Generating Enhanced Negatives for Training Language-Based Object Detectors

Shiyu Zhao[1,*] Long Zhao[3] Vijay Kumar B G[2] Yumin Suh[2]
Dimitris N. Metaxas[1] Manmohan Chandraker[2,4] Samuel Schulter[2]

[1] Rutgers University [2] NEC Laboratories America [3] Google Research [4] UC San Diego

Abstract

The recent progress in language-based open-vocabulary object detection can be largely attributed to finding better ways of leveraging large-scale data with free-form text annotations. Training such models with a discriminative objective function has proven successful, but requires good positive and negative samples. However, the free-form nature and the open vocabulary of object descriptions make the space of negatives extremely large. Prior works randomly sample negatives or use rule-based techniques to build them. In contrast, we propose to leverage the vast knowledge built into modern generative models to automatically build negatives that are more relevant to the original data. Specifically, we use large-language-models to generate negative text descriptions, and text-to-image diffusion models to also generate corresponding negative images. Our experimental analysis confirms the relevance of the generated negative data, and its use in language-based detectors improves performance on two complex benchmarks. Code is available at `https://github.com/xiaofeng94/Gen-Enhanced-Negs`.

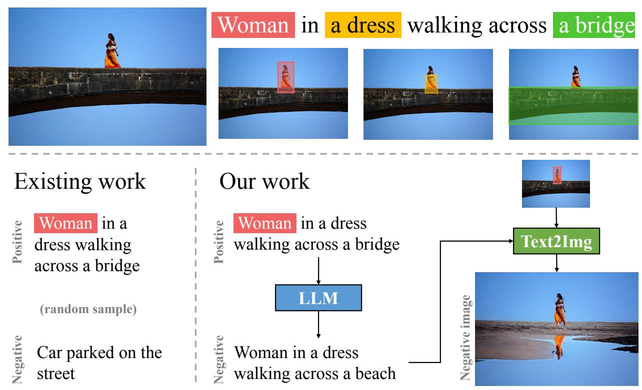


Figure 1. The key contribution of our work is to leverage large-language-models and text-to-image diffusion models to automatically generate negative object descriptions and images for training language-based object detectors. In contrast to prior work, our generated negatives are more relevant to the original data and provide a better training signal for detectors.

1. Introduction

Using natural language in object detection to describe semantics bears the potential to significantly increase the size of the detector's label space and enable novel applications. While standard detectors operate on a fixed label space [23, 38, 42], natural language allows for a broad spectrum of object descriptions, ranging from generic terms like "vehicle" to specific expressions like "the red sports car parked on the left side" [12, 17, 25, 30, 41, 53, 54]. Several works advanced language-based object detection over the past few years with novel training strategies [3, 5, 19, 22, 32, 34, 57] and model architectures [11, 15, 33, 45].

Referring expression or visual grounding datasets [14, 30, 36, 50, 54] provide the natural language object descriptions along with bounding box annotations needed for training. However, this data only describes what is present in the images, *but misses to describe what is not*. Yet, the notion of negatives is crucial for training discriminative models like language-based detectors [7, 24, 39].

Detection datasets with a fixed label space provide negative classes implicitly or explicitly, with exhaustive [23, 42] or federated [6, 16] annotations, respectively. Any part of an image that does not overlap significantly with a bounding box of category c is verified to *not be of that category* (for exhaustive annotation). On the other hand, the space of negatives for a free-form text description of an object is extremely large. While some existing datasets provide negative samples in free-form text [43, 46], they were not annotated with bounding boxes. Hence, existing language-based detectors often define the negatives for one object as the descriptions of all other objects in the same image or descriptions of other random samples [3, 11, 19]. However, such negatives may not be directly related to the original positive description and define a weaker training signal (see Fig. 1). By explicitly evaluating on human-curated negatives, a re-

* Part of this work was done during an internship at NEC Laboratories America. Correspondence to: Shiyu Zhao sz553@rutgers.edu

cent benchmark [41] identified a bias of existing language-based detectors to perform clearly better on positive rather than negative descriptions. However, creating a dataset with high-quality human-curated negatives for large-scale training is labor-intense and costly.

In this work, we propose to explicitly and automatically *generate* negative data in the form of free-form texts as well as images. Prior works [4, 7, 29, 43, 46, 55] rely on rule-based approaches with knowledge graphs and focus only on the language domain or the classification task. In contrast, we leverage generative large-language-models (LLMs) [35, 47] and text-to-image diffusion models [21, 40] to automatically create relevant but contradicting object descriptions along with the corresponding images for language-based object detection, see Fig. 1.

Given an object description of a dataset, we first use LLMs to generate a semantically contradicting description as the negative. Besides changing individual words (foils) based on explicit knowledge graphs or LLMs, like in prior work [4, 7, 20], we demonstrate improved detection performance with two alternative approaches. *(Re-combination):* An LLM first identifies all objects in a sentence, and then creates a contradicting one by re-arranging, ignoring or adding objects. *(In-context summaries):* We prompt an LLM to summarize the differences of a few (less than 100) positive-negative pairs collected from an existing image-level dataset [46]. This summary is then used as context to generate more such examples. Note that we do not need visual input for this step, allowing us to leverage powerful LLMs for semantic and textual reasoning. Moreover, while prior work only focused on the text [4, 7, 29, 43, 55], we also leverage text-to-image diffusion models like GLIGEN [21] to create images that match the generated negative descriptions of objects, which serves as additional training signal. While the direct output of such image-generation models is often noisy and even wrong (not matching the input description), we propose two filtering steps to reduce noise considerably (from 53% to 16% according to an empirical study). Having both negative object descriptions and the corresponding image, allows us to improve the discriminative loss for training language-based object detectors.

Our experiments demonstrate clear accuracy gains on two challenging benchmarks, +2.9AP on OmniLabel [41] and +3.3AP on D^3 [53], when adding our automatically-generated negative data into the training of baseline models like GLIP or FIBER. Moreover, we provide an in-depth analysis of the generated data (text and images) and how they contribute to better language-based detection.

Summary of contributions: (1) Automatic generation of semantically relevant but contradicting negative text and images with large-scale generative models. (2) Recipes to integrate such negative data into language-based detection models like FIBER [3] and GLIP [19] (3) Clear improvements on language-based detection benchmarks [41, 53] including a thorough analysis of the generated data.

2. Related work

Vision & language localization tasks: Open-vocabulary detection (OVD) requires a model to localize object category names without having seen explicit bounding box annotation for them [5, 7, 15, 52, 56, 59]. In contrast, we focus on the more general language-based object detection task [41, 53], which goes beyond simple category names. Referring expression comprehension (REC) aims at localizing the subject of a free-form text expression. However, REC benchmarks [30, 50, 54] fall short in evaluating all aspects of the more general language-based detection task [41, 53]. In visual grounding (VG) [36], the task is to localize noun phrases of a caption in the image. Although being a task on its own, VG datasets have recently been used mostly as training data for OVD. Our work focuses on general language-based object detection, which subsumes and generalizes standard detection, OVD and REC [12, 25, 41, 53].

Language-based object detectors: Two critical abilities of language-based detection are accurate localization and tight text-image fusion. Works like [1, 9, 28] use language-models like BERT [2, 27] to align regions extracted from (pre-trained) detectors with captions. The outstanding zero-shot classification accuracy of large-scale pre-trained models like CLIP [37] or [10, 13, 18] then sparked interest in extensions for localization, with different approaches like distillation [5], fine-tuning [15, 33], pseudo-labeling [34, 57, 58], or combinations thereof [3, 19]. We use such models as test bed, but explore the underlying training data with respect to negative samples.

Negative samples for object detection: The notion of negatives is crucial for training discriminative models [24, 39]. Also for object detection, hard negative mining [44] has proven beneficial for model training. However, these prior works aim to find hard negative training examples rather than negatives in the label space, because the label space is fixed in standard detection. For language-based datasets, the space of potential negatives is extremely large because object descriptions are free-form text. Prior works [4, 29, 43, 46, 55] investigate negative texts for general vision & language models with different strategies, including changing individual words (foil) with rules based on knowledge graphs [31] or with LLMs. SugarCrepe [8] shares a similar idea as us to get negative texts with in-context learning but for image-text level pretraining. For language-based detection, [7] explores such rule-based foils, while [20] uses LLMs with specific templates to replace object names with alternative descriptions. In contrast, our work (1) focuses on the localization task, (2) ex-

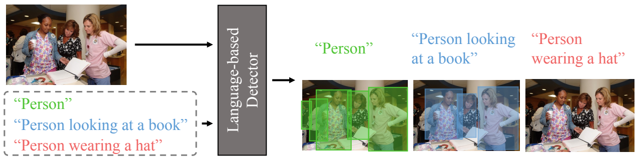


Figure 2. In language-based object detection, a detector receives as input an image and a (variable-length) list of free-form text descriptions of objects. For each description, the model predicts bounding boxes for objects that match the description.

plores more comprehensive strategies to generate negatives with LLMs, and (3) proposes to also generate corresponding negative images with text-to-image diffusion models.

3. Method

3.1. Language-based object detection

Task definition: Given an image and a list of object descriptions, the task is to output bounding boxes along with confidence scores for each description, as shown in Fig. 2. Note the multi-label setting where one object instance can be referred to by multiple descriptions, like "person" and "person looking at book". Also note that an object description might refer to zero objects in the image and the desired output is an empty set of boxes.

Training data: Many language-based detection models [3, 19] use a combination of object detection [23, 42] and visual grounding datasets [14, 36] for training their models. Both types of datasets provide images I and bounding boxes b_l to localize individual objects. Object detection data assigns each bounding box b_l a unique category c out of fixed label space $\mathcal{C}$. The exhaustive labeling of the fixed label space in detection datasets implies the space of negatives. An object of category c is not any of the categories $\mathcal{C} \setminus c$. On the other hand, grounding data provides an image caption t in free-form text, where subsets of words m_l (defined as indices of starting and ending characters in t) are linked to bounding box b_l. For grounding data, the space of negatives is extremely large because one can find as many textual descriptions that do not match t as desired due to the compositionality of free-form text. Many language-based detectors [3, 19] only use the words in t that are not referred to by m_l as negatives for the bounding box b_l. We argue that this choice is sub-optimal because these words may refer to entirely different objects and are easy to discriminate. In the following section, we explain how we can automatically generate negative samples that semantically related to the original text t and hence provide a better training signal.

3.2. Generating negative samples

Our goal is to automatically and explicitly generate negative samples based on the original text descriptions t to improve the training signal for language-based detectors. A key observation of our work is to leverage the vast knowledge encoded into large-language models (LLMs) [35, 47] and text-to-image diffusion models (T2I) [21, 40]. Besides proposing novel ways to instruct LLMs for generating negative text descriptions (Sec. 3.2.1), we also propose to generate negative images (Sec. 3.2.2).

3.2.1 Generating negative descriptions with LLMs

Given an object description t that matches the visual content inside a bounding box b_l, we define a "negative" description t' as any text that is *semantically different* to the original text. Furthermore, our intuition is that good negative descriptions are still semantically related to the original description, but not the same. An example is: "Person in red shirt" as the original description and "Person in blue shirt" as a contradicting negative one.

Prior work [4, 7, 43] explored rule-based approaches to generate negative text. However, such rules are typically limited to simple knowledge graphs and are limited to replacing only individual words, often just nouns, or swapping words. In contrast, we explore more powerful LLMs to automatically generate relevant negatives. To make the negative text generation efficient and economic, in all cases, we first leverage a strong instruction-tuned LLM [35] to generate 50k positive-negative pairs, and then finetune a LLaMA-7B [47] model with those pairs to then generate negative captions on large grounding datasets. In the following, we describe three ways to instruct an LLM for generating positive-negative pairs of object descriptions:

LLM-based foils: We first prompt an instruction-tuned LLM [35] to find concepts (i.e., objects, attributes and relationships) in object descriptions. Compared to rule-based parsers [51], LLMs can provide richer information. For example, for the caption "A transportation vehicle is carrying a crowd of people who are sitting and standing.", the parser ignores "sitting" and "standing", while LLMs regard them as attributes. Then, we pick one concept from the first step sequentially and prompt LLMs again to generate a negative caption by changing the concept. For both steps, the prompts are manually curated with the task definition and step-by-step instructions for the generation. Please find the exact prompt for the LLM in the supplement.

Re-combination: Next, we give the LLM more freedom in generating negative descriptions. We first prompt the LLM to identify all objects in the original caption, and then to re-combine them to create a new sentence different from the original one. We allow the LLM to ignore, change or add new objects. For example, given the caption "A boy is playing with his dog" and two objects "boy" and "dog", the LLM can output "The girl and her dog are playing fetch in the park". Detailed prompts for both identifying objects and re-combination are in the supplement.

In-context summary: Third, we enable LLMs to learn how to generate negative descriptions by providing human-annotated positive-negative pairs as in-context samples. We randomly sample 80 pairs of positive and negative texts from the Winoground dataset [46] and prompt the instruction-tuned LLM [35] to summarize the difference of those pairs in plain text. Then, instead of manually creating prompts to generate positive-negative pairs, we leverage the summary together with three randomly sampled Winoground pairs as prompts to the LLM, and generate several positive-negative pairs to finetune LLaMA. After finetuning, the LLaMA model is used to generate negative texts for given descriptions. This pipeline does not require handcrafted prompts to LLMs as the explanation of the concept of negatives and how to create them. The supplement contains full prompts for generating a summary, and generating positive-negative pairs for finetuning.

3.2.2 Generating negative images with T2I models

Given an original image I, a bounding box b and a corresponding object description t, we define a negative image I' as any image that has a different semantic content inside b. The rest of the image can be equivalent to I. To obtain such imagery, we start with visual grounding data that provide bounding boxes, positive captions with text phrases, and alignment between them. We propose a two-step process: First, we turn the positive caption into a negative one. Second, we use conditioned image generation tools to alter the visual content inside the bounding box b.

Negative text for negative images: Although we have already described an approach to generate negative descriptions in Sec. 3.2.1, doing so to generate a negative image requires a different approach. In this case, the generated negative text needs to preserve the alignment m_l to the ground truth bounding box b_l in order to instruct the generative image model GLIGEN [21]. Hence, we first select a bounding box b_l and mask out the corresponding words (known via m_l) in the text t. For example, "A boy is playing with his dog" turns into "A boy is playing with [Mask]" if the selected bounding box refers to "his dog". Again, we leverage LLMs [35] to fill in text for "[Mask]" to generate a negative text without reusing the original text. Please refer to Fig. 3 for illustrations.

We finetune a LLaMA-7B for the mask filling task with triplets of positive texts, masked texts, and negative texts. To reduce manual efforts, we follow the approach of in-context summary to get triplet samples. We apply this process twice: We start with only 5 manually created triplets to build a summary and generate 100 samples from the

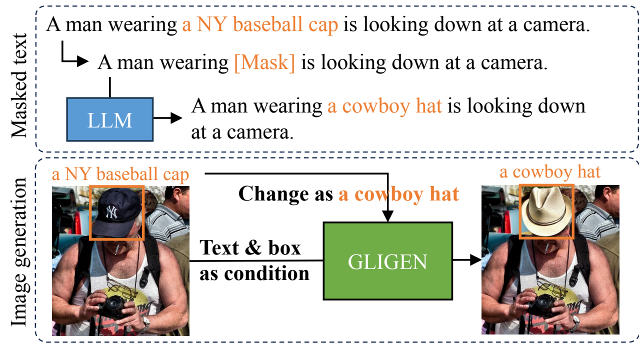


Figure 3. Overview of using LLMs [35, 47] and text-to-image diffusion models [21, 40] to generate negative images.

LLM [35] with human checks. We then repeat the process to generate 50k examples without human checks from a summary of the 100 generated examples. This increases diversity in the generated data.

Conditional image generation: Given an image I, a bounding box b and the altered text t', we generate a negative image I' that is equal to I except inside b, where the visual content is altered to match the text t'. To do so, we use the inpainting and conditioning abilities of GLIGEN [21], a T2I model [40]. Refer to [21] and our supplemental material for more details, and to Fig. 3 for an illustration of the process.

Mitigating noise in image generation: We found that the generated images are often noisy for any of the following reasons: (1) The altered text refers to a big bounding box that covers other smaller boxes. Large portions of the image are then generated and often do not match the concepts those smaller boxes originally covered. (2) The generated negative text does not match the bounding box that is either too small, too large or in a inappropriate position. (3) The T2I model fails to understand the negative text and generates wrong content. We propose two steps to filter such noisy images. First, we simply ignore ground truth boxes b_l for image generation if the box covers more than 75% of any other boxes in the image. Second, we adopt CLIP [37] to verify the semantic similarity of the generated image regions and the corresponding text. Specifically, we compute the similarity with CLIP between the generated image region (visual input) and the original and generated negative texts (text input). We filter out generated images that have a similarity score to the generated negative text lower than a user-defined threshold. Details on the filtering steps are given in the supplemental.

3.3. Learning from negative samples

Detector design and training objective: The generated data does not prescribe any specific architecture for the detector. A common choice, which we also use for our exper-

Figure 4. Illustration of the grounding loss used for training. Predictions that are matched with ground truth receive a positive signal from the associated text (tall rectangles). All other words receive a negative signal (short rectangles). The top left quarter shows the original loss used in [3, 19]. The other three quarters are related to our proposed *generated* negative data and provide additional training signals.

iments, is [3, 19]. The inputs are image I and text t, and the output is a set of bounding boxes $\hat{b}_i$ with corresponding logits $\hat{p}_i \in \mathbb{R}^T$. Here, T is the number of tokens required to represent text t. The ground truth can be represented by a binary assignment matrix $\mathbf{A} \in \mathbb{B}^{L \times T}$. Rows refer to ground truth boxes l and columns to tokens in t. Each element indicates if a token corresponds to a box l, which is given by the ground truth indices m_l. To define a loss, bipartite graph matching associates predictions with ground truth. For matched predictions, the target vector $g_i \in \mathbb{B}^T$ is the corresponding row from $\mathbf{A}$, while it is an all-zero vector for unmatched targets. The loss is then computed as $\mathcal{L} = \sum_i \ell_{\text{FL}}(\hat{p}_i, g_i)$ where FL refers to a focal loss. Fig. 4 illustrates the loss.

Integrating negative text: When sampling an image I, along with text t, boxes b_l and indices m_l, we also randomly sample $K > 1$ negative descriptions from $\{t'_j\}$ that defines the pool of negatives generated for text t. We randomly shuffle the order of all texts to avoid any biases on the location of the one positive description, and then concatenate them into one text string.

Integrating negative images: We explore two options: (1) Simply add the generated images I' along with their generated (but semantically matching) captions t' as additional visual grounding data. The original caption t, which was the starting point to generate the negative image I', is now used as the negative caption. In this way, both the original image I and the generated one I' have positive and negative descriptions. This option is illustrated in Fig. 4. (2) To better leverage the relation between the original and generated data, the second option is to pack them into a single training sample. We simply concatenate the images I and I', as well as the texts t and t'. The ground truth information m_l is updated accordingly. See supplement for details.

	OmniLabel				OmniLabel-Negative			
	AP	APc	APd	APdP	AP	APc	APd	APdP
Detic [60]	8.0	15.6	5.4	8.0	-	-	-	-
MDETR [11]	-	-	4.7	9.1	-	-	-	-
GLIP-T [19]	19.3	23.6	16.4	25.8	13.9	24.8	9.6	26.1
+ Ours	22.2	27.2	18.8	29.0	16.5	28.6	11.6	30.2
FIBER-B [3]	25.7	30.3	22.3	34.8	18.7	31.2	13.3	36.3
+ Ours	28.1	32.1	25.1	36.5	22.3	33.3	16.7	38.3

Table 1. Evaluation on the OmniLabel [41] benchmark.

4. Experiments

4.1. Experimental design

Training procedure: We choose two recent methods, GLIP-T [19] and FIBER-B [3], to demonstrate the effect of our automatically generated negatives. We use the official code and publicly available checkpoints as a starting point. The Flickr30k dataset [36] serves as our grounding dataset to generate the negative data. We then fine-tune GLIP-T and FIBER-B with both positive and negative data, along with the Objects365 detection dataset [42] for 1 epoch. Note that both Objects365 and Flickr30k are part of the original training set. We do not introduce any extra data except our generated negatives. Most hyper-parameters are equal to the original settings of GLIP and FIBER. Any exceptions are described in the supplement.

Evaluation benchmarks: We choose two recently proposed benchmarks, OmniLabel [41] and D^3 [53], as our test beds. These benchmarks evaluate more aspects of language-based detection than existing referring expressions [30, 50, 54] or open-vocabulary detection [6, 17] benchmarks. Specifically, both benchmarks contain complex object descriptions that go beyond simple category names from open-vocabulary detection benchmarks. Moreover, the descriptions can refer to zero, one or multiple instances in the image, in contrast to standard referring expression benchmarks. These properties enable a more stringent evaluation metric as in object detection, which is based on average precision (AP) in both OmniLabel [41] and D^3 [53]. Both benchmarks provide more fine-grained metrics. OmniLabel evaluates separately for categories, descriptions, and descriptions referring to at least one object, with APc, APd and APd-P, respectively. D^3 differentiates descriptions on absence ("Abs") and presence ("Pres") that indicate whether or not they contain any form of negation (e.g., "without"), as well as on text lengths. Finally, we create a specific split for OmniLabel, "OmniLabel-Negative", to evaluate the model only on images that contain at least one negative description (*i.e.*, not referring to any object).

	D^3 (default)			D^3 (by length of texts)			
	Full	Pres	Abs	S	M	L	XL
OFA-L [49]	4.2	4.1	4.6	4.9	5.4	3.0	2.1
OWL-ViT-L [33]	9.6	10.7	6.4	20.7	9.4	6.0	5.3
G-DINO-B [26]	20.7	20.1	22.5	22.6	22.5	18.9	16.5
OFA-DOD [53]	21.6	23.7	15.4	23.6	22.6	20.5	18.4
GLIP-T [19]	19.1	18.3	21.5	22.4	22.0	16.6	10.6
+ Ours	21.4	20.6	23.7	28.1	24.5	17.4	11.5
FIBER-B [3]	22.7	21.5	26.0	30.1	25.9	17.9	13.1
+ Ours	26.0	25.2	28.1	35.5	29.7	20.5	14.2

Table 2. Evaluation on the D^3 [53] benchmarks.

4.2. Benchmark comparisons

Tabs. 1 and 2 evaluate the impact of our generated negative training data on the OmniLabel [41] and D^3 [53] benchmarks. In both tables, the first set of rows are baselines provided by the benchmarks. The following rows show the main comparisons for GLIP-T [19] and FIBER-B [3] with and without adding our generated negative training data. First, we can see that adding negative data improves results across all metrics for both models and both benchmarks. On OmniLabel, we can see a +2.9% and +2.4% increase in AP for GLIP-T and FIBER-B, respectively. Similarly, we observe a +2.3% and +3.3% increase in the main metric of D^3 (AP on full descriptions) for GLIP-T and FIBER-B.

4.3. Analysis on negative texts

Effectiveness of different negative texts: We finetune FIBER-B without and with different kinds of negative texts mentioned in Sect. 3.2.1, i.e., Rule-based foils, LLM-based foils, Re-combination with LLMs, In-context summary with LLMs, and present results in Table 3. We find all kinds of negatives improve the original FIBER-B on both OmniLabel and D^3 benchmarks. Negative texts from LLMs generally achieve better results compared to LLM-based foils, which indicates that LLMs are powerful tools for negative text generation. Moreover, both recombination and in-context summary with LLMs outperform LLM-based foils in all metrics except APd-P. Note that APd-P refers to evaluations without negative label spaces, which is a task weaker than language-based detection. Based on such results, we argue that although word foils provide promising results in traditional studies [7, 43], it is sub-optimal to LLMs. We need to explore varied ways to unlock the ability of LLMs. We believe that our two solutions, i.e., Re-combination and In-context summary, provide a good starting point for future studies. Besides using only one kind of negative texts, we also explore the combinations of different kinds of negative texts in the supplement.

Diversity of rule-based and LLM-based negatives: In this part, we investigate the diversity of different negative texts. First, we count number of words for each negative text and provide the distribution for negatives of different sources in Fig. 5. As shown, all four distributions have a peak around 10 words, but the one of rule-based foils is higher than others. That means rule-based foils provide more negative texts with similar lengths.

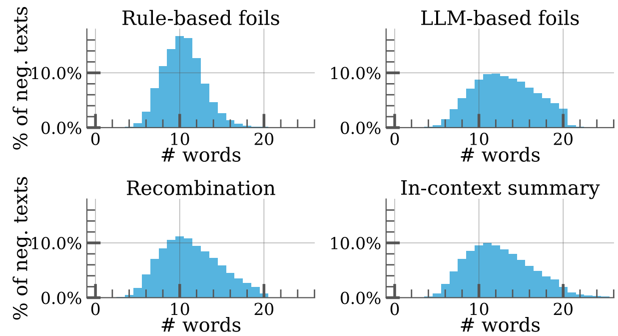


Figure 5. Percentage of negative texts with the numbers of words. Four negative generation methods are compared.

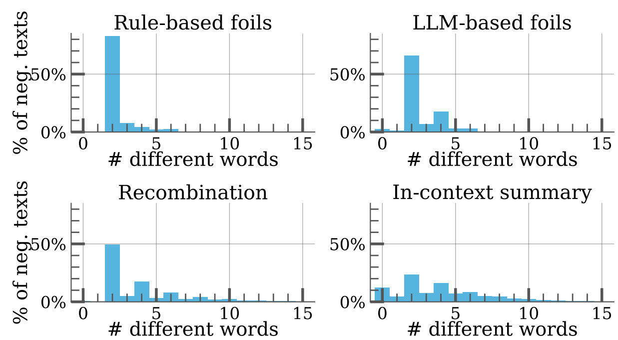


Figure 6. Percentage of negative texts with the numbers of words that are different from the original caption. Four negative generation methods are compared.

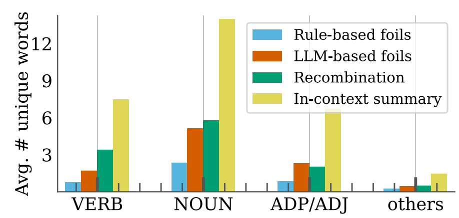


Figure 7. Average numbers of extra unique words per thousand generated negative texts, which are not included in the original dataset. We group words by their part-of-speech.

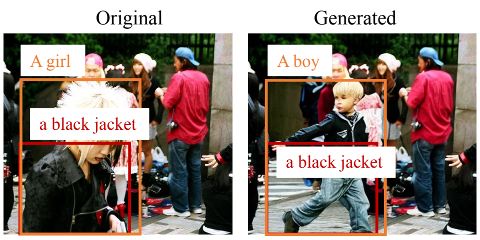
Second, we count the number of different words between the original positive caption and the negative caption, and present the distributions in Fig. 6. We find that LLM-based methods usually changes more words than rule-based foils, which increases the diversity. Moreover, in-context sum-

	Whole OmniLabel				OmniLabel-Negative				D^3		
	AP	APc	APd	APd-P	AP	APc	APd	APd-P	Full	Pres	Abs
Original FIBER-B	25.7	30.3	22.3	34.8	18.7	31.2	13.3	36.3	22.7	21.5	26.0
+ Rule-based foils	26.4	**31.7**	22.6	34.9	19.2	32.6	13.6	36.4	24.1	23.2	26.9
+ LLM-based foils	26.5	30.7	23.3	**35.9**	20.8	32.1	15.4	**38.0**	24.6	24.0	26.5
+ Re-combination	**26.9**	30.8	**23.9**	**35.9**	**21.1**	**32.3**	**15.6**	37.6	25.3	24.6	27.3
+ In-context summary	26.6	30.8	23.4	34.2	**21.1**	32.2	15.7	36.4	**25.7**	**25.2**	**27.5**

Table 3. Performance of FIBER-B trained with negative texts from four negative generation methods.

	OmniLabel			D^3
	APc	APd	APd-P	
Original FIBER	30.3	22.3	34.8	22.7
FIBER w/ neg. texts	30.7	23.9	35.5	25.9
+ W/ neg. img. directly	30.1	22.4	33.7	23.0
+ Box filter	31.0	23.8	35.4	23.6
+ Box&CLIP filters	31.1	24.2	35.9	24.1
+ Above + concat. img.	31.7	24.8	35.9	24.8
+ Above + weight ensemble	**32.1**	**25.2**	**36.5**	**26.0**

Table 4. FIBER trained with negative images.



Original Caption: A girl with blond spiky hair and a black jacket walking along a sidewalk

Edited Caption w/ LLM: A boy with blond spiky hair and a black jacket walking along a sidewalk .

Figure 8. Noisy generated images. The orange box contains the red box, and editing the orange changes the red unexpectedly.

mary has a more flat distribution compared to others. Probably, in-context summary learns how to generate negatives automatically from data and has less restrictions. Besides, in-context summary has more cases with no word changed where negative texts are generated by just shuffling words or concepts in the original text. Such shuffling is a common pattern of Winoground [46], and our in-context summary can learn such data specific patterns.

Third, we count how many extra words that does not exist in the original Flickr30k dataset are introduced in different negative generation methods. Fig. 7 shows the average number of extra words per 1000 negative texts. We group words into four part-of-speech categories, i.e., VERB, NOUN, ADP/ADJ, and others. As shown, LLMs introduce more extra words on average than rule-based foils probably because rule-based foils are limited in a predefined set of words. However, LLMs are open to any concepts and have great potentials of generating diverse texts. In-context summary introduces the most extra words for all categories, which is likely a benefit of learning negative generation from data. The above statistics indicate a clear view that LLMs generate more diverse data than rule-based foils.

4.4. Analysis on negative images

Noise in generated images: As mentioned in the last paragraph of Sect. 3.2.2, the raw generated images are noisy in several ways. First, the editing of a large box will override the context of smaller boxes that are covered by the large box. As shown in Fig. 8, GLIGEN did follow the instruction to generate a boy in the orange box, but the black jacket in the red box is missing. As a remedy, we apply our first de-noise step "Box Filter". That is, we ignore boxes that contain any other boxes when generating negative images Second, GLIGEN may generate contents with wrong attributes or objects, as shown in Fig. 9 (Left). Moreover, our generation pipeline includes some cases where the edited text and the bounding box does not match. As shown in Fig. 9 (Right), the box for "his lap" cannot be modified as "his knees". Thus, GLIGEN generates wrong contents. As described in Sect. 3.2.2, we adopt a pretrained CLIP model to judge if generated contents are correct, which mitigates the noise to some extent. As shown in Fig. 9, both negative images get low CLIP scores and can be filtered out with a threshold. We call such thresholding "CLIP Filter".

Subject studies on Box and CLIP filters: We employ human experts to check the amount of noisy generated images. First, for negative images w/o filter, w/ Box filter, and w/ Box&CLIP filters, we separately and randomly select 100 samples. Then, we ask two experts to check if a negative image is not noisy by comparing it with its caption and the original positive image. We regard an image as not noisy when both experts agree. As shown in Fig. 10, both filters reduce the noise. The Box filter improves from 47% to 63%, and the CLIP filter improves to 84%.

Effectiveness of generated negative images: To show the effectiveness of generated images themselves, we take captions of generated images as additional negative texts to in-context summary, and finetune a FIBER model as baseline. Then, we compare the baseline with variants of adding generated negative images in Table 4. As shown, the performance drops if we directly take raw negative images as new visual grounding data without any filters (i.e., W/ neg. img. directly). Probably, there are too much noise in raw negative images as shown in Fig. 10. When applying both Box and CLIP filters on negative images, we can achieve slight improvement on OmniLabel compared to using neg-

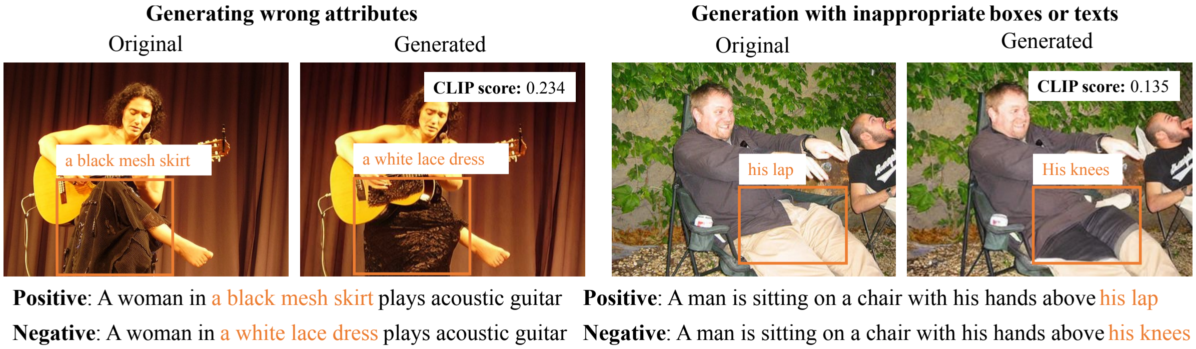


Figure 9. **Left:** Noisy negative images due to wrong attributes or objects generated by text-to-image models. **Right:** Noisy negative images caused by inappropriate bounding boxes or negative texts from LLMs. CLIP scores of generated images refer to the similarity between the box and the negative text compared to the positive text. Thresholding on CLIP scores remove those noisy images.

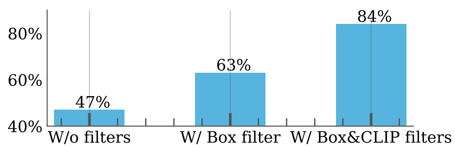


Figure 10. Percentage of good generated negative images.

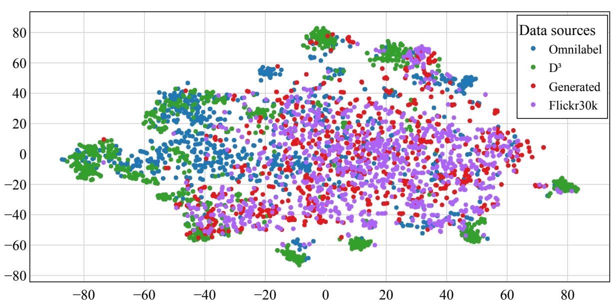


Figure 11. Distributions of image regions from Omnilabel, D^3, and our generated images. Visualization with t-SNE.

ative texts only.

Concatenating images during training: Following the idea of concatenating the positive and negative captions as the text input, we concatenate the positive and negative images as one input image during training. See supplement for an example. In this way, models are forced to tell the difference between the positive and negative images within one training iteration, which helps detectors to learn better about the negative. As shown in Table 4, such a simple technique improves upon "+ Box&CLIP filters" both on OmniLabel and D^3. Furthermore, we ensemble the weights of two FIBER models, one finetuned with negative texts only, and the other finetuned with both negative texts and images. Finally, compared to using negative texts only, we gain 1.3 APd on OmniLabel and no performance drop on D^3.

Looking into generated images and benchmarks: Table 4 shows that negative images help on OmniLabel but not much on D^3. We explore this on a data basis. We first crop image regions for generated images, OmniLabel images, D^3 images, and Flickr30k images based on the bounding boxes. Then, we randomly select 1000 image regions and feed them into a CLIP image encoder to get CLIP embeddings. Later, we input those embeddings to t-SNE [48] to illustrate the similarities between different image regions. As shown in Fig. 11, D^3's regions are grouped into several clusters, while OmniLabel and our generated regions are scattered in the center. This indicates that there is a clear domain gap between D^3 and the others. Thus, it is plausible that our generated images only helps on OmniLabel. In our view, the gap comes from that D^3 collect data in groups based on categories. In contrast, OmniLabel collects data randomly.

5. Conclusion

Language-based detection requires localization of objects by a referring free-form text descriptions. To train accurate models in a discriminative way, the training data must contain good negative samples. Starting with an existing dataset, we propose (1) novel ways to prompt LLMs for generating additional negative texts, and (2) generating negative images to complement the training signal. Based on our experimental evaluations, we conclude that such additional negative training data indeed translates into improved detection accuracy on standard benchmarks. Our analysis demonstrates the importance of diversity in the generated text, which is higher with our approach than with prior works, and the quality of the generated images, which our proposed filtering steps can significantly increase.

Acknowledgments: This research project has been partially funded by research grants to Dimitris N. Metaxas through NSF: 2310966, 2235405, 2212301, 2003874, and FA9550-23-1-0417.

References

[1] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. UNITER: UNiversal Image-TExt Representation Learning. In *ECCV*, 2020. 2

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 2

[3] 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. In *NeurIPS*, 2022. 1, 2, 3, 5, 6

[4] Sivan Doveh, Assaf Arbelle, Sivan Harary, Eli Schwartz, Roei Herzig, Raja Giryes, Rogerio Feris, Rameswar Panda, Shimon Ullman, and Leonid Karlinsky. Teaching Structured Vision & Language Concepts to Vision & Language Models. In *CVPR*, 2023. 2, 3

[5] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary Object Detection via Vision and Language Knowledge Distillation. In *ICLR*, 2022. 1, 2

[6] Agrim Gupta, Piotr Dollár, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019. 1, 5

[7] Ryota Hinami and Shin'ichi Satoh. Discriminative Learning of Open-Vocabulary Object Retrieval and Localization by Negative Phrase Augmentation. In *EMNLP*, 2018. 1, 2, 3, 6

[8] Cheng-Yu Hsieh, Jieyu Zhang, Zixian Ma, Aniruddha Kembhavi, and Ranjay Krishna. Sugarcrepe: Fixing hackable benchmarks for vision-language compositionality. In *NeurIPS*, 2023. 2

[9] Ronghang Hu and Amanpreet Singh. UniT: Multimodal Multitask Learning with a Unified Transformer. In *ICCV*, 2021. 2

[10] 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. 2

[11] 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*, 2021. 1, 5

[12] Aishwarya Kamath, Sara Price, Jonas Pfeiffer, Yann LeCun, and Nicolas Carion. TRICD: Testing Robust Image Understanding Through Contextual Phrase Detection. Technical report, NYU, 2023. 1, 2

[13] Zaid Khan, Vijay Kumar B.G., Xiang Yu, Samuel Schulter, Manmohan Chandraker, and Yun Fu. Single-Stream Multi-Level Alignment for Vision-Language Pretraining. In *ECCV*, 2022. 2

[14] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 123:32–73, 2017. 1, 3

[15] Weicheng Kuo, Yin Cui, Xiuye Gu, AJ Piergiovanni, and Anelia Angelova. F-vlm: Open-vocabulary object detection upon frozen vision and language models. In *ICLR*, 2023. 1, 2

[16] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020. 1

[17] Chunyuan Li, Haotian Liu, Liunian Harold Li, Pengchuan Zhang, Jyoti Aneja, Jianwei Yang, Ping Jin, Houdong Hu, Zicheng Liu, Yong Jae Lee, and Jianfeng Gao. ELEVATER: A Benchmark and Toolkit for Evaluating Language-Augmented Visual Models. In *NeurIPS*, 2022. 1, 5

[18] Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Deepak Gotmare, Shafiq Joty, Caiming Xiong, and Steven Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *NeurIPS*, 2021. 2

[19] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *CVPR*, pages 10965–10975, 2022. 1, 2, 3, 5, 6

[20] Liunian Harold Li, Zi-Yi Dou, Nanyun Peng, and Kai-Wei Chang. DesCo: Learning Object Recognition with Rich Language Descriptions. In *NeurIPS*, 2023. 2

[21] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *CVPR*, 2023. 2, 3, 4

[22] Chuang Lin, Peize Sun, Yi Jiang, Ping Luo, Lizhen Qu, Gholamreza Haffari, Zehuan Yuan, and Jianfei Cai. Learning object-language alignments for open-vocabulary object detection. In *ICLR*, 2023. 1

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *ECCV*, 2014. 1, 3

[24] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal Loss for Dense Object Detection. In *ICCV*, 2017. 1, 2

[25] Chang Liu, Henghui Ding, and Xudong Jiang. GRES: Generalized Referring Expression Segmentation. In *CVPR*, 2023. 1, 2

[26] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 6

[27] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019. 2

[28] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. In *NeurIPS*, 2019. 2

[29] Zixian Ma, Jerry Hong, Mustafa Omer Gul, Mona Gandhi, Irena Gao, and Ranjay Krishna. CREPE: Can Vision-Language Foundation Models Reason Compositionally? In *CVPR*, 2023. 2

[30] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L. Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, 2016. 1, 2, 5

[31] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, 1995. 2

[32] Zhixiang Min, Bingbing Zhuang, Samuel Schulter, Buyu Liu, Enrique Dunn, and Manmohan Chandraker. Neurocs: Neural nocs supervision for monocular 3d object localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21404–21414, 2023. 1

[33] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, Xiao Wang, Xiaohua Zhai, Thomas Kipf, and Neil Houlsby. Simple Open-Vocabulary Object Detection with Vision Transformers. In *ECCV*, 2022. 1, 2, 6

[34] Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling Open-Vocabulary Object Detection. In *NeurIPS*, 2023. 1, 2

[35] OpenAI. Gpt-3.5. 2022. 2, 3, 4

[36] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k Entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*, pages 2641–2649, 2015. 1, 2, 3, 5

[37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 2, 4

[38] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NeurIPS*, 2015. 1

[39] Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive Learning with Hard Negative Samples. In *ICLR*, 2021. 1, 2

[40] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2, 3, 4

[41] Samuel Schulter, Vijay Kumar BG, Yumin Suh, Konstantinos M Dafnis, Zhixing Zhang, Shiyu Zhao, Dimitris Metaxas, et al. OmniLabel: A Challenging Benchmark for Language-Based Object Detection. In *ICCV*, 2023. 1, 2, 5, 6

[42] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Jing Li, Xiangyu Zhang, and Jian Sun. Objects365: A Large-scale, High-quality Dataset for Object Detection. In *ICCV*, 2019. 1, 3, 5

[43] Ravi Shekhar, Sandro Pezzelle, Yauhen Klimovich, Aurelie Herbelot, Moin Nabi, Enver Sangineto, and Raffaella Bernardi. FOIL it! Find One mismatch between Image and Language caption. In *ACL*, 2017. 1, 2, 3, 6

[44] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training Region-based Object Detectors with Online Hard Example Mining. In *CVPR*, 2016. 2

[45] Sanjay Subramanian, William Merrill, Trevor Darrell, Matt Gardner, Sameer Singh, and Anna Rohrbach. ReCLIP: A Strong Zero-Shot Baseline for Referring Expression Comprehension. In *ACL*, 2022. 1

[46] Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing Vision and Language Models for Visio-Linguistic Compositionality. In *CVPR*, 2022. 1, 2, 4, 7

[47] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. 2, 3, 4

[48] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9 (86):2579–2605, 2008. 8

[49] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. OFA: Unifying Architectures, Tasks, and Modalities Through a Simple Sequence-to-Sequence Learning Framework. In *ICML*, 2022. 6

[50] Chenyun Wu, Zhe Lin, Scott Cohen, Trung Bui, and Subhransu Maji. PhraseCut: Language-based Image Segmentation in the Wild. In *CVPR*, 2020. 1, 2, 5

[51] Hao Wu, Jiayuan Mao, Yufeng Zhang, Yuning Jiang, Lei Li, Weiwei Sun, and Wei-Ying Ma. Unified Visual-Semantic Embeddings: Bridging Vision and Language with Structured Meaning Representations. In *CVPR*, 2019. 3

[52] Xiaoshi Wu, Feng Zhu, Rui Zhao, and Hongsheng Li. CORA: Adapting CLIP for Open-Vocabulary Detection with Region Prompting and Anchor Pre-Matching. In *CVPR*, 2023. 2

[53] Chi Xie, Zhao Zhang, Yixuan Wu, Feng Zhu, Rui Zhao, and Shuang Liang. Described Object Detection: Liberating Object Detection with Flexible Expressions. In *NeurIPS*, 2023. 1, 2, 5, 6

[54] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. Modeling Context in Referring Expressions. In *ECCV*, 2016. 1, 2, 5

[55] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *ICLR*, 2023. 2

[56] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-Vocabulary Object Detection Using Captions. In *CVPR*, 2021. 2

[57] Shiyu Zhao, Zhixing Zhang, Samuel Schulter, Long Zhao, BG Vijay Kumar, Anastasis Stathopoulos, Manmohan Chandraker, and Dimitris N Metaxas. Exploiting unlabeled data with vision and language models for object detection. In *ECCV*, pages 159–175. Springer, 2022. 1, 2

[58] Shiyu Zhao, Samuel Schulter, Long Zhao, Zhixing Zhang, Vijay Kumar B. G, Yumin Suh, Manmohan Chandraker, and Dimitris N. Metaxas. Improving pseudo labels for open-vocabulary object detection, 2023. 2

[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 *CVPR*, 2022. 2

[60] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *ECCV*, pages 350–368. Springer, 2022. 5