	31.3 (29.4/ 31.7 / 31.3)
4	Swin-T	O365+GoldG+CC15M	40.4 (36.0/41.7/ 40.0)
5	Swin-T	O365+GoldG+CC15M +IN1k	40.6 (38.2 / 42.0 /39.9)
6	Swin-L	O365+GoldG+CC15M	44.7 (43.1/ 46.3 /43.7)
7	Swin-L	O365+GoldG+CC15M +IN1k	44.7 (43.8 /45.4/ 44.3)

Table 13. **Effect of incorporating IN21k data for training.** r/c/f indicate rare/common/frequent categories, respectively. Zero-shot AP on LVIS minival5k is reported.

pair data, our method is capable of incorporating it into training. Specifically, we use the category name as the caption for each image. To select region proposals, similar to image-text pair, we collect category names in a batch to calculate the similarity with a region and select the maximum value as the objectness score. Considering classification image typically contains only 1 main object, we select top $k = 1$ proposal. Finally, the contrastive loss for image-text pair is replaced with the cross entropy loss for classification, and we also set loss weight $\lambda = 0.1$.

We perform experiments on ImageNet1k [13] (denoted as IN1k) and show the results in Table 13. 2 settings are considered: 1. we train IN1k with only the detection data, i.e., O365; and 2. we incorporate IN1k into the final version of DetCLIPv2, i.e., all data including O365, CC15M, GoldG and IN1k are used. During training, we use the same image-text pair setting for IN1k and replicate IN1k by 3 times to make it have a similar size to CC3M.

First, incorporating classification data when using only detection data can significantly improve the performance from 28.6 to 30.4 (rows 1 and 2), especially for rare categories (from 24.4 to 32.2 AP, +8 AP), yet it is slightly worse than using CC3M in terms of overall AP (rows 2 and 3). However, when all data are used, the advantage of IN1k diminishes, i.e., it brings only 0.2 overall AP for Swin-T-based model (rows 4 and 5) and no performance gain is observed for Swin-L-based model (rows 6 and 7). Therefore, we exclude the classification data from our method to keep it as neat as possible.

More results on LVIS. To make a comprehensive comparison with the existing methods, we also evaluate DetCLIPv2 with the complete validation set of LVIS [20] (including 20k images with 1203 categories), on both zero-shot and fine-tuning settings. Table 14 exhibits the results. DetCLIPv2 outperforms GLIP and DetCLIP by a large margin for both T/L models, e.g., DetCLIP-T surpasses GLIP-T/DetCLIP-T by 15.6/4.4 AP, respectively. Besides, by pre-training on large-scale hybrid data and fine-tuning on LVIS, Det-

Method	Detector (Backbone)	LVIS (zero-shot)			LVIS (fine-tune)		
		AP	AP _r / AP _c / AP _f	AP	AP _r / AP _c / AP _f		
Supervised	ATSS [58] (Swin-T)	-	- / - / -	28.4	18.9 / 27.3 / 33.6		
GLIP-T [30]	DyHead [11] (Swin-T)	17.2	10.1 / 12.5 / 25.2	-	- / - / -		
DetCLIP-T [53]	ATSS [58] (Swin-T)	28.4	25.0 / 27.0 / 31.6	-	- / - / -		
DetCLIPv2-T (ours)	ATSS [58] (Swin-T)	32.8	31.0 / 31.7 / 34.8	43.7	40.2 / 42.7 / 46.3		
Supervised	ATSS [58] (Swin-L)	-	- / - / -	38.3	28.5 / 38.1 / 42.9		
GLIP-L [30]	DyHead [11] (Swin-L)	26.9	17.1 / 23.3 / 36.4	-	- / - / -		
DetCLIP-L [53]	ATSS [58] (Swin-L)	31.2	27.6 / 29.6 / 34.5	-	- / - / -		
DetCLIPv2-L (ours)	ATSS [58] (Swin-L)	36.6	33.3 / 36.2 / 38.5	53.1	49.0 / 53.2 / 54.9		

Table 14. **Performance on LVIS [25] val split.** Fixed AP [12] is reported. DetCLIPv2 achieves SoTA performance.

Model	PascalVOC	AerialDrone	Aquarium	Rabbits	EgoHands	Mushrooms	Packages
GLIP-T	62.3	31.2	52.5	70.8	78.7	88.1	75.6
GLIPv2-T	66.4	30.2	52.5	74.8	80.0	88.1	74.3
DetCLIPv2-T (ours)	67.5	41.8	50.8	80.4	79.8	90.1	73.7
GLIP-L	69.6	32.6	56.6	76.4	79.4	88.1	67.1
GLIPv2-B	71.1	32.6	57.5	73.6	80.0	88.1	74.9
GLIPv2-H	74.4	36.3	58.7	77.1	79.3	88.1	74.3
DetCLIPv2-L (ours)	74.4	44.1	54.7	80.9	79.9	90	74.1

Model	Raccoon	Shellfish	Vehicles	Pistols	Pothole	Thermal	Avg
GLIP-T	61.4	51.4	65.3	71.2	58.7	76.7	64.9
GLIPv2-T	63.7	54.4	63.0	73.0	60.1	83.5	66.5
DetCLIPv2-T (ours)	70.8	54.8	66.5	77.7	54.8	82.2	68.5
GLIP-L	69.4	65.8	71.6	75.7	60.3	83.1	68.9
GLIPv2-B	68.2	70.6	71.2	76.5	58.7	79.6	69.4
GLIPv2-H	73.1	70.0	72.2	72.5	58.3	81.4	70.4
DetCLIPv2-L (ours)	69.4	61.2	68.1	80.3	57.1	81.1	70.4

Table 15. **Detailed fine-tuning AP (%) performance on ODinW13.**

CLIPv2 achieves significant improvements over the fully-supervised method, i.e., about 15 overall AP improvement can be observed for both T/L models.

Detailed fine-tuning results for ODinW13. Table 15 reports the detailed fine-tuning performance for 13 datasets contained in ODinW13, and we make a comparison between DetCLIPv2 and GLIP [30]/GLIPv2 [57]. DetCLIPv2-L/T surpass their GLIP [30]/GLIPv2 [57] counterparts on average AP over 13 datasets.

D. More Visualization Results

Figure 1-a and 1-b provide more visualization examples of word-region alignment learnt by DetCLIPv2, using the images from CC12M. As mentioned in the main paper, we find the optimal-match region in the image for each textual concept in the caption. As can be seen, DetCLIPv2 learns to recognize and locate various concepts with broad domain coverage, including comic objects (i.e., Monkey

King, Santa Claus, etc.), abstract concepts (i.e., ‘man’s best friend’ means a dog) and many concepts that are not covered by the detection/grounding data (i.e., tensioner, tattoos, lifebuoy and etc.), which demonstrates the effectiveness of learning from massive image-text pairs.

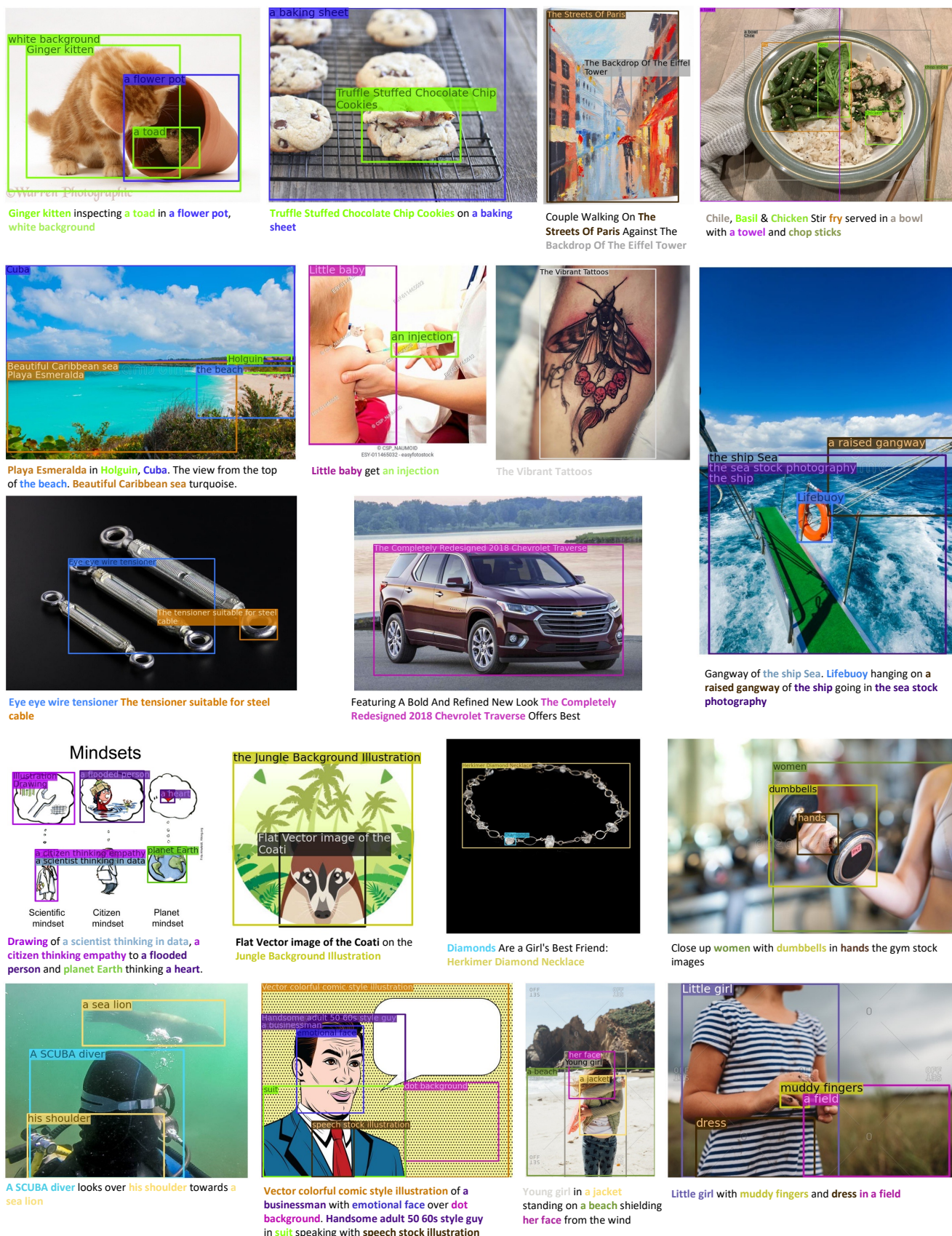
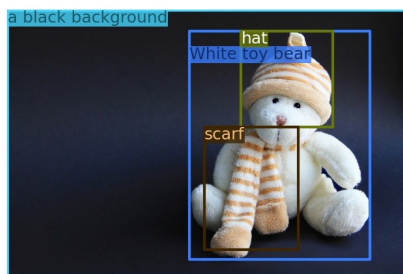
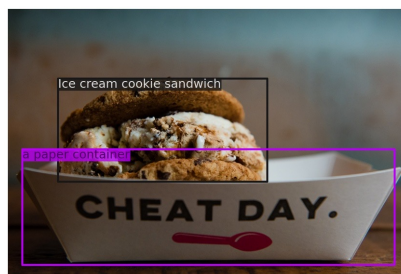


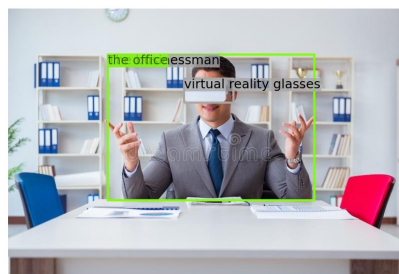
Figure 1-a. More visualizations for word-region alignment. DetCLIPv2 learns word-region alignment with broad domain coverage.



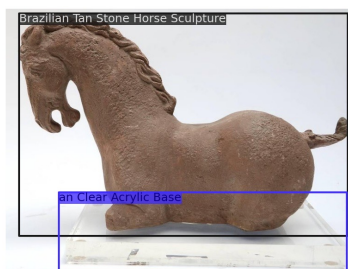
White toy bear in hat and scarf on a black background



Ice cream cookie sandwich in a paper container that reads "Cheat Day"



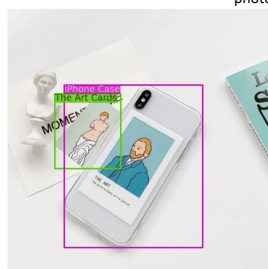
The businessman with virtual reality glasses in the office. Businessman with virtual reality glasses in the office stock photos



Brazilian Tan Stone Horse Sculpture on an Clear Acrylic Base, 1980s For Sale



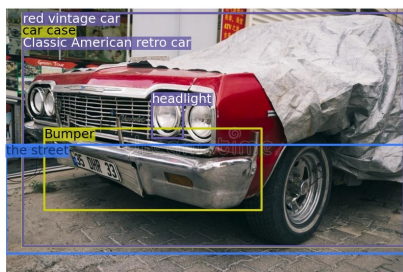
The Mask Loki Pendant Tippy Taste Jewelry



The Art Cards iPhone Case



"I'm ready to party like it's going out of style!" Monkey Business Fantasy Inspiration, Character Inspiration, Character Art, Character Design, Writing Inspiration, Monkey Art, Monkey King, Cyberpunk Character, Cyberpunk Art



Classic American retro car under car case on the street. Bumper and headlight of red vintage car. Turkey, Cappadocia stock images



In the Cayman Islands, there are many opportunities to have fun with man's best friend.



Golden statue of the god from Hinduism seated stock images



Silhouette of a deer head with big horns royalty free illustration



4 pc dream catcher hollow silver tassel indian feather all match women necklace stud earring bracelet & ring bohemian style fashion fine



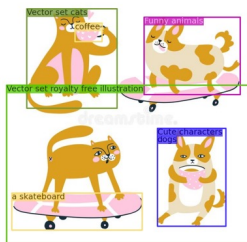
Surrealistic illustration of a hatching shaman. Trying to please a giant park bird, detailed intricate colorful drawing, outlined royalty free illustration



A rainbow appears at Manning Park prior to the event.



Christmas illustration with Santa Claus. New year vector illustration. Hand drawn. Funny Santa Claus on skis and a skateboard royalty free illustration



Funny animals drink coffee and ride a skateboard. Vector set cats and dogs. Cute characters. Vector set royalty free illustration

Figure 1-b. More visualizations for word-region alignment (cont.). DetCLIPv2 learns word-region alignment with broad domain coverage.

References

- [1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, pages 6077–6086, 2018.
- [2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.
- [3] Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *CVPR*, pages 2846–2854, 2016.
- [4] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *NeurIPS*, 2020.
- [5] Zhaowei Cai, Gukyeon Kwon, Avinash Ravichandran, Erhan Bas, Zhuowen Tu, Rahul Bhotika, and Stefano Soatto. X-detr: A versatile architecture for instance-wise vision-language tasks. *arXiv preprint arXiv:2204.05626*, 2022.
- [6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229. Springer, 2020.
- [7] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021.
- [8] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.
- [9] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. preprint arXiv:1604.06174, 2016.
- [10] Ze Chen, Zhihang Fu, Rongxin Jiang, Yaowu Chen, and Xian-Sheng Hua. Slv: Spatial likelihood voting for weakly supervised object detection. In *CVPR*, pages 12995–13004, 2020.
- [11] Xiyang Dai, Yinpeng Chen, Bin Xiao, Dongdong Chen, Mengchen Liu, Lu Yuan, and Lei Zhang. Dynamic head: Unifying object detection heads with attentions. In *CVPR*, pages 7373–7382, 2021.
- [12] Achal Dave, Piotr Dollár, Deva Ramanan, Alexander Kirillov, and Ross Girshick. Evaluating large-vocabulary object detectors: The devil is in the details. *arXiv preprint arXiv:2102.01066*, 2021.
- [13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009.
- [14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *ACL*, 2019.
- [15] Zi-Yi Dou, Aishwarya Kamath, Zhe Gan, Pengchuan Zhang, Jianfeng Wang, Linjie Li, Zicheng Liu, Ce Liu, Yann LeCun, Nanyun Peng, et al. Coarse-to-fine vision-language pre-training with fusion in the backbone. *arXiv preprint arXiv:2206.07643*, 2022.
- [16] Dario Fontanel, Matteo Tarantino, Fabio Cermelli, and Barbara Caputo. Detecting the unknown in object detection. *arXiv preprint arXiv:2208.11641*, 2022.
- [17] Mingfei Gao, Chen Xing, Juan Carlos Nieves, Junnan Li, Ran Xu, Wenhao Liu, and Caiming Xiong. Towards open vocabulary object detection without human-provided bounding boxes. *arXiv preprint arXiv:2111.09452*, 2021.
- [18] Peng Gao, Zhengkai Jiang, Haoxuan You, Pan Lu, Steven CH Hoi, Xiaogang Wang, and Hongsheng Li. Dynamic fusion with intra-and inter-modality attention flow for visual question answering. In *CVPR*, pages 6639–6648, 2019.
- [19] 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*, 2, 2021.
- [20] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *CVPR*, pages 5356–5364, 2019.
- [21] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python. 2020.
- [22] Matthew Inkawhich, Nathan Inkawhich, Hai Li, and Yiran Chen. Self-trained proposal networks for the open world. *arXiv preprint arXiv:2208.11050*, 2022.
- [23] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021.
- [24] Borui Jiang, Ruixuan Luo, Jiayuan Mao, Tete Xiao, and Yunying Jiang. Acquisition of localization confidence for accurate object detection. In *ECCV*, 2018.
- [25] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdetr-modulated detection for end-to-end multi-modal understanding. In *ICCV*, pages 1780–1790, 2021.
- [26] Dahun Kim, Tsung-Yi Lin, Anelia Angelova, In So Kweon, and Weicheng Kuo. Learning open-world object proposals without learning to classify. *IEEE Robotics and Automation Letters*, 7(2):5453–5460, 2022.
- [27] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *ICML*, 2021.
- [28] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv preprint arXiv:2201.12086*, 2022.
- [29] Kaican Li, Kai Chen, Haoyu Wang, Lanqing Hong, Chaoqiang Ye, Jianhua Han, Yukuai Chen, Wei Zhang, Chunjing

- Xu, Dit-Yan Yeung, et al. Coda: A real-world road corner case dataset for object detection in autonomous driving. *arXiv e-prints*, pages arXiv-2203, 2022.
- [30] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. *arXiv preprint arXiv:2112.03857*, 2021.
- [31] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*, pages 121–137. Springer, 2020.
- [32] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.
- [33] Wei Liu, Sihan Chen, Longteng Guo, Xinxin Zhu, and Jing Liu. Cptr: Full transformer network for image captioning. *arXiv preprint arXiv:2101.10804*, 2021.
- [34] 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 *ICCV*, pages 10012–10022, 2021.
- [35] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.
- [36] Jinjie Mai, Meng Yang, and Wenfeng Luo. Erasing integrated learning: A simple yet effective approach for weakly supervised object localization. In *CVPR*, pages 8766–8775, 2020.
- [37] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. Mixed precision training. In *ICLR*, 2018.
- [38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021.
- [39] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- [40] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016.
- [41] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.
- [42] Hamid Rezaatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *CVPR*, pages 658–666, 2019.
- [43] 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 *ICCV*, pages 8430–8439, 2019.
- [44] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, pages 2556–2565, 2018.
- [45] 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.
- [46] Peng Tang, Chetan Ramaiah, Yan Wang, Ran Xu, and Caiming Xiong. Proposal learning for semi-supervised object detection. In *WACV*, pages 2291–2301, 2021.
- [47] Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. In *NeurIPS*, volume 33, pages 18583–18599, 2020.
- [48] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, 2019.
- [49] Johnathan Xie and Shuai Zheng. Zsd-yolo: Zero-shot yolo detection using vision-language knowledge distillation. *arXiv preprint arXiv:2109.12066*, 2021.
- [50] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, pages 2048–2057. PMLR, 2015.
- [51] Mengde Xu, Zheng Zhang, Han Hu, Jianfeng Wang, Lijuan Wang, Fangyun Wei, Xiang Bai, and Zicheng Liu. End-to-end semi-supervised object detection with soft teacher. In *ICCV*, pages 3060–3069, 2021.
- [52] Jianwei Yang, Jiasen Lu, Dhruv Batra, and Devi Parikh. A faster pytorch implementation of faster r-cnn. <https://github.com/jwyang/faster-rcnn.pytorch>, 2017.
- [53] 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.
- [54] Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. Filip: Fine-grained interactive language-image pre-training. In *ICLR*, 2022.
- [55] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Exploring visual relationship for image captioning. In *ECCV*, pages 684–699, 2018.
- [56] Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy. Open-vocabulary detr with conditional matching. *arXiv preprint arXiv:2203.11876*, 2022.
- [57] Haotian Zhang, Pengchuan Zhang, Xiaowei Hu, Yen-Chun Chen, Liunian Harold Li, Xiyang Dai, Lijuan Wang, Lu Yuan, Jenq-Neng Hwang, and Jianfeng Gao. Glipv2: Unifying localization and vision-language understanding. *arXiv preprint arXiv:2206.05836*, 2022.
- [58] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *CVPR*, pages 9759–9768, 2020.

- [59] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, and Jianfeng Gao. Regionclip: Region-based language-image pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16793–16803, June 2022.
- [60] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Phillip Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. *arXiv preprint arXiv:2201.02605*, 2022.
- [61] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *CVPR*, pages 9308–9316, 2019.
- [62] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.
- [63] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le. Rethinking pre-training and self-training. In *NeurIPS*, volume 33, pages 3833–3845, 2020.