Title

Learning Visually Grounded Intelligence with Language

Author

Li, Liunian

Publication Date

2024

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
Los Angeles

Learning Visually Grounded Intelligence with Language

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Computer Science

by

Liunian Li

2024

ABSTRACT OF THE DISSERTATION

Learning Visually Grounded Intelligence with Language

by

Liunian Li
Doctor of Philosophy in Computer Science
University of California, Los Angeles, 2024
Professor Kai-Wei Chang, Chair

To build an Artificial Intelligence system that can assist us in daily lives, the ability to understand the world around us through visual input is essential. Prior studies train visual perception models by defining concept vocabularies and annotate data against the fixed vocabulary. It is hard to define a comprehensive set of everything, and thus they are hard to generalize to novel concepts and domains. In this thesis, I turn to language as a scalable and effective tool to build visually grounded models. Intuitively, natural languages are the most effective medium of learning and communication for humans. I will introduce two lines of work to train models to understand the visual world with language as supervision. The first line of work is inspired by masked language modeling such as BERT, and extends that to build contextualized representation models for vision and language. These models can be fine-tuned to perform vision-language tasks such as answering questions about an image. The second line of work uses language to supervise object detection models and enables object detection with prompts, where the users could specify custom needs and domain knowledge in a text prompt, and the model situates its predictions based on the text on the fly.

The dissertation of Liunian Li is approved.

Guy Van den Broeck

Yizhou Sun

Achuta Kadambi

Cho-Jui Hsieh

Kai-Wei Chang, Committee Chair

University of California, Los Angeles

2024

To my family.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First and foremost, I would like to express my greatest gratitude for my advisor Kai-Wei Chang. When I joined the group in the summer of 2018 as an intern, he encouraged me to dream big and patiently guided me through my first research project here. He is always warm and supportive and I have always enjoyed discussing research ideas with him. I cannot express enough how grateful I am to have him as my advisor. I could not have asked for a better advisor.

I would also like to thank the members of my dissertation committee – Professor Guy Van den Broeck, Professor Yizhou Sun, Professor Achuta Kadambi, and Professor Cho-Jui Hsieh. They have given valuable feedback and inspired me to reflect on my research.

I am lucky to have had the opportunity to work with many brilliant mentors and collaborators, who are always so kind to share their research wisdom and inspire me to become a better researcher. Especially, I would like to thank Mark Yatskar and Pengchuan Zhang, who introduced me to the field of vision-language and visual perception. This dissertation would not be possible with their guidance and help.

I would like to express my deepest gratitude to my friends and labmates at UCLA. Ph.D. is a long journey and I want to thank them for always being there through ups and downs. We have had so many fond and unique memories. My life would have been colorless without them. I will forever miss my days at Room 368, and the many restaurants we frequent.

I would like to express my gratitude to my family for their unwavering support. I deeply miss them, and their love and belief in me have been my pillars of strength.

Finally, I extend my deepest appreciation to all who have helped me throughout my journey. I sincerely thank you for your invaluable support. Your guidance and assistance have been instrumental, and I am eternally grateful.

VITA

2015–2019 B.S. (Computer Science), Peking University.

PUBLICATIONS

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, Kai-Wei Chang. VisualBERT: A Simple and Performant Baseline for Vision and Language. 2019.

Liunian Harold Li, Haoxuan You*, Zhecan Wang*, Alireza Zareian, Shih-Fu Chang, Kai-Wei Chang. Unsupervised Vision-and-Language Pre-training Without Parallel Images and Captions. NAACL. 2021.

Liunian Harold Li*, Pengchuan Zhang*, Haotian Zhang*, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, Jianfeng Gao. Grounded Language-Image Pre-training. CVPR. 2022.

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

CHAPTER 1

Introduction

Humans do not learn by reading from text only. Artificial Intelligence (AI) systems need the ability to perceive and understand visual signals. As a motivating example, consider Figure 1.1. Suppose that we want to develop a system that can automatically report accidents from surveillance footage. The grounded agent should be able to detect related entities, draw the connection between the upside-down car and the ambulance, and report in natural language that there has been an accident. To achieve this goal, the model needs to learn the mapping between vision and fine-grained or abstract concepts such as "accident". Teaching the model to perform such visual grounding is especially challenging. Prior studies curate datasets by labeling images with a vocabulary of a few thousand concepts (e.g., ImageNet [DDS09a] has 1,000 classes while VQA [GKS17a] has an answer pool size of less than 4,000). The limited concept pool does not support fine-grained recognition or complex reasoning.

In this thesis, we turn to language as a scalable and effective tool to build visually grounded models. Intuitively, humans do not learn primarily from fixed and task-specific labels. Instead, natural languages are the most effective medium of learning and communication for humans. We propose representation learning methods to train vision-language models on easily-accessible language supervision. Learning with language also enables inference with language, where the user could communicate with the model at inference time. The user specifies needs, new domain knowledge, and constraints in natural language instructions. The model situates its predictions based on the instructions on the

fly without re-training. This does not only greatly enhance the models' ability to generalize and adapt efficiently, but also facilitates humans to collaborate with the machine easily.

In Part 1 of this thesis, we focus on building contextualized representations given an image and associated text, which could be used for tasks such as answering questions regarding an image or identifying hateful memes. VisualBERT [LYY19] (Chapter 2) presents one of the first pre-training methods for vision-language models. People record and describe the world in pictures and text and it is easy to collect such data. We draw inspiration from self-supervised learning and propose reconstructive visually-grounded objectives to encourage a representation model to build cross-modal alignment implicitly. Conceptually, the objectives operate like a cloze test, where the model needs to fill in blanks in a caption according to the image. Thus, the model "reads" through millions of semantic rich image-caption pairs and excels at tasks such as detecting hateful memes. The model is a combination of BERT [DCL19a] and pre-trained object proposals systems such as Faster-RCNN [RHG15]. It seamlessly transfers to many vision-language tasks. Fig. 1.1 shows an attention map of VisualBERT, which captures the intricate association between "accident" and the ambulance and the upside-down car in the associated image.

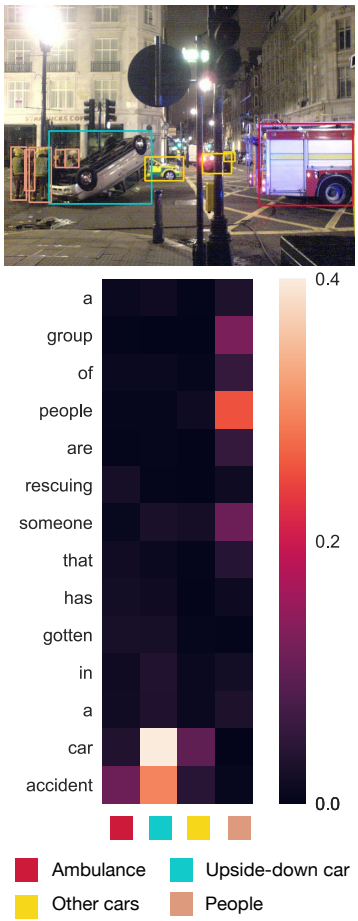

Figure 1.1: Attention map produced by VisualBERT [LYY19].

In many domains, it is expensive to collect grounded (aligned) data. For example, in the medical domain, X-ray scans paired with detailed medical reports are hard to collect while raw X-rays and unpaired literature are easy to collect. In Unsupervised Visu-

alBERT [LYW21] (Chapter 3), we explore unsupervised vision-language pre-training with unaligned image and text corpora. This research direction aligns with the theme of unsupervised and self-supervised learning that moves from heavily-annotated data to unannotated data, e.g. unsupervised machine translation [FSB20]. Instead of relying on image-caption pairs, we rely only on an object detector to provide low-level alignment between regions and their labels. These object labels serve as "anchor points" between the visual and text representation space. We reuse the reconstructive objective from VisualBERT and find that Unsupervised VisualBERT achieves performance close to supervised training.

In Part 1, we rely on visual perception models to extract visual features and build vision-language representations on top of the frozen features. These visual perception models are typically trained to map the visual world to a few thousand pre-defined semantic concepts (the label set), and generalization to novel concepts and domains has been a long-standing challenge. This consequently bottlenecks the performance of vision-language models. In Part 2, we argue that learning visual perception and learning vision-language representations should not be treated as separate problems. We turn the problem of visual perception into a vision-language learning problem. In our work GLIP [LZZ22] (Chapter 4), we focus on object detection, a representative visual perception task, which involves locating objects in an image. We posit that object detection can be cast into a vision-language task, where the model, augmented with a language branch, is trained to recognize objects based on a text query. The model takes in an image and a text prompt – either a synthesized sentence as a concatenation of category names or a natural language sentence. The task is to identify the correspondence between phrases in the prompt and objects (or regions) in an image. The reformulation immediately allows us to pre-train the model on image-caption data with a contrastive objective. During inference, we can easily transfer the model to new tasks by writing down the target categories as a text prompt (e.g., "Detect: Bicycle. Car. Ambulance.").

Can we go beyond just using object names in the text prompt? Can we query the model with complex language expressions that include specifications of fine-grained details, such as colors, shapes, and relations? We find that simply incorporating language descriptions into queries does not guarantee accurate interpretation by the models. They often disregards contextual information in the language descriptions and instead relies heavily on detecting objects solely by their names. In Chapture 5, we propose DesCo [LDP23], a description-conditioned way to learn object detection with rich language descriptions. We employ a large language model as a commonsense knowledge engine to generate rich language descriptions of objects. Then we design context-sensitive queries to improve the model's ability in deciphering intricate nuances embedded within descriptions and enforce the model to focus on context rather than object names alone. DesCo can interpret semantic-rich queries much more accurately and significantly outperforms GLIP.

An overview of the chapters are as follows:

- Chapter 1 introduces the challenge of learning vision-language models and presents an overview of the dissertation.
- Chapter 2 presents VisualBERT, one of the first vision-language pre-training methods.
- Chapter 3 presents Unsupevised VisualBERT, which can be trained without aligned image-text pairs.
- Chapter 4 presents GLIP, one of the first language-based open-world object detection systems.
- Chapter 5 presents DesCo, which learns object detection with rich language descriptions and allows semantic-rich prompts at inference time.
- Chapter 6 concludes this dissertation and discusses future research directions.

Part I

Vision-Language Pre-Training

CHAPTER 2

Pre-training with Image-Tex Pairs

Research on learning multi-modal grounding has a long history. Starting from creating ImageNet based on the WordNet vocabulary to collecting over 1 million image-question-answer pairs (VQA), curated datasets with human annotations have greatly accelerated multi-modal research. However, such human annotations in the form of image/object labels and short answers have fundamental limitations. For example, VQA has an answer pool size of less than 4,000, while the largest object detection dataset has a concept pool of less than 2,000. The limited concept pool does not support the learning of fine-grained concepts or complex reasoning, which are indispensable for vision-language grounding.

We posit that semantic-rich multi-modal knowledge exists in a natural form: image-caption pairs. People record and describe the world in pictures and text on the Internet and it is easy to collect millions of such data. In this chapter, we introduce VisualBERT, a simple and flexible representation model for a broad range of vision-and-language tasks. VisualBERT integrates BERT [DCL18], Transformer-based model [VSP17] for natural language processing, and pre-trained object proposals systems such as Faster-RCNN [RHG15] and it can be applied to a variety of vision-and-language tasks. In particular, image features extracted from object proposals are treated as unordered input tokens and fed into VisualBERT along with text. The text and image inputs are jointly processed by multiple Transformer layers in VisualBERT (See Figure 2.2). The rich interaction among words and object proposals allows the model to capture the intricate associations between text

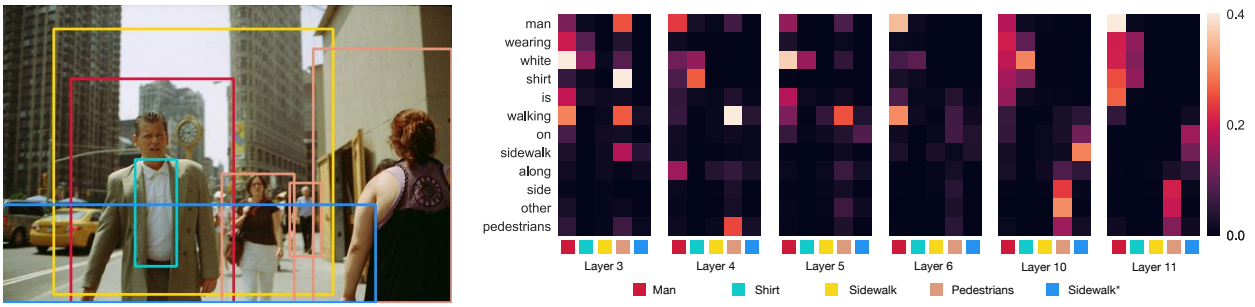

Figure 2.1: Attention weights of some selected heads in VisualBERT. In high layers (e.g., the 10-th and 11-th layer), the model can implicitly grounding visual concepts (e.g., "other pedestrians" and "man wearing white shirt"). The model also captures certain syntactic dependency relations (e.g., "walking" is aligned to the *man* region in the 6-th layer). The model also refines its understanding over the layers, incorrectly aligning "man" and "shirt" in the 3-rd layer but correcting them in higher layers.

and image.

Similar to BERT, pre-training VisualBERT on external resource can benefit downstream applications. We pre-train VisualBERT on image caption data, where *detailed semantics* of an image are expressed in natural language. We propose two *visually-grounded* language model objectives for pre-training: 1) part of the text is masked and the model learns to predict the masked words based on the remaining text and visual context; 2) the model is trained to determine whether the provided text matches the image. We show that such pre-training on image caption data is important for VisualBERT to learn transferable text and visual representations.

We conduct comprehensive experiments on four vision-and-language tasks: (1) visual question answering (VQA 2.0 [GKS17a]), (2) visual commonsense reasoning (VCR [ZBF19]), (3) natural language for visual reasoning (NLVR^2, [SZZ19]), and (4) region-to-phrase grounding (Flickr30K, [PWC15]). Results demonstrate that by pre-training VisualBERT on the COCO image caption dataset [CFL15], VisualBERT outperforms or rivals with the state-of-the-art models. We further provide detailed ablation study to justify our design choices.

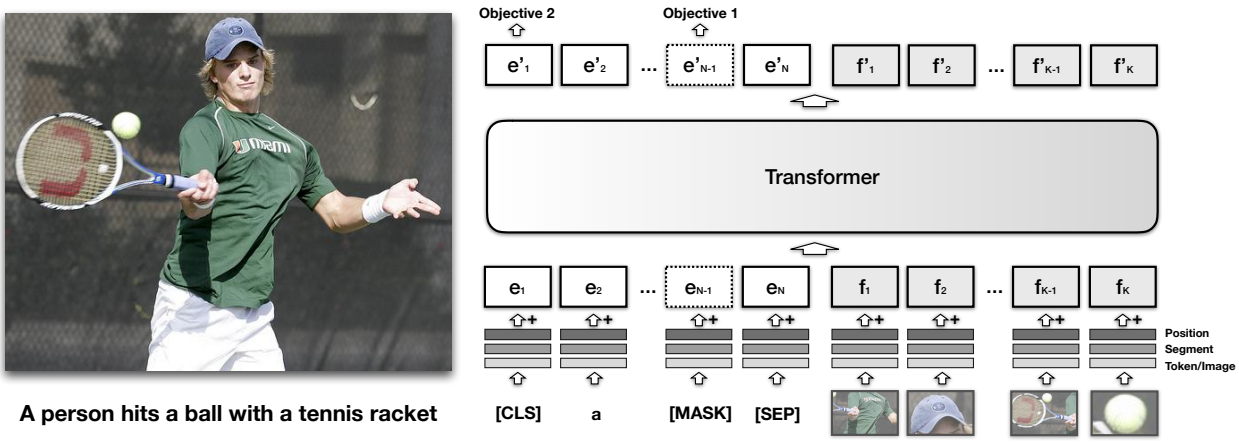

Figure 2.2: The architecture of VisualBERT. Image regions and language are combined with a Transformer to allow the self-attention to discover implicit alignments between language and vision. It is pre-trained with a masked language modeling (Objective 1), and sentence-image prediction task (Objective 2), on caption data and then fine-tuned for different tasks. See §2.1.2 for more details.

Further quantitative and qualitative analysis reveals how VisualBERT allocates attention weights to align words and image regions internally. We demonstrate that through pre-training, VisualBERT learns to ground entities and encode certain dependency relationships between words and image regions, which attributes to improving the model's understanding on the detailed semantics of an image (see an example in Figure 2.1).

2.1 VisualBERT

In this section we introduce VisualBERT, a model for learning joint contextualized representations of vision and language.

2.1.1 Architecture

The core of our idea is to reuse the self-attention mechanism within the Transformer to implicitly align elements of the input text and regions in the input image. In addition to all the components of BERT, we introduce a set of visual embeddings to model an image.

Each visual embedding corresponds to a bounding region in the image, derived from an object detector.

Each visual embedding is computed by summing three embeddings: (1) a visual feature representation of the bounding region, computed by a convolutional neural network, (2) a segment embedding indicating it is an image embedding as opposed to a text embedding, and (3) a position embedding, which is used when alignments between words and bounding regions are provided as part of the input, and set to the sum of the position embeddings corresponding to the aligned words (see VCR in §2.2). The visual embeddings are then passed to the multi-layer Transformer along with the original set of text embeddings, allowing the model to implicitly discover useful alignments between both sets of inputs, and build up a new joint representation.[1]

2.1.2 Training VisualBERT

We would like to adopt a similar training procedure as BERT but VisualBERT must learn to accommodate both language and visual input. Therefore we reach to a resource of paired data: MS-COCO [CFL15] that contains images each paired with 5 independent captions. Our training procedure contains three phases:

Task-agnostic pre-training. Here we train VisualBERT on COCO using two *visually-grounded* language model objectives. (1) Masked language modeling with the image. Some elements of text input are masked and must be predicted but vectors corresponding to image regions are not masked. (2) Sentence-image prediction. For COCO, where there are multiple captions corresponding to one image, we provide a text segment consisting of two captions. One of the caption is describing the image, while the other has a 50% chance to be another corresponding caption and a 50% chance to be a randomly drawn caption. The model is trained to distinguish these two situations.

[1] If text and visual input embeddings are of different dimension, we project the visual embeddings into a space of the same dimension as the text embeddings.

Task-specific pre-training. Before fine-tuning VisualBERT to a downstream task, we find it beneficial to train the model using the data of the task with the masked language modeling with the image objective. This step allows the model to adapt to the new target domain.

Fine-tuning. This step mirrors BERT fine-tuning, where a task-specific input, output, and objective are introduced, and the Transformer is trained to maximize performance on the task.

2.2 Experiment

We evaluate VisualBERT on four different types of vision-and-language applications: (1) Visual Question Answering (VQA 2.0) [GKS17a], (2) Visual Commonsense Reasoning (VCR) [ZBF19], (3) Natural Language for Visual Reasoning (NLVR^2) [SZZ19], and (4) Region-to-Phrase Grounding (Flickr30K) [PWC15]. For all tasks, we use the Karpathy train split [KF15] of COCO for task-agnostic pre-training, which has around 100k images with 5 captions each. The Transformer encoder in all models has the same configuration as BERT$_{Base}$: 12 layers, a hidden size of 768, and 12 self-attention heads. The parameters are initialized from the pre-trained BERT$_{Base}$ parameters released by [DCL18].

For the image representations, each dataset we study has a different standard object detector to generate region proposals and region features. To compare with them, we follow their settings, and as a result, different image features are used for different tasks (see details in the subsections). [2] For consistency, during task-agnostic pre-training on COCO, we use the same image features as in the end tasks. For each dataset, we evaluate three variants of our model:

VisualBERT: The full model with parameter initialization from BERT that undergoes pre-training on COCO, pre-training on the task data, and fine-tuning for the task.

[2]Ideally, we can use the best available detector and visual representation for all tasks, but we would like to compare methods on similar footing.

VisualBERT w/o Early Fusion: VisualBERT but where image representations are not combined with the text in the initial Transformer layer but instead at the very end with a new Transformer layer. This allows us to test whether interaction between language and vision throughout the whole Transformer stack is important to performance.

VisualBERT w/o COCO Pre-training: VisualBERT but where we skip task-agnostic pre-training on COCO captions. This allows us to validate the importance of this step.

Following [DCL18], we optimize all models using Adam [KB15]. We set the warm-up step number to be 10% of the total training step count unless specified otherwise. Batch sizes are chosen to meet hardware constraints and text sequences whose lengths are longer than 128 are capped. Experiments are conducted on Tesla V100s and GTX 1080Tis, and all experiments can be replicated on at most 4 Tesla V100s each with 16GBs of GPU memory. Pre-training on COCO generally takes less than a day on 4 cards while task-specific pre-training and fine-tuning usually takes less. Other task-specific training details are in the corresponding sections.

2.2.1 VQA

Given an image and a question, the task is to correctly answer the question. We use the VQA 2.0 [GKS17a], consisting of over 1 million questions about images from COCO. We train the model to predict the 3,129 most frequent answers and use image features from a ResNeXt-based Faster RCNN pre-trained on Visual Genome [JNC18].

We report the results in Table 2.1, including baselines using the same visual features and number of bounding region proposals as our methods (first section), our models (second section), and other incomparable methods (third section) that use external question-answer pairs from Visual Genome (+VG) , multiple detectors [YLY19] (+Multiple Detectors) and ensembles of their models. In comparable settings, our method is significantly simpler and outperforms existing work.

Model	Test-Dev	Test-Std
Pythia v0.1 [JNC18]	68.49	-
Pythia v0.3 [SNS19]	68.71	-
VisualBERT w/o Early Fusion	68.18	-
VisualBERT w/o COCO Pre-training	70.18	-
VisualBERT	70.80	71.00
Pythia v0.1 + VG + Other Data Augmentation [JNC18]	70.01	70.24
MCAN + VG [YYC19]	70.63	70.90
MCAN + VG + Multiple Detectors [YYC19]	72.55	-
MCAN + VG + Multiple Detectors + BERT [YYC19]	72.80	-
MCAN + VG + Multiple Detectors + BERT + Ensemble [YYC19]	75.00	75.23

Table 2.1: Model performance on VQA. VisualBERT outperforms Pythia v0.1 and v0.3, which are tested under a comparable setting.

2.2.2 VCR

VCR consists of 290k questions derived from 110k movie scenes, where the questions focus on visual commonsense. The task is decomposed into two multi-choice sub-tasks wherein we train individual models: question answering (Q $\rightarrow$ A) and answer justification (QA $\rightarrow$ R). Image features are obtained from a ResNet50 [HCH16] and "gold" detection bounding boxes and segmentations provided in the dataset are used[3]. The dataset also provides alignments between words and bounding regions that are referenced to in the text, which we utilize by using the same position embeddings for matched words and regions.

[3]In the fine-tuning stage, for VisualBERT (with/without Early Fusion), ResNet50 is fine-tuned along with the model as we find it beneficial. For reference, VisualBERT with a fixed ResNet50 gets 51.4 on the dev set for Q $\rightarrow$ AR. The ResNet50 of VisualBERT w/o COCO Pre-training is not fine-tuned with the model such that we could compare it with R2C fairly.

Results on VCR are presented in Table 2.2. We compare our methods against the model released with the dataset which builds on BERT (R2C) and list the top performing single model on the leaderboard (B2T2). Our ablated VisualBERT w/o COCO Pre-training enjoys the same resource as R2C, and despite being significantly simpler, outperforms it by a large margin. The full model further improves the results. Despite substantial domain difference between COCO and VCR, with VCR covering scenes from movies, pre-training on COCO still helps significantly.

Model	Q → A		QA → R		Q → AR	
	Dev	Test	Dev	Test	Dev	Test
R2C [ZBF19]	63.8	65.1	67.2	67.3	43.1	44.0
B2T2 (Leaderboard; Unpublished)	-	72.6	-	75.7	-	55.0
VisualBERT w/o Early Fusion	70.1	-	71.9	-	50.6	-
VisualBERT w/o COCO Pre-training	67.9	-	69.5	-	47.9	-
VisualBERT	70.8	71.6	73.2	73.2	52.2	52.4

Table 2.2: Model performance on VCR. VisualBERT w/o COCO Pre-training outperforms R2C, which enjoys the same resource while VisualBERT further improves the results.

2.2.3 NLVR^2

NLVR^2 is a dataset for joint reasoning about natural language and images, with a focus on semantic diversity, compositionality, and visual reasoning challenges. The task is to determine whether a natural language caption is true about a pair of images. The dataset consists of over 100k examples of English sentences paired with web images. We modify the segment embedding mechanism in VisualBERT and assign features from different images with different segment embeddings. We use an off-the-shelf detector from Detec-

tron [GRG18] to provide image features and use 144 proposals per image.[4]

Results are in Table 2.3. VisualBERT w/o Early Fusion and VisualBERT w/o COCO Pre-training surpass the previous best model MaxEnt by a large margin while VisualBERT widens the gap.

Model	Dev	Test-P	Test-U	Test-U (Cons)
MaxEnt [SZZ19]	54.1	54.8	53.5	12.0
VisualBERT w/o Early Fusion	64.6	-	-	-
VisualBERT w/o COCO Pre-training	63.5	-	-	-
VisualBERT	67.4	67.0	67.3	26.9

Table 2.3: Comparison with the state-of-the-art model on NLVR[2]. The two ablation models significantly outperform MaxEnt while the full model widens the gap.

2.2.4 Flickr30K Entities

Flickr30K Entities dataset tests the ability of systems to ground phrases in captions to bounding regions in the image. The task is, given spans from a sentence, selecting the bounding regions they correspond to. The dataset consists of 30k images and nearly 250k annotations. We adapt the setting of BAN [KJZ18], where image features from a Faster R-CNN pre-trained on Visual Genome are used. For task specific fine-tuning, we introduce an additional self-attention block and use the average attention weights from each head to predict the alignment between boxes and phrases. For a phrase to be grounded, we take whichever box receives the most attention from the last sub-word of the phrase as the model prediction.

Results are listed in Table 2.4. VisualBERT outperforms the current state-of-the-art

[4]We conducted a preliminary experiment on the effect of the number of object proposals we keep per image. We tested models with 9, 18, 36, 72, and 144 proposals, which achieve an accuracy of 64.8, 65.5, 66.7, 67.1, and 67.4 respectively on the development set.

model BAN. In this setting, we do not observe a significant difference between the ablation model without early fusion and our full model, arguing that perhaps a shallower architecture is sufficient for this task.

Model	R@1		R@5		R@10		Upper Bound	
	Dev	Test	Dev	Test	Dev	Test	Dev	Test
BAN [KJZ18]	-	69.69	-	84.22	-	86.35	86.97	87.45
VisualBERT w/o Early Fusion	70.33	-	84.53	-	86.39	-		
VisualBERT w/o COCO Pre-training	68.07	-	83.98	-	86.24	-	86.97	87.45
VisualBERT	70.40	71.33	84.49	84.98	86.31	86.51		

Table 2.4: Comparison with the state-of-the-art model on the Flickr30K. VisualBERT holds a clear advantage over BAN.

2.3 Ablation Study

In this section we conduct extensive analysis on what parts of our approach are important to VisualBERT's strong performance. We conduct our ablation study on NLVR^2 and include two ablation models in §2.2 and four additional variants of VisualBERT for comparison. For ease of computations, all these models are trained with only 36 features per image (including the full model). Our analysis (Table 2.5) aims to investigate the contributions of the following four components in VisualBERT:

C1: Task-agnostic pre-training. We investigate the contribution of task-agnostic pre-training by entirely skipping such pre-training (VisualBERT w/o COCO Pre-training) and also by pre-training with only text but no images from COCO (VisualBERT w/o Grounded Pre-training). Both variants underperform, showing that pre-training on paired vision and language data is important.

C2: Early fusion. We include VisualBERT w/o Early Fusion introduced in §2.2 to verify the importance of allowing early interaction between image and text features, con-

	Model	Dev
	VisualBERT	66.7
C1	VisualBERT w/o Grounded Pre-training	63.9
C1	VisualBERT w/o COCO Pre-training	62.9
C2	VisualBERT w/o Early Fusion	61.4
C3	VisualBERT w/o BERT Initialization	64.7
C4	VisualBERT w/o Objective 2	64.9

Table 2.5: Performance of the ablation models on NLVR^2. Results confirm that task-agnostic pre-training (C1) and early fusion of vision and language (C2) are essential for VisualBERT.

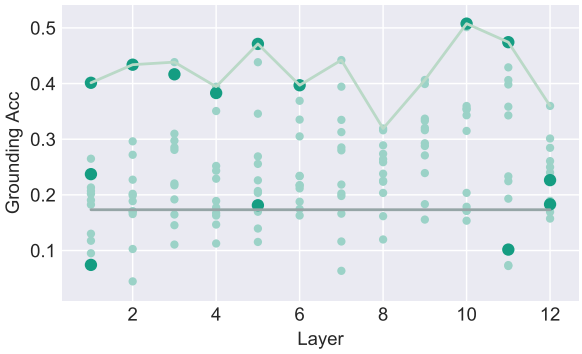

Figure 2.3: Entity grounding accuracy of the attention heads of VisualBERT. The rule-based baseline is dawn as the grey line. We find that certain heads achieves high accuracy while the accuracy peaks at higher layers.

firming again that multiple interaction layers between vision and language are important.

C3: BERT initialization. All the models discussed so far are initialized with parameters from a pre-trained BERT model. To understand the contributions of the BERT initialization, we introduce a variant with randomly initialized parameters. The model is then trained as the full model. While it does seem weights from language-only pre-trained BERT are important, performance does not degrade as much as we expect, arguing that the model is likely learning many of the same useful aspects about grounded language during COCO pre-training.

C4: The sentence-image prediction objective. We introduce a model without the sentence-image prediction objective during task-agnostic pre-training (VisualBERT w/o Objective 2). Results suggest that this objective has positive but less significant effect, compared to other components.

Overall, the results confirm that the most important design choices are task-agnostic

pre-training (C1) and early fusion of vision and language (C2). In pre-training, both the inclusion of additional COCO data and using both images and captions are paramount.

2.4 Dissecting Attention Weights

In this section we use Flickr30K as a diagnostic dataset to understand whether VisualBERT's pre-training phase actually allows the model to learn implicit alignments between bounding regions and text phrases. We show that many attention heads within VisualBERT accurately track grounding information and that some are even sensitive to syntax, attending from verbs to the bounding regions corresponding to their arguments within a sentence. Finally, we show qualitative examples of how VisualBERT resolves ambiguous groundings through multiple layers of the Transformer.

2.4.1 Entity Grounding

First, we attempt to find attention heads within VisualBERT that could perform entity grounding, i.e., attending to the corresponding bounding regions from entities in the sentence. Specifically, we use the ground truth alignments from the evaluation set of Flickr30K. For each entity in the sentence and for each attention head in VisualBERT, we look at the bounding region which receives the most attention weight. Because a word is likely to attend to not only the image regions but also words in the text, for this evaluation, we mask out the head's attention to words and keep only attention to the image regions. Then we compute the how often the attention of a particular head agrees with the annotations in Flickr30K.

We report this accuracy[5], for all 144 attention heads in VisualBERT, organized by layer, in Figure 2.3. We also consider a baseline that always chooses the region with the high-

[5] Despite that some heads are accurate at entity grounding, they are not actively attending to the image regions. For example, a head might be only allocating 10% of its attention weights to all image regions, but it assigns the most of the 10% weights to the correct region. We represent heads paying on average less than 20% of its attention weights from the entities to the regions with smaller and light-colored dots and others with larger and bright dots.

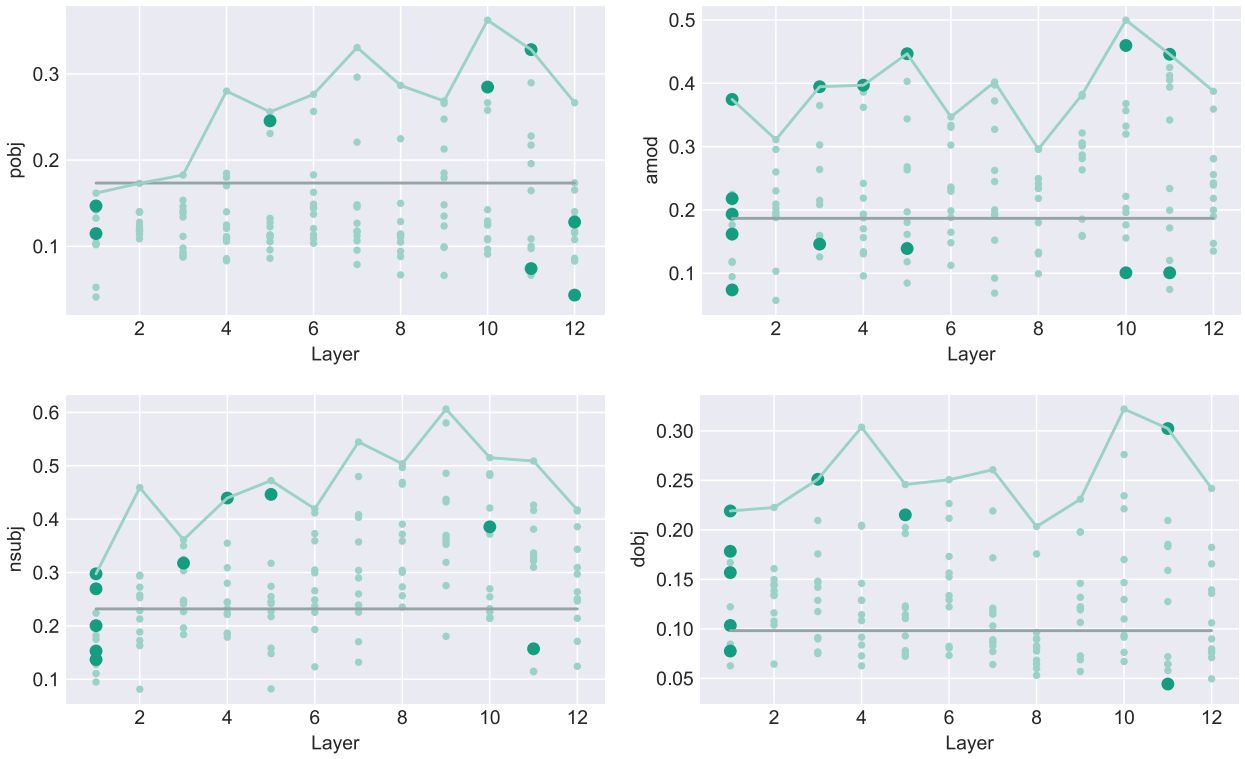

Figure 2.4: Accuracy of attention heads of VisualBERT for predicting four specific dependency relationships ("pobj", "amod", "nsubj", and "dobj") across modality. The grey lines denote a baseline that always chooses the region with the highest detection confidence. We observe that VisualBERT is capable of detecting these dependency relationships without direct supervision.

est detection confidence. We find that VisualBERT achieves a remarkably high accuracy though it is not exposed to any direct supervision for entity grounding. The grounding accuracy also seems to improve in higher layers, showing the model is less certain when synthesizing the two inputs in lower layers, but then becomes increasingly aware of how they should align. We show examples of this behavior in §2.4.3.

2.4.2 Syntactic Grounding

Given that many have observed that the attention heads of BERT can discover syntactic relationships [VTM19, CKL19], we also analyze how grounding information is passed through syntactic relationships that VisualBERT may have discovered. In particular, given two words that are connected with a dependency relation, $w_1 \xrightarrow{r} w_2$, we would like to know how often the attention heads at w_2 attend to the regions corresponding to w_1, and vice-versa. For example, in Figure 2.1, we would like to know if there is an attention head that, at the word "walking", is systematically attending to the region corresponding to the "man", because "man" and "walking" are related through a "nsubj" relation, under the Stanford Dependency Parsing formalism [DM08].

To evaluate such syntactic sensitivity in VisualBERT, we first parse all sentences in Flickr30K using AllenNLP's dependency parser [DM17, GGN18]. Then, for each attention head in VisualBERT, given that two words have a particular dependency relationship, and one of them has a ground-truth grounding in Flickr30K, we compute how accurately the head attention weights predict the ground-truth grounding. Examination of all dependency relationships shows that in VisualBERT, there exists at least one head for each relationship that significantly outperforms guessing the most confident bounding region. We highlight a few particularly interesting dependency relationships in Figure 2.4. Many heads seem to accurately associate arguments with verbs (i.e. "pobj", "nsub", and "dobj" dependency relations), arguing that VisualBERT is resolving these arguments, implicitly and without supervision, to visual elements.

2.4.3 Qualitative Analysis

Finally, we showcase several interesting examples of how VisualBERT changes its attention over the layers when processing images and text, in Figure 2.1 and Figure 2.5. To generate these examples, for each ground-truth box, we show a predicted bounding region closest to it and manually group the bounding regions into different categories. We

also include regions that the model is actively attending to, even if they are not present in the ground-truth annotation (marked with an asterisk). We then aggregate the attention weights from words to those regions in the same category. We show the best heads of 6 layers that achieve the highest entity grounding accuracy.

Overall, we observe that VisualBERT seems to refine alignments through successive Transformer layers. For example, in the bottom left image in Figure 2.5, initially the word "husband" and the word "woman" both have significant attention weight on regions corresponding to the woman. By the end of the computation, VisualBERT has disentangled the woman and man, correctly aligning both. Furthermore, there are many examples of syntactic alignments. For example, in the same image, the word "teased" aligns to both the man and woman while "by" aligns to the man. Finally, some coreference seems to be resolved, as, in the same image, the word "her" is resolved to the woman.

Figure 2.5: Attention weights of some selected heads in VisualBERT on 6 examples. The first column is 3 random examples where alignments match Flickr30k annotations while the second column is 3 random examples where alignments do not match.

CHAPTER 3

Pre-training with Unaligned Image and Text Data

VisualBERT, along with other pre-trained vision-and-language (V&L) models [LDD19, TB19, SZC19, CLY20], have achieved high performance on various V&L tasks. However, different from pre-trained language models, such as BERT [DCL19a], which are trained on easily-accessible unannotated text corpora, existing V&L models are still a step away from self-supervision. They require a massive amount of aligned text-image pairs for "mask-and-predict" pre-training. Such aligned data are costly to collect and hard to scale up. For example, the widely used MS-COCO dataset [CFL15] requires extensive annotation from crowd workers.[1]

In this chapter, we explore *unsupervised V&L pre-training* with unaligned image and text corpora.[2] This research direction aligns with the theme of unsupervised and self-supervised learning that moves from heavily-annotated data to unannotated data, e.g. unsupervised machine translation [LCD18] and unsupervised image captioning [FML19]. Unsupervised V&L pre-training is highly desirable as in many domains, aligned data is scarce (e.g. multimodal hate speech detection [KFM20] and the medical domain [LWL20]) and it is easier to collect unaligned text and images. In addition to its practical implication, our endeavour challenges the widely held notion that image-caption corpora is indispensable for pre-training [LDD19] and brings valuable insight into the role that aligned data

[1] Other datasets also require cumbersome curation. For example, while Conceptual Captions is crawled from the web, the authors report that from 5 billion images gathered over the Internet, only 3 million have paired high-quality captions after filtering [SDG18, CSD21].

[2] Following [LCD18] and [FML19], we use the term "unsupervised" to refer to pre-training with unaligned data, while "supervised" refers to pre-training with aligned text and images.

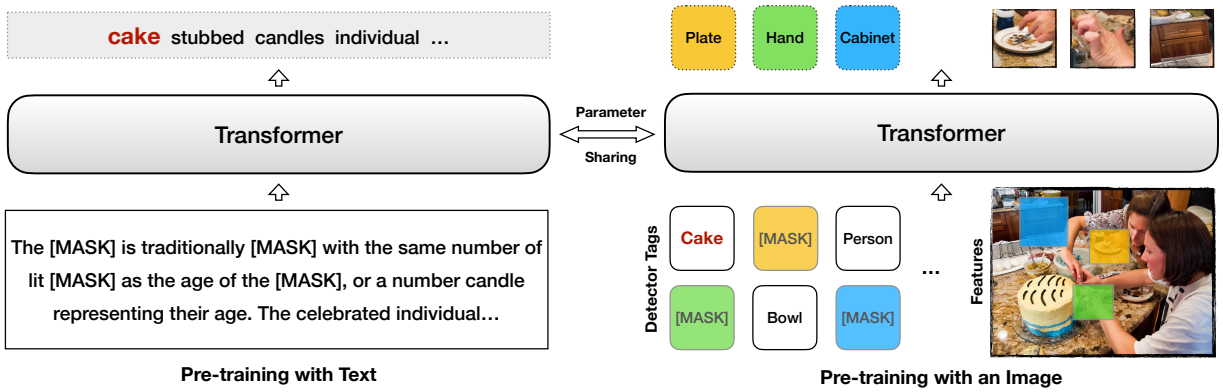

Figure 3.1: An illustration of pre-training without aligned data. Given text, the model is trained to predict masked words; given an image, the model is trained to predict masked regions and detector tags. The semantic class "cake" appears in both the language modality and the visual modality and is linked through the detector tags. Note that we do not require a text segment with the word *cake* to appear together with the image. Rather, we assume that as long as the text corpora are general enough, the word *cake* will appear in the textual modality eventually. The model can thus learn V&L representations from such weak supervision signals.

play in V&L pre-training.

We are inspired by works on multi-lingual contextual language models [PSG19]. If we treat an image as a set of regions and each region as a visual token [DBK20], V&L models share a similar goal with multi-lingual models as they both learn shared representations across different domains. Although a multi-lingual language model pre-trained on non-parallel corpora such as mBERT [DCL19b] cannot align or translate languages out-of-the-box, its representation spaces for different languages can be easily aligned with a linear probe [CWL20]. This property suggests the existence of universal latent symmetries in the unaligned contextual embedding spaces and is believed to contribute to mBERT's cross-lingual transfer ability. Thus we hypothesize that strong V&L representations can be similarly learned by "mask-and-predict" pre-training on unaligned language and vision

data.

We propose unsupervised V&L pre-training with unaligned text and images (see an illustration in Figure 3.1). Specifically, we take VisualBERT [LYY19] as a running example and apply unsupervised pre-training, resulting in Unsupervised VisualBERT (U-VisualBERT). The model takes the form of a single Transformer that can accept inputs from both modalities. During each step of pre-training, unlike the existing models that observe a batch of text-image pairs, our model observes either a batch of text segments or a batch of images. When provided with text, part of the text is masked and the model is trained to predict the masked words; when provided with an image, part of the image regions are masked and the model is trained to predict properties of the masked regions.

To further encourage cross-modal fusion, we leverage the tags from an object detector as "anchor points" [LYL20]. For every object, we append its detected tag as a word to the visual input. The mask-and-predict objective is applied to the tags. For instance, for the image in Figure 3.1, the model can observe "*cake*" appears naturally as a word, a tag, and an image region. The direct typing of image regions and words can be learned and serves as a starting point for further alignment. The function of the detector tags resembles that of the "overlapping vocabulary" in multi-lingual language models, i.e., identical strings that appear in different languages with the same meanings (e.g., "DNA" appears in both English and French). As the "overlapping vocabulary" improves cross-lingual transfer [WD19], we argue the detector tags can improve cross-modal grounding.

We first conduct controlled experiments by pre-training on an English image-caption corpus without providing the alignment, following unsupervised machine translation and image captioning [GJC19]. Results on four English V&L benchmarks (VQA [GKS17b], NLVR^2 [SZZ19], Flickr30K Image Retrieval [PWC15], and RefCOCO+ [YPY16a]) show that U-VisualBERT achieves comparable performance as models with access to text-image pairs (Section 3.3).

Additionally, our approach is effective in practical settings, 1) when using indepen-

dently collected images and captions and 2) when using images and general-domain text (BookCorpus [ZKZ15]) without any captions (Section 3.4.1). Quantitative and qualitative analysis confirms the anchoring effect of the detector tags (Section 3.4.2). As a byproduct, we conduct preliminary experiments to show the promise of the approach in a semi-supervised setting, where a hybrid model pre-trained with both aligned and additional unaligned data surpasses a model pre-trained only on aligned data. (Section 3.4.3). The above experiments demonstrate the wide applicability of our method.

3.1 Related Work

Pre-trained V&L transformers. Various V&L models that are pre-trained with a "mask-and-predict" objective on aligned text-image data have been proposed [LDD19, TB19, LYY19, SZC19, CLY20, LDF20, ZPZ20a, HZL20, YTY20, GCL20]. Two kinds of designs have been proposed. Two-stream models [LDD19, TB19, YTY20] utilize separate Transformers [VSP17] for each modality and a cross-modality module is adopted. Single-stream models [LYY19, SZC19, CLY20] directly input the text and visual embeddings into one single Transformer. They have been widely used by downstream tasks [KFM20]. Probing tasks [CGC20a] confirm that they capture useful V&L information after pre-training.

Two studies also try to incorporate "tag" information during pre-training. Oscar [LYL20] adds detected tags as additional signals when pre-training with aligned data. We, however, do so for pre-training with unaligned data and show that the tags serve a more important role in unsupervised pre-training (Section 3.4.2). VIVO [HYL20] targets novel object captioning. They use manually annotated image-tag data for pre-training and image-caption data for fine-tuning. We do not use manually annotated data and the tags are noisily generated by a detector.

Self-supervised Representation Learning Self-supervision involves creating supervision objectives from natural data, often by corrupting the input and training the

model to reconstruct the input [KZB19] or contrastive learning [CKN20]. Self-supervised training on language [PNI18, DCL19a] such as BERT has been proven useful for various NLP tasks [LGB19], while self-supervised visual representation learning has been centered around learning low-level visual features, in hope of enhancing the backbone CNN [DGE15, PKD16, NF16, CKN20]. In this study, we conduct V&L pre-training by optimizing a reconstructive objective on unlabeled language-only and image-only data. Thus, our proposed model could be regarded as "self-supervised". Notably, our contextual visual representation is built on top of a pre-trained detector, operating at a level above local visual features.

Unsupervised multi-lingual language model. This work is inspired by multi-lingual representations trained without parallel corpora [DCL19b]. They are effective for cross-lingual transfer, which involves learning a model in one language and applying it to another with no additional training. Studies [WD19, CWL20] have confirmed several design choices that facilitate such transfer, e.g. shared parameters and overlapping vocabularies across languages, and we make similar design choices in U-VisualBERT (Section 3.2.2). We argue that multi-lingual representations bear resemblance to multi-modal representations as both seek to encode the alignment between two domains [CGC20b].

Unsupervised grounding learning. Prior works have explored learning grounding with weak or no supervision [RRH16, XSJ17, WTS20]. Closest to this study is unsupervised image captioning [FML19, LRN19, GJC19], which conducts image captioning with unpaired images and captions. Similar to this work, the detector tags serve as the anchor points for image captioning. However, unsupervised image captioning still requires captions, while our approach works with easy-to-collect general-domain text without any caption text (Section 3.4.1).

3.2 Approach

We first take Supervised VisualBERT (S-VisualBERT) as an example and illustrate how a typical V&L model is pre-trained with aligned data. Then we introduce unsupervised V&L pre-training, and the resulting model Unsupervised VisualBERT (U-VisualBERT).

3.2.1 Background

As mentioned in Section 3.1, there are several V&L representation learning methods based on BERT. We take Supervised VisualBERT (S-VisualBERT) as an example, which will also be used as a baseline in the experiments. S-VisualBERT is modified from the original VisualBERT [LYY19] and augmented with the visual objectives from LXMERT [TB19] and detector tags similar to Oscar [LYL20] (discussed in detail in Section 3.2.2).

Every input to S-VisualBERT contains a text segment T and an image I. The text and the image are first mapped into embedding vectors respectively. Text embeddings $\boldsymbol{T}$ is a matrix in which each column vector represents the embedding of a subword in the text sequence, i.e. $\boldsymbol{T} = [\boldsymbol{w}_{1:n}]$. Following BERT, each subword embedding $\boldsymbol{w}_i$ is the sum of its token, position, and segment embedding. Image embeddings $\boldsymbol{I}$ include both the image region embeddings $\boldsymbol{r}_{1:m}$ and the detector tag embeddings $\boldsymbol{d}_{1:l}$ (see Section 3.2.2 for details). Each region embedding $\boldsymbol{r}_i$ is the sum of a visual feature vector from the detector and a spatial box coordinate embedding [TB19]. The text and visual embeddings are then passed through a Transformer to built contextual representations.

The model is pre-trained with a mask-and-predict objective. Given a text-image pair $[T, I]$ from the aligned dataset D, we randomly mask out some words w_i, some regions r_j, and some tags d_k to obtain masked $[\tilde{T}, \tilde{I}]$. The model is trained to predict the masked words, the properties of the masked regions, and the masked tags given $[\tilde{T}, \tilde{I}]$. The pre-training objective can be summarized as:

$$\min_{\theta} \sum_{[T,I]\in D} L_{T+I+M}\left(f_\theta([\tilde{T},\tilde{I}]),[T,I]\right).$$

f_θ represents the embedding layer and the multi-layer Transformer. L_{T+I+M} is the sum of 1) the masked language model loss L_T, 2) the image reconstruction loss L_I, and 3) an "text-image match" objective L_M. Specifically, L_I includes a *tag reconstruction* loss L_I^{tag} (more details in Section 3.2.2) and the two visual losses as in LXMERT [TB19]: the *region feature regression* loss L_I^{ref}, which forces the model to regress to the visual vector, and the *noisy label classification* loss L_I^{cls}, which predicts the detected labels of masked objects with the cross-entropy loss. With a probability of 0.5, we provide the model with a mismatched text-image pair instead of a matched pair, and L_M asks the model to predict whether the image matches the text. After the model is pre-trained, it can be fine-tuned for V&L tasks similar to how BERT is fine-tuned for NLP tasks.

3.2.2 Unsupervised Pre-training

We introduce the two core design choices of unsupervised pre-training: masked pre-training with unaligned data and the detector tags.

Masked pre-training with unaligned data. We assume access to a text corpus D_T and an image corpus D_I for pre-training. During every pre-training step, we randomly sample either a batch of text from D_T or a batch of images from D_I. No alignment between text and images is provided to the model. When pre-training with a text segment T, the model is trained to reconstruct T given the masked $\tilde{T}$.[3] When pre-training with an image I, the model is trained to reconstruct I given the masked $\tilde{I}$. A single Transformer is used throughout two modalities (i.e. θ shared across modalities). The pre-training objective can be summarized as:

[3]We adopt the next sentence prediction task in BERT when long documents are available.

$$\min_{\theta} \sum_{T \in D_T} L_T(f_\theta(\tilde{T}), T) + \sum_{I \in D_I} L_I(f_\theta(\tilde{I}), I).$$

After pre-training, the model is fine-tuned on downstream tasks just as its supervised counterpart, with the input being a text-image pair.

Detector tags. While masked pre-training with unaligned data in itself achieves non-trivial performance (Section 3.4.2), we find it beneficial to provide noisy alignment signals in the form of the detector tags. When modeling an image I, for each region detected, we append the tag outputted by the object detector to the input. The detector [RHG15] is pre-trained on a general object detection dataset [KZG17, AHB18] and the tags are essentially a bag of words that provide some noisy grounding signals to the model. During pre-training, we apply the mask-and-predict objective to the tags, which further encourages grounding.

We process the detector tags as a subword sequence $d_{1:l}$ with spatial coordinates.[4] Every tag subword is embedded as the sum of its token embedding and a spatial coordinate embedding. The token embedding is the same as the token embedding used in text modeling, while the spatial coordinate embedding is the same as the coordinate embedding of the corresponding region. The coordinate embedding allows the model to distinguish tags from different regions.[5] With the detector tags added, the image I is embedded as a sequence of image region features $\boldsymbol{r}_{1:m}$ followed by a sequence of detector tag embeddings $\boldsymbol{d}_{1:l}$, i.e. $\boldsymbol{I} = [\boldsymbol{r}_{1:m}; \boldsymbol{d}_{1:l}]$. The tags are added during both pre-training and fine-tuning. Further, during pre-training, certain tag subwords are masked and the *tag reconstruction* loss L_I^{tag} supervises the model to predict the masked tags. The tags are predicted just as masked subwords are predicted in text modeling. The prediction softmax layer is shared between the tag and text subwords.

[4]Each tag corresponds to a region. A tag could be split into multiple subwords, so the total length of the tag subword sequence l is equal to or larger than the number of regions m.

[5]This design differs from that of Oscar [LYL20]. Oscar does not add the coordinate embeddings to tags to encourage the fusion of tag and visual representations.

The parameters involved in modeling tags include the token embedding, the coordinate embedding, and the subword softmax embedding. These embedding parameters are shared across modalities and encourage the model to project text, visual, and tag representations into the same space (see Section 3.4.2 for an example). This resembles the design in multi-lingual language models, which use shared BPE embeddings and softmax weights across languages [WD19].

3.3 Experiment

As the domain and quality of data may affect the model performance, the conventional practice in unsupervised learning is to use aligned corpora without providing alignments, allowing for controlled comparison with a supervised model. For example, unsupervised machine translation creates unaligned corpora by splitting up parallel corpora [LCD18] while unsupervised image captioning [GJC19] create unaligned corpus by shuffling images and captions from MSCOCO [CFL15]. Following prior work, we first conduct experiments by using Conceptual Captions (CC) [SDG18] as the source of images and text for both the supervised and unsupervised model. Later in Section 3.4.1, we show that our method is effective when the images and captions are collected independently and when no caption text is used.

U-VisualBERT. The model is pre-trained with shuffled captions and images. At each training step, we sample either a batch of images or a batch of text. Following VL-BERT [SZC19], we find it beneficial to include BookCorpus [ZKZ15], a general-domain text corpus, during pre-training. In sum, U-VisualBERT is trained on 3M images from CC, 3M captions from CC, and 2.5M text segments from BookCorpus[6].

[6]Our version of BookCorpus contains around 5M text segments with 64 words per segment. For computational reasons, we downsample the dataset such that during each epoch, the model observes only half of the text segments from BookCorpus. This downsampling is also done for the other VisualBERT variants.

S-VisualBERT. We introduce a Supervised VisualBERT (S-VisualBERT) trained with aligned data as introduced in Section 3.2.1. S-VisualBERT is pre-trained on 3M caption-image pairs from CC and 2.5M text segments from BookCorpus.

Compared models. Additionally, we list the performance of a Base VisualBERT that is initialized from BERT and does not undergo further pre-training. Previously reported supervised models that are trained on CC are also listed, including ViLBERT, VL-BERT, and UNITER. For UNITER, we include the version that is trained only on CC ($\text{UNITER}_{\text{cc}}$)[7]. Although their network architectures differ from ours and cannot be directly compared, they jointly paint the picture of the performance we should expect by pre-training on CC. Models developed before BERT are listed as Pre-BERT ([GJY19] for VQA, [SZZ19] for NLVR^2, [LCH18] for Flickr30K, and [YLS18] for RefCOCO+).

Setup. For all the VisualBERT variants introduced in the paper, we initialize them from BERT_{base} and pre-train for 10 epochs on their respective pre-training datasets with a batch size of 144. All models can be trained within 3 days on 4 V100s each with 16GB of memory. We use the Adam optimizer [KB15] with a linear-decayed learning-rate schedule [DCL19a] and a peak learning rate at 6×10^{-5}. We conduct evaluations by fine-tuning on four downstream tasks: Visual Question Answering (VQA 2.0) [GKS17b], Natural Language for Visual Reasoning (NLVR^2) [SZZ19], Image Retrieval (Flickr 30K) [PWC15], and Referring Expression (RefCOCO+) [YPY16a]. We use a Faster R-CNN pre-trained on the Visual Genome dataset to extract region features [AHB18]. For each task, we follow the recommended setting in previous works.

Results. Table 3.1 summarizes the results. For each model, we list the type and amount of data used during pre-training.[8] To control for randomness, we report the means and

[7]The results are from Appendix A.6 of [CLY20].

[8]For models initialized from BERT, we do not count the BERT pre-training data. VL-BERT uses both BookCorpus and Wikipedia during V&L pre-training. We estimate that the two corpora roughly have 5OM segments with 64 words per segment. With a different pre-processing style (e.g. longer segments), the

Model	Aligned	Unaligned		VQA	NLVR^2		Flickr30K			RefCOCO+		
		Image	Text	Test-Dev	Dev	Test-P	R@1	R@5	R@10	Dev	TestA	TestB
Pre-BERT	-	-	-	70.22	54.1	54.8	48.60	77.70	85.20	65.33	71.62	56.02
ViLBERT	3M	0	0	70.55	-	-	58.78	85.60	91.42	72.34	78.52	62.61
VL-BERT	3M	0	∼50M	71.16	-	-	-	-	-	71.60	77.72	60.99
UNITER$_{cc}$	3M	0	0	**71.22**	-	-	-	-	-	72.49	79.36	63.65
S-VisualBERT	3M	0	2.5M	70.87$_{\pm.02}$	**73.44**$_{\pm.51}$	**73.93**$_{\pm.51}$	**61.19**$_{\pm.06}$	**86.32**$_{\pm.12}$	**91.90**$_{\pm.02}$	**73.65**$_{\pm.11}$	**79.48**$_{\pm.36}$	**64.49**$_{\pm.22}$
Base	0	0	0	69.26	68.40	68.65	42.86	73.62	83.28	70.66	77.06	61.43
U-VisualBERT	0	3M	5.5M	**70.74**$_{\pm.06}$	**71.74**$_{\pm.24}$	**71.02**$_{\pm.47}$	**55.37**$_{\pm.49}$	**82.93**$_{\pm.07}$	**89.84**$_{\pm.21}$	**72.42**$_{\pm.06}$	**79.11**$_{\pm.08}$	**64.19**$_{\pm.54}$

Table 3.1: Evaluation results on four V&L benchmarks. Our unsupervised model trained with unaligned data (U-VisualBERT) achieves close performance with a supervised model trained with aligned data (S-VisualBERT). U-VisualBERT also rivals with several supervised models such as ViLBERT on most metrics.

standard deviations of U-VisualBERT and S-VisualBERT across three runs.

U-VisualBERT outperforms the Base model on all benchmarks, while only lagging behind S-VisualBERT slightly on VQA, NLVR^2, and RefCOCO+. U-VisualBERT even surpasses or rivals with some supervised models (e.g., ViLBERT on VQA and RefCOCO+, VL-BERT on RefCOCO+, and UNITER$_{cc}$ on RefCOCO+). This shows that a model through unsupervised pre-training can perform comparably with supervised models.

On Flickr30K Image Retrieval, the difference between U-VisualBERT and S-VisualBERT is more evident. The task focuses on identifying if an image and a text segment are coherent. S-VisualBERT is provided with explicit signals for such a task with the "text-image match" objective L_M during pre-training (Section 3.2.1). While U-VisualBERT is not provided with such explicit signals, it still performs better than the Base model. Further, if we were to remove the explicit signal (i.e. the "text-image match" objective) when pre-training on aligned data, S-VisualBERT without L_M achieves only 57.98 on R@1, much closer to U-VisualBERT.

number of segments may change.

Model	Text		VQA	NLVR^2		Flickr30K			RefCOCO+		
	Caption	General	Test-Dev	Dev	Test-P	R@1	R@5	R@10	Dev	TestA	TestB
Base	-	-	69.26	68.40	68.65	42.86	73.62	83.28	70.66	77.06	61.43
U-VisualBERT	CC	BC	**70.74**	71.74	71.02	55.37	**82.93**	89.84	72.42	79.11	64.19
U-VisualBERT$_{\text{SBU}}$	SBU	BC	70.70	**71.97**	**72.11**	**56.12**	82.82	**90.12**	**73.05**	**79.48**	64.19
U-VisualBERT$_{\text{NC}}$	-	BC	70.47	71.47	71.19	54.36	82.22	89.24	72.96	79.30	**64.25**

Table 3.2: Unsupervised pre-training is applicable when images and captions are collected independently (U-VisualBERT$_{\text{SBU}}$) or when no caption text is provided (U-VisualBERT$_{\text{NC}}$).

3.4 Analysis

In this section, we analyze the effect of the text data and the role of the detector tags.

3.4.1 The Effect of Text Data

The assumption behind unsupervised pre-training is that the detector tags should appear both in the images and text corpus, serving as the grounding anchor points. When the images and captions come from the same corpus, such an assumption clearly holds, and unsupervised pre-training works well (Section 3.3). However, we are curious if such an assumption still holds 1) if images and captions come from independently collected corpora (U-VisualBERT$_{\text{SBU}}$) and 2) if no caption text but general-domain text is provided (U-VisualBERT$_{\text{NC}}$).

The latter setting bears great practical value. Conceptually, collecting caption-style text could be as hard as collecting image-caption data as images and captions seldom appear separately. It is desirable to explore training V&L representations without caption-style text. Thus we experiment pre-training with general-domain text, which could be easier to collect.

U-VisualBERT$_{\text{SBU}}$. We use 3M images from CC and 1M captions from SBU captions [OKB11a]. To compensate for the different amounts of text between CC and SBU, we upsample the BookCorpus so that the amount of text data used by U-VisualBERT$_{\text{SBU}}$ is roughly the same as U-VisualBERT.

U-VisualBERT$_{\text{NC}}$. The model is trained on images from CC and text from BookCorpus, a general-domain corpus.

Results. Unsupervised pre-training is effective in both scenarios (Table 3.2). When pre-training images and text are collected independently, U-VisualBERT$_{\text{SBU}}$ achieves similar performance as U-VisualBERT, with the latter higher on VQA, and the former higher on the other three tasks.

When no caption text is used, the performance on NLVR^2 and RefCOCO+ remains unaffected while the performance on VQA and Flickr30K drops slightly, potentially because the language style of VQA and Flickr30K is similar to captions, benefiting U-VisualBERT. Such results are not surprising. In general-domain corpora like Wikipedia, grounded words take up a decent portion (>25%) [TB20]. Thus the tags appear in pre-training text corpora with a non-trivial frequency and U-VisualBERT$_{\text{NC}}$ learns from such signals. The above results suggest the applicability of unsupervised pre-training to many language-only and image-only datasets, which are easier to collect than image-caption datasets [TL18, SSS17].

3.4.2 The Detector Tags as Anchor Points

We study the effect of the detector tags in unsupervised and supervised pre-training, respectively.

W-VisualBERT$_{\text{NT}}$. U-VisualBERT$_{\text{NT}}$ observes no tags and only dense region features for image embeddings during pre-training and fine-tuning. For comparison, a base model without tags is introduced (Base$_{\text{NT}}$), which is initialized from BERT and does undergo fur-

Model	VQA	NLVR^2		Flickr30K			RefCOCO+		
	Test-Dev	Dev	Test-P	R@1	R@5	R@10	Dev	TestA	TestB
Base_{NT}	69.06	51.98	52.73	48.40	78.20	87.18	70.15	76.91	61.72
U-VisualBERT$_{\text{NT}}$	69.87	67.90	68.92	50.56	80.22	88.32	71.94	77.79	62.38
U-VisualBERT	70.74	71.74	71.02	55.37	82.93	89.84	72.42	79.11	64.19
S-VisualBERT$_{\text{NT}}$	70.49	72.56	73.53	60.26	85.58	91.64	72.70	77.93	62.99
S-VisualBERT	70.87	73.44	73.93	61.19	86.32	91.90	73.65	79.48	64.49
H-VisualBERT	$71.05_{\pm.02}$	$73.80_{\pm.26}$	$74.82_{\pm.25}$	$60.28_{\pm.60}$	$86.30_{\pm.35}$	$92.06_{\pm.28}$	$74.01_{\pm.25}$	$80.18_{\pm.23}$	$64.89_{\pm.24}$

Table 3.3: Detector tags show a larger impact in the unsupervised setting (U-VisualBERT$_{\text{NT}}$ vs. U-VisualBERT) than in the supervised setting (S-VisualBERT$_{\text{NT}}$ vs. S-VisualBERT). Semi-supervised pre-training (H-VisualBERT) shows marginal improvement over supervised pre-training (S-VisualBERT).

ther pre-training.

S-VisualBERT$_{\text{NT}}$. To study the effect of the detector tags when aligned data are present, we introduce S-VisualBERT$_{\text{NT}}$ which is trained on aligned data but observes no tags for image embeddings.

Result. We first find that even without tags, unsupervised pre-training benefits downstream tasks (Table 3.3). U-VisualBERT$_{\text{NT}}$ outperforms Base_{NT} on all metrics with a large margin. We attribute this to the (unaligned) contextual V&L representation learned through pre-training. This bears resemblance to the observation in multi-lingual language models that the shared vocabulary across languages (i.e. anchor points) is not necessary for cross-lingual transfer [CWL20].

Further, while the detector tags are beneficial for both supervised and unsupervised pre-training, the performance improvement is more evident for the latter. For example, performance difference on VQA between U-VisualBERT and U-VisualBERT$_{\text{NT}}$ is 0.95

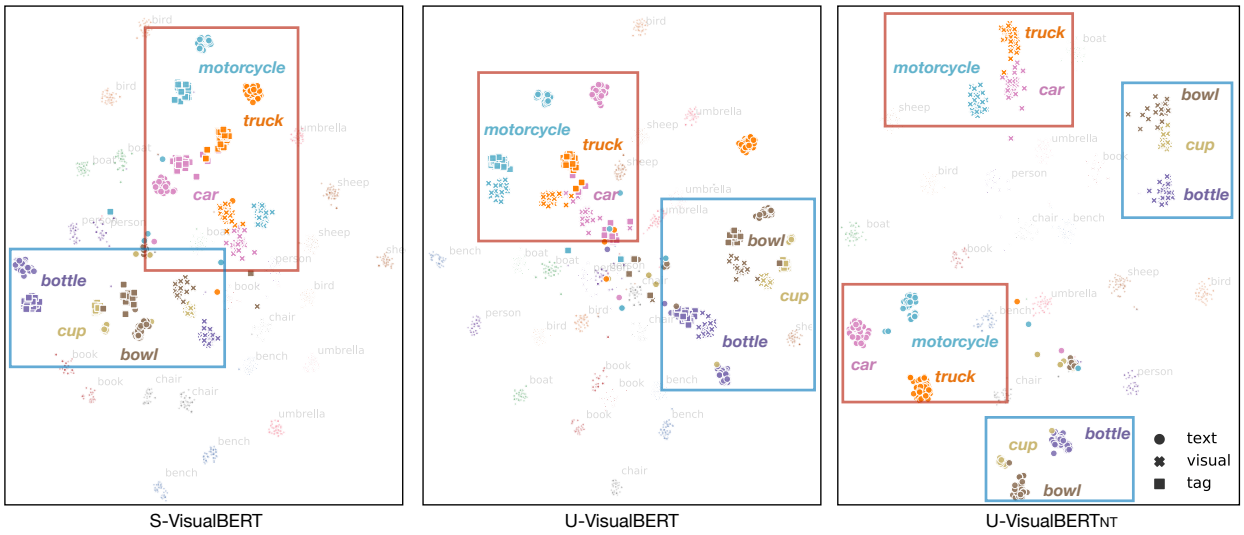

Figure 3.2: Visualization of the contextual representations of S-VisualBERT, U-VisualBERT, and U-VisualBERT$_{\text{NT}}$. The tags help to fuse text and visual representations for S-VisualBERT and U-VisualBERT. In U-VisualBERT$_{\text{NT}}$, common structures emerge in the text and visual representation spaces even though they are not aligned.

(70.82 vs. 69.87) while the difference between S-VisualBERT and S-VisualBERT$_{\text{NT}}$ is 0.41 (70.90 vs. 70.49). The results are expected. When aligned data are present, object tags serve as additional signals while in unsupervised pre-training, they serve as the only source from which grounding is learned.

Visualization. To gain a direct sense of how the detector tags help bridge the modalities, we visualize the contextual representation spaces of S-VisualBERT, U-VisualBERT, and U-VisualBERT$_{\text{NT}}$ in Figure 3.2. For each of the most frequent 15 object classes in the COCO dataset [CFL15], we randomly sample at most 50 instances and take the last-layer contextual representations of the words, the objects, and the tags (when available) and visualize them with t-SNE [MH08]. We highlight the representations of six selected classes.

Though trained without aligned data, U-VisualBERT can group text, tag, and visual representations by their semantic classes. Similar phenomena can be observed in S-VisualBERT.

U-VisualBERT$_{\text{NT}}$, lacking any signal to align the two spaces, does not show signs of such behaviour. In U-VisualBERT$_{\text{NT}}$, text and visual representations are almost completely separated (e.g., the two *disjoint red* rectangles in the figure on the right). However, some common structures emerge in both modalities. For instance, representations for "car", "truck", and "motorcycle", the three semantically-related classes, are close to each other, in both the textual and visual modality (the red rectangles); representations for "cup", "bottle", and "bowl" are close (the blue rectangles). This also holds for the other two models and resembles what is observed in [LYL20] and [IZF20].

3.4.3 Semi-Supervised Pre-Training

Unsupervised pre-training in itself has great practical and research value in many domains where aligned data is scarce. As a byproduct, we wonder if the approach could find its use in a semi-supervised setting, where we pre-train a model with both aligned data and unaligned data.

H-VisualBERT. We introduce a *hybrid* model that is trained on the 3M aligned data from Conceptual Captions (CC) and additional unaligned 1.7M images from Open Images (OI) [KRA20]. When a training sample comes from CC, we provide the model with a text-image pair, and when the training sample comes from OI, we provide only the image. We do not use any manually annotated visual labels provided in OI.

Result. We control for randomness by running H-VisualBERT for three times and report the means and stand deviations. We observe that H-VisualBERT brings consistent improvement upon S-VisualBERT on most tasks (Table 3.3) except Flickr30K[9]. This preliminary result is promising as the dataset scale in this experiment is relatively small (million-scale). Meanwhile, unannotated data generally could not improve upon a model trained

[9]On Flickr30K, the performance between H-VisualBERT and S-VisualBERT is similar, potentially because the "image-text match" objective is the dominant contributor and additional image-only data during pre-training have limited benefit (Section 3.3).

with annotated data significantly, unless drastically scaled up [HFW20]. We leave large-scale experiments to future work.

Part II

Language-Based Visual Perception

CHAPTER 4

Open-World Object Detection with Language Supervision

Visual recognition models are typically trained to predict a fixed set of pre-determined object categories, which limits their usability in real-world applications since additional labeled data are needed to generalize to new visual concepts and domains. CLIP [RKH21] shows that *image-level* visual representations can be learned effectively on large amounts of raw image-text pairs. Because the paired texts contain a boarder set of visual concepts than any pre-defined concept pool, the pre-trained CLIP model is so semantically rich that it can be easily transferred to downstream image classification and text-image retrieval tasks in zero-shot settings. However, to gain fine-grained understanding of images, as required by many tasks, such as object detection [RHG15, LGG17], segmentation [LSD15, CPK17], human pose estimation [XWW18, SXL19], scene understanding [KRA18, XZC17, HYH21], action recognition [JKF19], vision-language understanding [LDD19, TB19, CLY19, SZC19, LYY19, LDF19, ZPZ20b, LYL20, LT20, ZLH21], *object-level* visual representations are highly desired.

In this Chapter, we show that *phrase grounding*, which is a task of identifying the fine-grained correspondence between phrases in a sentence and objects (or regions) in an image, is an effective and scalable pre-training task to learn an object-level, language-aware, and semantic-rich visual representation, and propose Grounded Language-Image Pre-training (GLIP). Our approach unifies the phrase grounding and object detection tasks in that object detection can be cast as context-free phrase grounding while phrase ground-

ing can be viewed as a contextualized object detection task. We highlight our key contributions as follows.

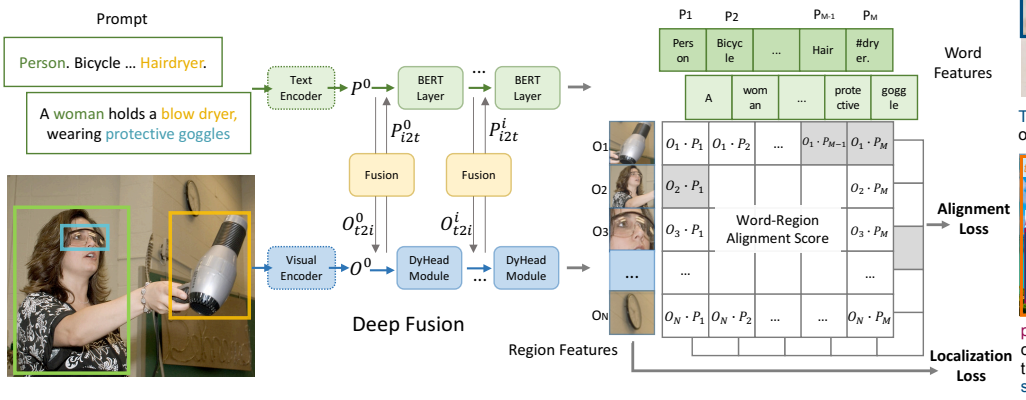

Figure 4.1: A unified framework for detection and grounding. Unlike a classical object detection model which predicts a categorical class for each detected object, we reformulate detection as a grounding task by aligning each region/box to phrases in a text prompt. GLIP jointly trains an image encoder and a language encoder to predict the correct pairings of regions and words. We further add the cross-modality deep fusion to early fuse information from two modalities and to learn a language-aware visual representation.

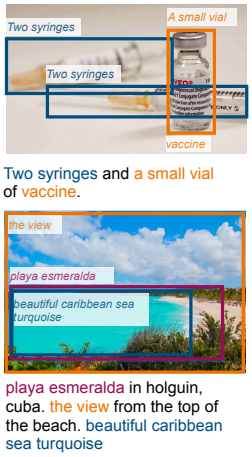

Figure 4.2: Grounding predictions from GLIP. GLIP can locate rare entities, phrases with attributes, and even abstract words.

Unifying detection and grounding by reformulating object detection as phrase grounding. The reformulation changes the input of a detection model: it takes as input not only an image but also a text prompt that describes *all* the candidate categories in the detection task[1]. For example, the text prompt for COCO object detection [LMB14a] is a text string that consists of 80 phrases, i.e., the 80 COCO object class names, joined by ". ", as shown in Figure 4.1 (Left). Any object detection model can be converted to a grounding model by replacing the object classification logits in its box classifier with the

[1]Different from typical phrase grounding tasks, phrases in the text prompt for an object detection task may not be present in the image.

word-region alignment scores, i.e., dot product of the region (or box) visual features and the token (or phrase) language features, as shown in Figure 4.1 (Right). The language features are computed using a language model, which gives the new detection (or grounding) model a dual-encoder structure. Different from CLIP that fuses vision and language only at the last dot product layer [RKH21], we show that deep cross-modality fusion applied by GLIP, as shown in Figure 4.1 (Middle), is crucial to learn high-quality language-aware visual representations and to achieve superior transfer learning performance. The unification of detection and grounding also allows us to pre-train using both types of data and benefits both tasks. On the detection side, the pool of visual concepts is significantly enriched thanks to the grounding data. On the grounding side, detection data introduce more bounding box annotations and help train a new SoTA phrase grounding model.

Scaling up visual concepts with massive image-text data. Given a good grounding model (teacher), we can augment GLIP pre-training data by automatically generating grounding boxes for massive image-text-paired data, in which noun phrases are detected by an NLP parser [BKL09]. Thus, we can pre-train our (student) GLIP-Large model (GLIP-L) on 27M grounding data, including 3M human-annotated fine-grained data and 24M web-crawled image-text pairs. For the 24M image-text pairs, there are 78.1M high-confidence (> 0.5) phrase-box pseudo annotations, with 58.4M unique noun phrases. We showcase two real examples of the generated boxes in Figure 4.2. The teacher model can accurately localize some arguably hard concepts, such as *syringes*, *vaccine*, *beautiful caribbean sea turquoise*, and even abstract words (*the view*). Training on such semantic-rich data delivers a semantic-rich student model. In contrast, prior work on scaling detection data simply cannot predict concepts out of the teacher models' pre-defined vocabulary [ZGL20]. In this study, we show that this simple strategy of scaling up grounding data is empirically effective, bringing large improvements to LVIS and 13 downstream detection tasks, especially on rare categories (Sections 4.3.2 and 4.4). When the pre-trained GLIP-L model is fine-tuned on COCO, it achieves 60.8 AP on COCO 2017val and 61.5 on test-dev,

surpassing the current public SoTA models [DCX21, XZH21] that scale up object detection data in various approaches.

Transfer learning with GLIP: one model for all. The grounding reformulation and semantic-rich pre-training facilitate domain transfer. GLIP can be transferred to various tasks with few or even no additional human annotations. When the GLIP-L model is directly evaluated on the COCO and LVIS datasets (without seeing any images in COCO during pre-training), it achieves 49.8 and 26.9 AP on COCO val2017 and LVIS val, respectively, surpassing many supervised baselines. When evaluated on 13 existing object detection datasets, spanning scenarios including fine-grained species detection, drone-view detection, and ego-centric detection, the setting which we term "Object Detection in the Wild" (ODinW) (Section 4.4.1), GLIP exhibits excellent data efficiency. For example, a zero-shot GLIP-L outperforms a 10-shot supervised baseline (Dynamic Head) pre-trained on Objects365 while a 1-shot GLIP-L rivals with a fully supervised Dynamic Head. Moreover, when task-specific annotations are available, instead of tuning the whole model, one could tune only the task-specific prompt embedding, while keeping the model parameters unchanged. Under such a prompt tuning setting (Section 4.4.2), one GLIP model can simultaneously perform well on all downstream tasks , reducing the fine-tuning and deployment cost.

4.1 Related Work

Standard object detection systems are trained to localize a fixed set of object classes predefined in crowd-labeled datasets, such as COCO [LMB14a], OpenImages (OI) [KRA18], Objects365 [SLZ19], and Visual Genome (VG) [KZG17], which contains no more than 2,000 object classes. Such human-annotated data are costly to scale up. GLIP presents an affordable solution by reformulating object detection as a phrase grounding (word-to-region matching) problem, and thus enables the use of grounding and massive image-text-paired data. Though our current implementation is built upon Dynamic Head (Dy-

Head) [DCX21], our unified formulation can be generalized to any object detection systems [RDG16, LGG17, DCX21, DLH16, RHG15, CPW19, CMS20, ZSL20, DCX21] for scalable Grounded Language-Image Pre-training.

Recently, there is a trend to develop vision-and-language approaches to visual recognition problems, where vision models are trained with free-form language supervision. For example, CLIP [RKH21] and ALIGN [JYX21] perform cross-modal contrastive learning on hundreds or thousands of millions of image-text pairs and can directly perform open-vocabulary image classification. By distilling the knowledge from the CLIP/ALIGN model into a two-stage detector, ViLD [GLK21] is proposed to advance zero-shot object detection. Alternatively, MDETR [KSL21] trains an end-to-end model on existing multi-modal datasets which have explicit alignment between phrases in text and objects in image. Our GLIP inherits the semantic-rich and language-aware property of this line of research, achieves SoTA object detection performance and significantly improves the transferability to downstream detection tasks.

This study focuses on domain transfer for object detection. The goal is to build one pre-trained model that seamlessly transfers to various tasks and domains, in a zero-shot or few-shot manner. Our setting differs from zero-shot detection [BSS18, RKP20, ZRH21, GLK21, ZWS20, RKB20], where some categories are defined as unseen/rare and not present in the training set. We expect GLIP to perform well on rare categories (Section 4.3.2) but we do not explicitly exclude any categories from our training set, because grounding data are so semantically rich that we expect them to cover many rare categories. This resembles the setting in open-vocabulary object detection [ZRH21], which expects raw image-text data to cover many rare categories. However, beyond performance on rare categories, which is the focus in prior work, we also consider the transfer cost in real-world scenarios, i.e., how to achieve the best performance with the least amount of data, training budget, and deployment cost (Section 4.4).

4.2 Grounded Language Image Pre-training

Conceptually, object detection and phrase grounding bear a great similarity. They both seek to localize objects and align them to semantic concepts. This synergy motivates us to cast the classical object detection task into a grounding problem and propose a unified formulation (Sec 4.2.1). We further propose to add deep fusion between image and text, making the detection model language-aware and thus a strong grounding model (Sec 4.2.2). With the reformulation and deep fusion, we can pre-train GLIP on scalable and semantic-rich grounding data (Sec 4.2.3).

4.2.1 Unified Formulation

Background: object detection. A typical detection model feeds an input image into a visual encoder Enc_I, with CNN [HCH16, TL19] or Transformer [LLC21, ZDY21, YLZ21] as backbone, and extracts region/box features O, as shown in Figure 4.1 (Bottom). Each region/box feature is fed into two prediction heads, i.e., a box classifier $\mathcal{C}$ and a box regressor $\mathcal{R}$, which are trained with the classification loss $\mathcal{L}_{\text{cls}}$ and the localization loss $\mathcal{L}_{\text{loc}}$, respectively:

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{loc}}. \tag{4.1}$$

In two-stage detectors, a separate region proposal network (RPN) with RPN loss $\mathcal{L}_{\text{rpn}}$ is used to distinguish foreground from background and refine anchors. Since $\mathcal{L}_{\text{rpn}}$ does not use semantic information of object classes, we merge it into the localization loss $\mathcal{L}_{\text{loc}}$. In one-stage detectors, localization loss $\mathcal{L}_{\text{loc}}$ may also contain the centerness loss [TSC19].

The box classifier $\mathcal{C}$ is typically implemented using a simple linear layer, and the classification loss $\mathcal{L}_{\text{cls}}$ can be written as:

$$O = \text{Enc}_I(\text{Img}),\ S_{\text{cls}} = OW^T,\ \mathcal{L}_{\text{cls}} = loss(S_{\text{cls}}; T). \tag{4.2}$$

Here[2], $O \in \mathbb{R}^{N \times d}$ are the object/region/box features of the input image, $W \in \mathbb{R}^{c \times d}$ is

[2] N is the number of region/box features, d is the visual feature hidden dimension, c is the number of

the weight matrix of the box classifier $\mathcal{C}$, $S_{\text{cls}} \in \mathbb{R}^{N \times c}$ are the output classification logits, $T \in \{0,1\}^{N \times c}$ is the target matching between regions and classes computed from the classical many-to-1 matching [RDG16, LGG17, DLH16, RHG15] or the bipartite Hungarian match [CMS20, ZSL20, DCX21]. $loss(S;T)$ is typically a cross-entropy loss for two-stage detectors and a focal loss [LGG17] for one-stage detectors.

Object detection as phrase grounding. Instead of classifying each region/box into c classes, we reformulate detection as a grounding task, by grounding/aligning each region to c phrases in a text prompt (see Figure 4.1). How to design a text prompt for a detection task? Given object classes [person, bicycle, car, ..., toothbrush], one simple way is

$$\text{Prompt} = \text{“Detect: person, bicycle, car, ... , toothbrush”},$$

in which each class name is a candidate phrase to be grounded. One could design better prompts, by providing more expressive descriptions of these classes and/or by exploiting the preference of a pre-trained language model. For example, when the pre-trained BERT model [DCL18] is used to initialize our language encoder Enc_L, the prompt “person. bicycle. car. ... . toothbrush” works better than the more human-friendly prompt described above. We will discuss the prompt design in Section 4.4.2.

In a grounding model, we compute the alignment scores S_{ground} between image regions and words in the prompt:

$$O = \text{Enc}_I(\text{Img}),\ P = \text{Enc}_L(\text{Prompt}),\ S_{\text{ground}} = OP^\top, \tag{4.3}$$

where $P \in \mathbb{R}^{M \times d}$ is the contextual word/token features from the language encoder and plays a similar role to the weight matrix W in Equation 4.2, as shown in Figure 4.1 (Right). The grounding model, consisting of both the image encoder Enc_I and the language encoder Enc_L, is trained end-to-end by minimizing the loss defined in Equation 4.1 & Equation 4.2,

---

object classes, and we ignore the bias in the box classifier for simplicity.

with a simple replacement of the classification logits S_{cls} in Equation 4.2 with the region-word aligment scores S_{ground} in Equation 4.3.

However, in Equation 4.2, we now have the logits $S_{\text{ground}} \in \mathbb{R}^{N \times M}$ and the target $T \in \{0,1\}^{N \times c}$. The number of (sub)-word tokens M is always larger than the number of phrases c in the text prompt due to four reasons: 1) some phrases contain multiple words, e.g., "traffic light"; 2) some single-word phrases are splitted into multiple (sub)-word tokens, e.g., "toothbrush" to "tooth#" and "#brush"; 3) some are the added tokens, such as "Detect:", ",", special tokens in language models, and 4) a [NoObj] token is added at the end of the tokenized sequence. When the $loss$ is a (focal) binary sigmoid loss (the $loss$ we use in Section 4.3 & 4.4), we expand the original target matrix $T \in \{0,1\}^{N \times c}$ to $T' \in \{0,1\}^{N \times M}$ by making all sub-words positive match if a phrase is a positive match and all added tokens negative match to all image features. With this change, the $loss(S_{\text{ground}}; T')$ remains the same. During inference, we average token probabilities as the phrase probability. [3]

Equivalence between detection and grounding. With the above reformulation, we can convert any detection model into a grounding model, and the two views, i.e., detection and grounding, are theoretically equivalent for both training and inference[4]. We also verify this empirically: the SoTA DyHead detector [DCX21] with Swin-Tiny backbone gives the same performance on COCO val2017 before and after our reformulation. With the reformulation, a pre-trained phrase grounding model can be directly applied to any object detection task, thanks to the free-form input of the language encoder. This makes it possible to transfer our GLIP model to arbitrary detection tasks in a zero-shot manner.

[3] When the $loss$ is a multi-class cross entropy (CE) loss, following MDETR [KSL21], all box proposals with no positive match are matched to the [NoObj] token. The $loss(S, T')$ becomes a multi-label multi-class CE loss, and we sum token probabilities as phrase probability during inference.

[4] The equivalence holds when all candidate categories can fit into one prompt. For certain detection tasks (e.g., Objects365 [SLZ19]), in practice, we can split the categories into multiple prompts during training and inference.

Related work. Our grounding formulation is inspired by MDETR [KSL21], and our grounding loss shares the same spirit of MDETR's fine-grained contrastive loss. We go further than MDETR by finding an effective approach to reformulate detection as grounding and a simple unified loss for both detection and grounding tasks. Our grounding model also resembles models for zero-shot detection [BSS18, RKP20, GLK21, ZWS20, RKB20]. The seminal work of Bansal et al. [BSS18] enables a detection model to conduct zero-shot detection, by using the pre-trained Glove word embedding [PSM14] as the phrase features $P \in \mathbb{R}^{c \times d}$, if written in the form of Equation 4.3. Recently, phrase features extracted from pre-trained deep language models are introduced in open-vocabulary detection [ZRH21]. Our GLIP model differs from zero-shot detection in that GLIP provides a unified view of detection and grounding, and thus enables the two crucial ingredients, i.e., language-aware deep fusion and scaling up with image-text-paired data, as to be described next.

4.2.2 Language-Aware Deep Fusion

In Equation 4.3, the image and text are encoded by separate encoders and only fused at the end to calculate the alignment scores. We call such models *late-fusion* models. In vision-language literature [LDD19, TB19, CLY19, SZC19, LYY19, LDF19, ZPZ20b, LYL20, KSL21], deep fusion of visual and language features is necessary to learn a performant phrase grounding model. Therefore, we introduce deep fusion between the image and language encoders, which fuses the image and text information in the last few encoding layers, as shown in Figure 4.1 (Middle). Concretely, when we use DyHead [DCX21] as the image encoder and BERT [DCL18] as the text encoder, the deep-fused encoder can be written as:

$$O^i_{\text{t2i}}, P^i_{\text{i2t}} = \text{X-MHA}(O^i, P^i), \quad i \in \{0, 1, .., L-1\} \tag{4.4}$$

$$O^{i+1} = \text{DyHeadModule}(O^i + O^i_{\text{t2i}}), \quad O = O^L, \tag{4.5}$$

$$P^{i+1} = \text{BERTLayer}(P^i + P^i_{\text{i2t}}), \quad P = P^L, \tag{4.6}$$

where L is the number of DyHeadModules in DyHead [DCX21], BERTLayer is newly-added BERT Layers on top of the pre-trained BERT, O^0 denote the visual features from the vision backbone, and P^0 denote the token features from the language backbone (BERT). The cross-modality communication is achieved by the cross-modality multi-head attention module (X-MHA) Equation 4.4, followed by the single modality fusion and updated in Equation 4.5 & Equation 4.6. Without added context vectors (O^i_{t2i} for vision modality and P^i_{i2t} for language modality), the model is reduced to a *late-fusion* model.

In the cross-modality multi-head attention module (X-MHA) Equation 4.4, each head computes the context vectors of one modality by attending to the other modality:

$$
\begin{aligned}
&O^{(q)} = OW^{(q,I)}, P^{(q)} = PW^{(q,L)}, Attn = O^{(q)}(P^{(q)})^\top/\sqrt{d}, \\
&P^{(v)} = PW^{(v,L)}, \;\; O_{\text{t2i}} = \text{SoftMax}(Attn)P^{(v)}W^{(out,I)}, \\
&O^{(v)} = OW^{(v,I)}, \;\; P_{\text{i2t}} = \text{SoftMax}(Attn^\top)O^{(v)}W^{(out,L)},
\end{aligned}
$$

where $\{W^{(\text{symbol},I)}, W^{(\text{symbol},L)} : \text{symbol} \in \{q, v, out\}\}$ are trainable parameters and play similar roles to those of query, value, and output linear layers in Multi-Head Self-Attention [VSP17], respectively.

The deep-fused encoder brings two benefits. 1) It improves the phrase grounding performance. 2) It makes the learned visual features language-aware, and thus the model's prediction is conditioned on the text prompt. This is crucial to achieve the goal of having one model serve all downstream detection tasks (shown in Section 4.4.2).

4.2.3 Pre-training with Scalable Semantic-Rich Data

Considerable efforts have been devoted to collecting detection data that are rich in semantics and large in quantity. However, human annotations have been proven costy and limited [KRA18, GDG19]. Prior work seeks to scale up in a self-training fashion [ZGL20]. They use a teacher (a pre-trained detector) to predict boxes from raw images and generate pseudo detection labels to train a student model. But the generated data are still limited in

terms of the size of the concept pool, as the teacher can only predict labels defined in the concept pool, constructed on the existing datasets. In contrast, our model can be trained on both detection and, more importantly, grounding data. We show that grounding data can provide rich semantics to facilitate localization and can be scaled up in a self-training fashion.

First, the gold grounding data cover a much larger vocabulary of visual concepts than existing detection data. The largest attempts at scaling up detection vocabulary still cover no more than 2,000 categories [KZG17, GDG19]. With grounding data, we expand the vocabulary to cover virtually any concepts that appear in the grounded captions. For example, Flickr30K [PWC15] contains 44,518 unique phrases while VG Caption [KZG17] contains 110,689 unique phrases, orders of magnitude larger than the vocabulary of detection data. We provide an empirical study in Section 4.3.4 to show that 0.8M gold grounding data brings a larger improvement on detecting rare categories than additional 2M detection data.

Further, instead of scaling up detection data, we show a promising route to obtaining semantically rich data: scaling up grounding data. We use a simple approach inspired by self-training. We first pre-train a *teacher* GLIP with gold (human-annotated) detection and grounding data. Then we use this teacher model to predict boxes for web-collected image-text data, with noun phrases detected by an NLP parser [BKL09]. Finally, a *student* model is trained with both the gold data and the generated pseudo grounding data. As shown in Figure 4.2, the teacher is capable of generating accurate boxes for semantically rich entities.

Why can the student model possibly outperform the teacher model? While discussions remain active in the self-training literature [ZGL20], in the context of visual grounding, we posit that the teacher model is utilizing the language context and language generalization ability to accurately ground concepts that it may not inherently know. For example, in Figure 4.2, the teacher may not directly recognize certain concepts such as *vaccine* and

Model	Backbone	Deep Fusion	Pre-Train Data		
			Detection	Grounding	Caption
GLIP-T (A)	Swin-T	✗	Objects365	-	-
GLIP-T (B)	Swin-T	✓	Objects365	-	-
GLIP-T (C)	Swin-T	✓	Objects365	GoldG	-
GLIP-T	Swin-T	✓	Objects365	GoldG	Cap4M
GLIP-L	Swin-L	✓	FourODs	GoldG	Cap24M

Table 4.1: A detailed list of GLIP model variants.

turquoise, if they are not present in gold data. However, the rich language context such as syntactic structures can provide strong guidance for the teacher model to perform an "educated guess". The model can localize *vaccine* if it can localize *a small vail*; it can localize *turquoise* if it can find *caribbean sea*. When we train the student model, the "educated guess" of the teacher model becomes a "supervised signal", enabling the student model to learn the concept of *vaccine* and *turquoise*.

4.3 Transfer to Established Benchmarks

After pre-training, GLIP can be applied to grounding and detection tasks with ease. We show strong direct domain transfer performance on three established benchmarks: 1) MS-COCO object detection (COCO) [LMB14a] containing 80 common object categories; 2) LVIS [GDG19] covering over 1000 objects categories; 3) Flickr30K [PWC15], for phrase grounding. We train 5 variants of GLIP (Table 4.1) to ablate its three core techniques: 1) unified grounding loss; 2) language-aware deep fusion; 3) and pre-training with both types of data.

GLIP-T (A) is based on a SoTA detection model, Dynamic Head [DCX21], with our word-region alignment loss replacing the classification loss. It is based on the Swin-Tiny backbone and pre-trained on O365 (Objects365 [SLZ19]), which contains 0.66M images and 365 categories. As discussed in Section 4.2.1, the model can be viewed as a strong classical zero-shot detection model [BSS18], relying purely on the language encoder to generalize

Model	Backbone	Pre-Train Data	Zero-Shot	Fine-Tune
			2017val	2017val / test-dev
Faster RCNN	RN50-FPN	-	-	40.2 / -
Faster RCNN	RN101-FPN	-	-	42.0 / -
DyHead-T [DCX21]	Swin-T	-	-	49.7 / -
DyHead-L [DCX21]	Swin-L	-	-	58.4 / 58.7
DyHead-L [DCX21]	Swin-L	O365,ImageNet21K	-	60.3 / 60.6
SoftTeacher [XZH21]	Swin-L	O365,SS-COCO	-	60.7 / 61.3
DyHead-T	Swin-T	O365	43.6	53.3 / -
GLIP-T (A)	Swin-T	O365	42.9	52.9 / -
GLIP-T (B)	Swin-T	O365	44.9	53.8 / -
GLIP-T (C)	Swin-T	O365,GoldG	**46.7**	55.1 / -
GLIP-T	Swin-T	O365,GoldG,Cap4M	46.3	54.9 / -
GLIP-T	Swin-T	O365,GoldG,CC3M,SBU	46.6	**55.2** / -
GLIP-L	Swin-L	FourODs,GoldG,Cap24M	**49.8**	**60.8** / 61.0
GLIP-L	Swin-L	FourODs,GoldG+,COCO	-	- / **61.5**

Table 4.2: Zero-shot domain transfer and fine-tuning on COCO. GLIP, without seeing any images from the COCO dataset, can achieve comparable or superior performance than prior supervised models (e.g. GLIP-T under Zero-Shot v.s. Faster RCNN under Fine-Tune). When fully fine-tuned on COCO, GLIP-L surpasses the SoTA performance.

to new concepts.

GLIP-T (B) is enhanced with language-aware deep fusion but pre-trained only on O365.

GLIP-T (C) is pre-trained on 1) O365 and 2) GoldG, 0.8M human-annotated gold grounding data curated by MDETR [KSL21], including Flickr30K, VG Caption [KZG17], and GQA [HM19b]. We have removed COCO images from the dataset. It is designed to verify the effectiveness of gold grounding data

GLIP-T is based on the Swin-Tiny backbone and pre-trained on the following data: 1)

Model	Backbone	MiniVal [KSL21]				Val v1.0			
		APr	APc	APf	AP	APr	APc	APf	AP
MDETR [KSL21]	RN101	20.9	24.9	24.3	24.2	-	-	-	-
MaskRCNN [KSL21]	RN101	26.3	34.0	33.9	33.3	-	-	-	-
Supervised-RFS [GDG19]	RN50	-	-	-	-	12.3	24.3	32.4	25.4
GLIP-T (A)	Swin-T	14.2	13.9	23.4	18.5	6.0	8.0	19.4	12.3
GLIP-T (B)	Swin-T	13.5	12.8	22.2	17.8	4.2	7.6	18.6	11.3
GLIP-T (C)	Swin-T	17.7	19.5	**31.0**	24.9	7.5	11.6	**26.1**	16.5
GLIP-T	Swin-T	**20.8**	**21.4**	**31.0**	**26.0**	**10.1**	**12.5**	25.5	**17.2**
GLIP-L	Swin-L	**28.2**	**34.3**	**41.5**	**37.3**	**17.1**	**23.3**	**35.4**	**26.9**

Table 4.3: Zero-shot domain transfer to LVIS. While using no LVIS data, GLIP-T/L outperforms strong supervised baselines (shown in gray). Grounding data (both gold and self-supervised) bring large improvements on APr.

O365, 2) GoldG as in GLIP-T (C), and 3) Cap4M, 4M image-text pairs collected from the web with boxes generated by GLIP-T (C). We also experiment with existing image caption datasets: CC (Conceptual Captions with 3M data) [SDG18] and SBU (with 1M data) [OKB11b]. We find that CC+SBU GLIP-T performs slightly better than Cap4M GLIP-T on COCO, but slightly worse on the other datasets. For simplicity, we report both versions on COCO but only the Cap4M model for the other tasks.

GLIP-L is based on Swin-Large and trained with: 1) FourODs (2.66M data), 4 detection datasets including Objects365, OpenImages [KDA17], Visual Genome (excluding COCO images) [KZG17], and ImageNetBoxes [KSH12]; 2) GoldG as in GLIP-T (C); and 3) CC12M+SBU, 24M image-text data collected from the web with generated boxes.

4.3.1 Zero-Shot and Supervised Transfer on COCO

We conduct experiments on MS-COCO to evaluate models' transfer ability to common categories. We evaluate under two settings: 1) zero-shot domain transfer, and 2) supervised

Row	Model	Data	Val			Test		
			R@1	R@5	R@10	R@1	R@5	R@10
1	MDETR-RN101	GoldG+	82.5	92.9	94.9	83.4	93.5	95.3
2	MDETR-ENB5	GoldG+	83.6	93.4	95.1	84.3	93.9	95.8
3	GLIP-T	GoldG	84.0	95.1	96.8	84.4	95.3	97.0
4		O365,GoldG	84.8	94.9	96.3	85.5	95.4	96.6
5		O365,GoldG,Cap4M	**85.7**	**95.4**	**96.9**	**85.7**	**95.8**	**97.2**
6	GLIP-L	FourODs,GoldG,Cap24M	**86.7**	**96.4**	**97.9**	**87.1**	**96.9**	**98.1**

Table 4.4: Phrase grounding performance on Flickr30K entities. GLIP-L outperforms previous SoTA by 2.8 points on test R@1.

transfer, where we fine-tune the pre-trained models using the standard setting. For the fine-tuning setting, we additionally test the performance of a GLIP-L model, where we include the COCO images in the pre-training data (the last row). Specifically, we add the full GoldG+ grounding data and COCO train2017 to the pre-training data. Note that part of COCO 2017val images are present in GoldG+ [KSL21]. Thus we only report the test-dev performance of this model.

We introduce an additional baseline: DyHead pre-trained on Objects365. We find that COCO 80 categories are fully covered in Objects365. Thus we can evaluate DyHead trained on Objects365 in a "zero-shot" way: during inference, instead of predicting from 365 classes, we restrict the model to predict only from the COCO 80 classes. We list standard COCO detection models for reference. We also list two state-of-the-art models pre-trained with extra data.

Results are present in Table 4.2. Overall, GLIP models achieve strong zero-shot and supervised performance. Zero-shot GLIP models rival or surpass well-established supervised models. The best GLIP-T achieves 46.7 AP, surpassing Faster RCNN; GLIP-L achieves 49.8 AP, surpassing DyHead-T. Under the supervised setting, the best GLIP-T brings 5.5 AP improvement upon the standard DyHead (55.2 v.s. 49.7). With the Swin-

Large backbone, GLIP-L surpasses the current SoTA on COCO, reaching 60.8 on 2017val and 61.5 on test-dev, without some bells and whistles in prior SoTA [XZH21] such as model EMA, mix-up, label smoothing, or soft-NMS.

We analyze the zero-shot performance of GLIP and find three contributing factors: close domain overlap between Objects365 and COCO, deep fusion, and grounding data. As Objects365 covers all categories in COCO, the O365 pre-trained DyHead-T shows strong performance, reaching 43.6 zero-shot AP; reformulating the model into a grounding model, we observe a slight performance drop (GLIP-T (A)); adding deep fusion boosts the performance by 2 AP (GLIP-T (B)); the largest contributor is the gold grounding data, with which GLIP-T (C) reaches a zero-shot AP of 46.7. While the addition of image-text data brings slight or no improvement on COCO (GLIP-T v.s. GLIP-T (C)), we find it essential in generalizing to rare classes, as we show in the LVIS experiments.

4.3.2 Zero-Shot Transfer on LVIS

We evaluate the model's ability to recognize diverse and rare objects on LVIS in a zero-shot setting. We report on MiniVal containing 5,000 images introduced in MDETR as well as the full validation set v1.0. Results are present in Table 4.3. We list three supervised models trained on the annotated data of LVIS. GLIP exhibits strong zero-shot performance on all the categories. GLIP-T is on par with supervised MDETR while GLIP-L outperforms Supervised-RFS by a large margin.

The benefit of using grounding data is evident. Gold grounding data brings a 4.2-point improvement on MiniVal APr (model C v.s. model B). Adding image-text data further improves performance by 3.1 points. We conclude that the semantic richness of grounding data significantly helps the model recognize rare objects.

4.3.3 Phrase Grounding on Flickr30K Entities

We evaluate the model's ability to ground entities in natural language on Flickr30K entities [PWC15]. Flickr30K is included in the gold grounding data so we directly evaluate the models after pre-training as in MDETR [KSL21]. We use the any-box-protocol specified in MDETR. Results are present in Table 4.4. We evaluate three versions of GLIP with different pre-training data. We list the performance of MDETR, the SoTA grounding model. MDETR is trained on GoldG+, containing 1.3M data (GoldG is a subset of GoldG+ excluding COCO images).

GLIP-T with GoldG (Row 3) achieves similar performance to MDETR with GoldG+, presumably due to the introduction of Swin Transformer, DyHead module, and deep fusion. More interestingly, the addition of detection data helps grounding (Row 4 v.s. 3), showing again the synergy between the two tasks and the effectiveness of our unified loss. Image-text data also helps (Row 5 v.s. 4). Lastly, scaling up (GLIP-L) can achieve 87.1 Recall@1, outperforming the previous SoTA by 2.8 points.

4.3.4 Analysis

In this section, we perform ablation study by pre-training GLIP-T on different data sources (Table 4.5). We answer two research questions. First, our approach assumes that the use of a detection dataset to bootstraps the model. One natural question is what the effect of this detection dataset is and whether grounding data still brings improvement when paired with different detection data. We pre-train GLIP with three different detection datasets (Row 1-6). We find that adding grounding data brings consistent improvement in all the three settings.

Second, we have shown the effectiveness of grounding data for both common and rare categories. One orthogonal direction is to scale up detection data by including more images and categories (Section 4.2.3). We intend to provide an empirical comparison be-

Row	Pre-Training Data	COCO	LVIS MiniVal			
		2017val	AP_r	AP_c	AP_f	AP
1	VG w/o COCO	26.9	4.9	10.4	23.2	16.1
2	+ GoldG	29.2	7.8	14.0	24.5	18.5
3	OpenImages	29.9	12.8	12.1	17.8	14.9
4	+ GoldG	33.6	15.2	16.9	24.5	20.4
5	O365	44.9	13.5	12.8	22.2	17.8
6	+GoldG	**46.7**	17.7	19.5	31.0	24.9
7	O365,GoldG,Cap4M	46.3	**20.8**	21.4	31.0	26.0
8	FourODs	46.3	15.0	**22.5**	**32.8**	**26.8**

Table 4.5: Effect of different detection data.

tween scaling up detection data and grounding data. We present GLIP trained with 4 public detection datasets (Row 8) as an extreme attempt at scaling up detection data with human annotations. The model is trained with 2.66M detection data in total, with an aligned vocabulary of over 1,500 categories. However, it still trails behind Row 6 on COCO and AP_r of LVIS, where Row 6 is trained with only 0.66M detection data and 0.8M gold grounding data. Adding image-text data further widens the gap on LVIS AP_r (20.8 versus 15.0). We conclude that grounding data are indeed more semantic-rich and a promising alternative to scaling up detection data.

4.4 Object Detection in the Wild

To evaluate GLIP's transferability to diverse real-world tasks, we curate an "Object Detection in the Wild" (ODinW) setting. We choose 13 public datasets on Roboflow[5], each requiring a different localization skill. Many of the datasets are designed with a specific ap-

[5] `https://public.roboflow.com/object-detection`

plication purpose to mimic real-world deployment scenarios. For example, EgoHands requires locating hands of a person; Pothole concerns detecting holes on the road; ThermalDogsandPeople involves identifying dogs and persons in infrared images.

We demonstrate that GLIP facilitates transfer to such diverse tasks on two dimensions. (1) GLIP brings great data efficiency, reaching the same performance with significantly less task-specific data than baselines (Section 4.4.1). (2) GLIP enables new domain transfer strategies: when adapting to a new task, we can simply change the text prompt and keep the entire grounding model unchanged. This greatly reduces deployment cost because it allows one centralized model to serve various downstream tasks (Section 4.4.2).

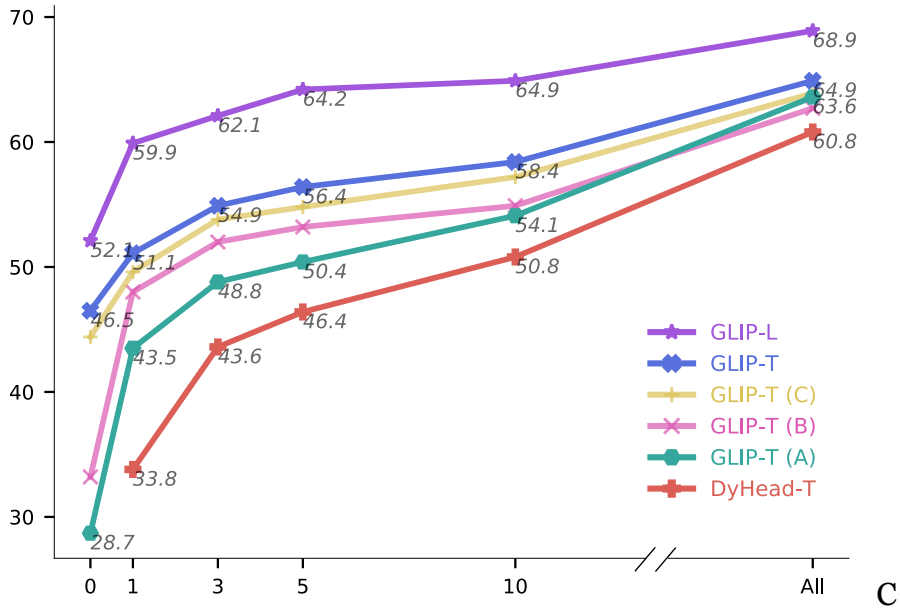

Figure 4.3: Data efficiency of models. X-axis is the amount of task-specific data, from zero-shot to all data. Y-axis is the average AP across 13 datasets. GLIP exhibits great data efficiency, while each of our proposed approach contributes to the data efficiency.

4.4.1 Data Efficiency

We vary the amount of task-specific annotated data, from zero-shot (no data provided), to X-shot (providing at least X examples per category [KLW19, YCX19, WRH19]), to using

all data in the training set. We fine-tune the models on the provided data and use the same hyper-parameters for all models. Each dataset comes with pre-specified category names. As GLIP is language-aware, we find it beneficial to re-write some pre-specified names with more descriptive language (see Section 4.4.2 for a discussion). We compare with the SoTA detector DyHead-T, pre-trained on Objects365. We test with the standard COCO-trained DyHead-T and find it giving similar performance. For simplicity, we report only the former. We also experiment with the scaled cosine similarity approach [WHD20] but find it slightly underperforming the vanilla approach so we report only the latter.

Results are shown in Figure 4.3. We find that unified grounding reformulation, deep fusion, grounding data, and model scale-up all contribute to the improved data efficiency (from the bottom red line (Dyhead-T) up to the upper purple line (GLIP-L)). As a result, GLIP exhibits transformative data efficiency. A zero-shot GLIP-T outperforms 5-shot DyHead-T while a one-shot GLIP-L is competitive with a fully supervised DyHead-T.

Examining the per-dataset performance, we find that grounding data brings significant improvement especially on certain tasks that test novel concepts. We plot the per-dataset performance on 5 selected datasets in Figure 4.4. On Pothole and EgoHands, which contain categories not present in Objects365, the models without grounding data (GLIP-T A&B) achieve an AP of less than 5, while the models with grounding data can achieve an AP of over 17 and 45, respectively.

4.4.2 One Model for All Tasks

As neural models become larger, how to reduce deployment cost has drawn an growing research interest. Recent work on language models [SRL20], image classification [ZLL21], and object detection [WHD20] has explored adapting a pre-trained model to a new domain but only changing the least amount of parameters. Such a setting is often denoted as linear probing [KJZ18], prompt tuning [ZLL21], or efficient task adapters [GGZ21]. The ultimate goal is to have a single model to simultaneously serve various tasks, and each task

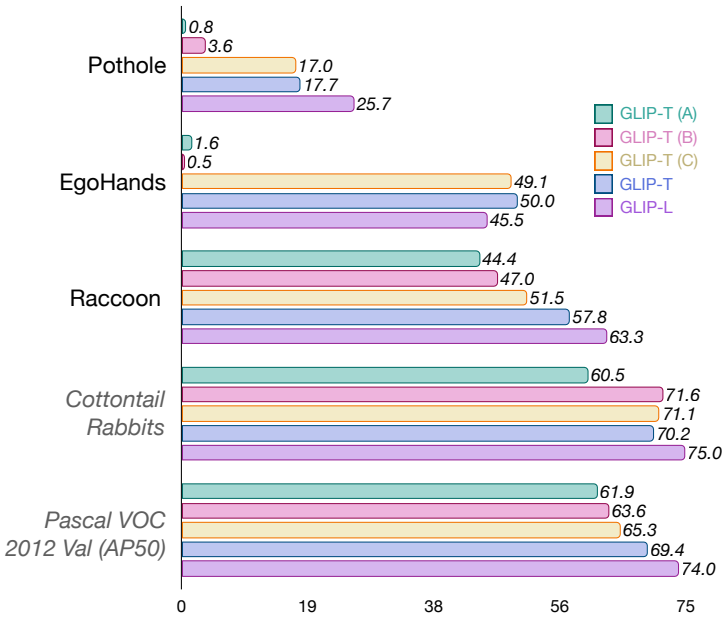

Figure 4.4: Per dataset zero-shot performance. The first 3 datasets contain novel categories not present in the Objects365 vocabulary while the last 2 datasets' categories are covered by Obj365 data. Grounding data bring significant benefit to novel categories.

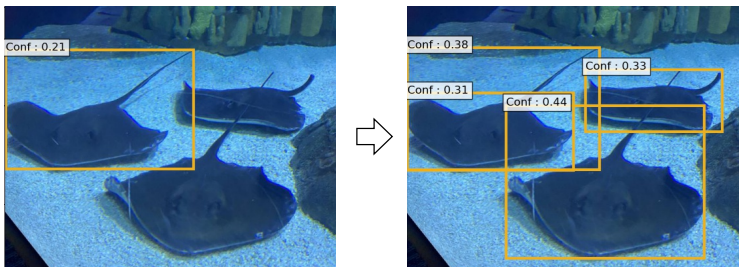

Figure 4.5: A manual prompt tuning example from the Aquarium dataset in ODinW. Given an expressive prompt ("flat and round"), zero-shot GLIP can detect the novel entity "stingray" better. For simplicity, we show only the predictions for the class "stingray".

adds only a small amount of task-specific parameters or no parameters to the pre-trained model. This reduces training and storage cost. In this section, we evaluate models against the metric of deployment efficiency.

Manual prompt tuning. As GLIP performs language-aware localization, i.e., the output of GLIP is heavily conditioned on the language input, we propose an efficient way for GLIP to do task transfer: for any novel categories, the user can use expressive descriptions in the text prompt, adding attributes or language context, to inject domain knowledge and help GLIP transfer. For example, on the left hand side of Figure 4.5, the model fails to localize all occurrences of the novel entity "stingray". However, by adding the attributes to the prompt, i.e., "flat and round", the model successfully localizes all occurrences of stringrays. With this simple prompt change, we improve the AP50 on stingray from 4.6 to 9.7. This resembles the prompt design technique in GPT-3 [BMR20] and is practically appealing, as it requires no annotated data or model re-training.

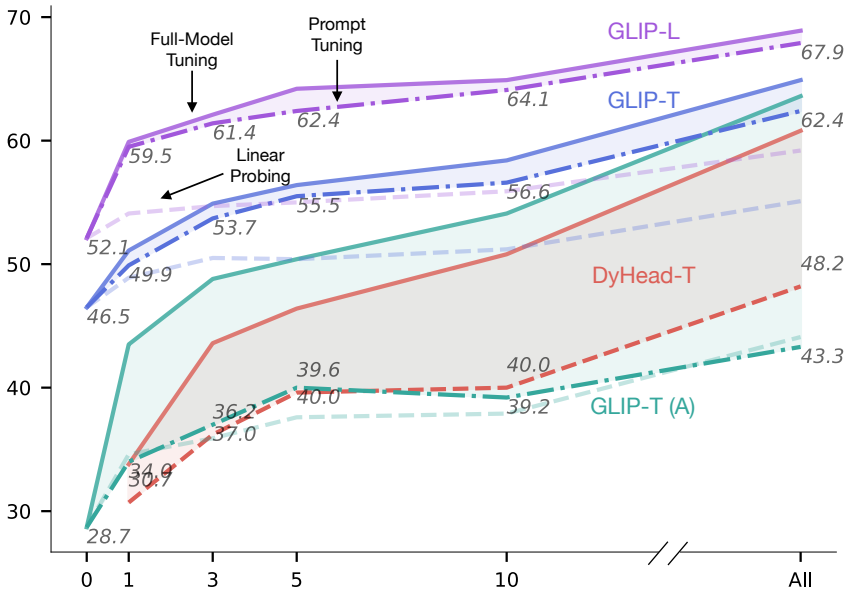

Figure 4.6: Effectiveness of prompt tuning. Solid lines are full-model tuning performance; dashed lines are prompt/linear probing performance. By only tuning the prompt embeddings, GLIP-T and GLIP-L can achieve performance close to full-model tuning, allowing for efficient deployment.

Prompt tuning. We further consider the setting where we have access to task-specific training data but wish to tune the least amount of parameters for easy deployment. For

classical detection models, Wang et al. [WHD20] report the effectiveness of "linear probing" (i.e., train only the box regression and classification head). GLIP can also be "linear probed", where we only fine-tune the box head and a projection layer between the region and prompt embeddings. Because of the language-aware deep fusion, GLIP supports a more powerful yet still efficient transfer strategy: prompt tuning [SRL20, LAC21]. For GLIP, as each detection task has only one language prompt (e.g., the prompt for Pothole could be "Detect pothole." for all images), we first get prompt embeddings P^0 from the language backbone, then discard the language backbone and only fine-tune P^0 as the task-specific input (Section 4.2.2).

We evaluate the models' performance under three settings (Figure 4.6): linear probing, prompt tuning (only applicable for GLIP), and full-model tuning. For DyHead-T, prompt tuning is not applicable as the traditional object detection model cannot accept language input; the gap between linear probing and full-model tuning is large. GLIP-T (A) has no language-aware deep fusion; thus prompt tuning and linear tuning achieve similar performance and lag significantly behind full-model tuning. However, for GLIP-T and GLIP-L, prompt tuning almost matches the full-tuning results, without changing any of the grounding model parameters. Interestingly, as the model and data size grow larger, the gap between full-model tuning and prompt tuning becomes smaller (GLIP-L v.s. GLIP-T), echoing the findings in NLP literature [LYF21]. In Table **??**, we report the prompt tuning performance with full data of GLIP-L (along with the prompt-tuning performance on COCO) and the model can achieve high performance on 14 datasets with one set of parameter weights.

4.5 Conclusion

GLIP unifies the object detection and phrase grounding tasks to learn an object-level, language-aware, and semantic-rich visual representation. After pre-training, GLIP showed promising results on zero-shot and fine-tuning settings on well-established benchmarks

and 13 downstream tasks. We leave a detailed study of how GLIP scales with text-image data size to future work.

CHAPTER 5

Learning with Rich Language Descriptions

Training visual recognition models to classify or detect objects with a fixed set of predefined categories has been the convention for a long time. However, models trained using this approach often encounter difficulties when adapting to unfamiliar concepts and domains. There has been a paradigm shift towards *training visual recognition models with language supervision*, using a contrastive objective on a large amount of image-text data containing a diverse range of visual concepts. These models can then be transferred to downstream tasks via language queries. GLIP is one such example and can perform object detection by querying the model with "Detect: person, cat, dog $\cdots$".

Early applications of these models typically utilize simple language queries that consist of object names. However, language queries can convey much richer and more comprehensive information, such as object attributes, shapes, textures, and relations. These pieces of information can be especially useful for identifying novel visual concepts that do not appear in the training corpus or specifying specific needs. For example, the concept of "mallet" can be described as "a kind of tool, wooden handle with a round head" (Figure 5.1, bottom-left). This decomposes object recognition into recognizing fine-grained details (such as attributes, sub-parts, shapes, etc.) and aligning them to the descriptions. Several studies [LZZ22, SLH22, MV23] have explored the idea of guiding language-based recognition models using such descriptive prompts. However, few existing models take complex queries into account during training. As a result, current models often struggle with recognizing intricate object names, attributes, and relations described in natural

language sentences [YBK23, TJB22] (see examples in Figure 5.1).

Figure 5.1: Comparison between our model (DesCo-GLIP) and the baseline (GLIP [LZZ22]). Each image is paired with a positive query for target object and a negative query for confusable object. A successful model should locate the target object and ignore the confusable object in the image based on fine-grained specifications for shapes, subparts, relations, etc. We highlight the descriptions that match and do not match to the queried object in blue and red, respectively. Results show that our model can successfully localize the target object and suppresses the negative query even for the difficult cases when the object name is not in the query or the object.

In this study, we develop a vision-language model capable of leveraging description-rich language queries to perform object detection. This work aligns with the recent surge of interest in instruction/prompt-aware vision-language models (see a discussion in Sec-

tion 5.1). Our goal is to equip VLMs with the ability to comprehend complex language input describing visual concepts, similar to the capability of large language models. We specifically study instruction/prompts in the form of descriptive queries. We focus on object detection, as it requires fine-grained recognition and is more challenging than image-level tasks. However, our method can be generalized to other vision tasks such as classification and segmentation [ZDY23].

We identify two major challenges that prevent existing models from efficiently utilizing rich language descriptions: (1) Fine-grained descriptions are rare in image-caption data used for training current VLMs[1]. This resembles the reporting bias phenomenon [PAR21]: when writing captions for images, humans tend to directly mention the entities rather than give a detailed description. (2) Even when provided with data rich in descriptions, models often lack the incentive to leverage these descriptions effectively. The main training objective is to align positive phrases with relevant regions while suppressing negative phrases. However, if positive/negative phrases can be distinguished without descriptions, the training mechanism fails to incentivize the model to use the provided description. For example, a positive phrase like "A toy bear holding a *mallet*, which has a wooden handle with a round head," and a negative phrase like "A toy bear holding an *ax*, which has a long handle and a sharp blade," can be differentiated based solely on the words *mallet* and *ax*. This issue resembles the issue discovered by [YBK23], where vision-language models ignore word order and treat a query as a "bag-of-words" due to insufficient incentives from the contrastive objective. In addition, current models suffer severe hallucination when given natural language queries (in contrast to "template-like" queries) due to shortcuts introduced in training query formulation. This can be seen in the bottom-right picture of Figure 5.1, where GLIP hallucinates and predicts multiple wrong boxes for "microphone" while "microphone" does not appear in the image.

[1]We count the region-label data used by models like GLIP as image-caption data because the labels are converted into captions through templates.

Based on the observations, we present a **Des**cription-**Co**nditioned (DesCo) paradigm of learning object recognition models from language descriptions based on two synergistic ideas:

(1) **Generating descriptions with large language models.** Instead of learning from raw image-caption pairs, we use large language models as a world knowledge base and generate detailed descriptions based on the original caption. We prompt GPT-3 [BMR20] with "What features should object detection models focus on for {an entity in the caption}?". This serves as a scalable approach to transfer the image-caption data into image-description data.

(2) **Context-sensitive query construction.** As discussed, even if we provide descriptions during pre-training, models can still ignore the language context. Our solution is to create a "context-sensitive query", which is a set of positive and negative phrases that can only be distinguished by reading the descriptions (Figure 5.2). We explore two strategies: 1) constructing "Winograd-like" [Hir81, TJB22] queries by using large language models to generate confusable object descriptions and captions and 2) generalizing the original grounding task to allow full-negative queries, reducing hallucination.

We apply our approach to fine-tune two state-of-the-art language-conditioned object detection models GLIP [LZZ22] and FIBER [DKG22]. We use the same raw training data as the baselines but convert the data into description-rich queries. We evaluate our methods in two settings. (1) Zero-shot generalization to novel categories (LVIS [GDG19]), where we use GPT-3 to generate descriptions given class names. DesCo-GLIP (Tiny) improves upon GLIP (Tiny) by 10.0 APr, even outperforming the larger GLIP (Large); DesCo-FIBER improves upon FIBER by 9.1 APr. (2) Zero-shot generalization to natural descriptions given by humans (OmniLabel [SSD23]). DesCo-GLIP and DesCo-FIBER improve upon the baselines by 4.5 AP and 3.6 AP, setting a new state-of-the-art performance level.

5.1 Related work

Language-based visual recognition models. Visual recognition models are typically trained to make predictions based on a fixed set of classes [KH09, DDS09b, LMB14b, SLZ19, MCL14, ZZP17]. The trained models are hard to generalize to open-domain settings. Recent studies have developed visual recognition models that take into account language queries, i.e. language-based recognition. This line of research can be traced back to early work of generalizing image classification [SGM13] and object detection [BSS18] models with word embeddings. Recently, CLIP [RKH21] reformulates image classification as image-text matching and pre-trains models on large-scale image-caption pairs to learn transferrable representations. They demonstrate strong zero-shot performance on various classification tasks. Recent work has applied the technique to fine-grained recognition tasks, such as object detection [KSL21, GLK22, LZZ22, ZYZ22, ZLL22, CKR22, MGS22, DKG22, LZR23], and segmentation [LWB22, GGC22, HKL22, XDL22, ZLZ23, LZS23]. These works either use pure image-text data as supervision [XDL22], or reformulate labeled data into image-text data [LWB22], or pseudo labels image-text data with fine-grained labels [LZZ22]. Orthogonal to architecture design or scaling-up, which is the focus of many prior studies, this study points out that the vanilla way of using image-text data is insufficient and studies how to train these models to take more flexible and informative language queries.

Vision-language models with complex prompts. As vision recognition models become language-aware and language models become vision-aware [TMC21, ZCS23, LGY23], there is a growing interest in studying whether these models can take complex language prompts, such as task instructions (e.g., GPV [GKK22, KCG22], SEEM [ZYZ23], Vision-LLM [WCC23]), descriptions [LLL22], or even dialogues (e.g., LLaVa [LLW23]). We study specifically descriptive prompts, which are especially useful for generalizing to novel categories and customized detection needs; a model that can understand descriptive prompts

can also serve as the backbone for supporting aforementioned other types of prompts. Similar to our work, K-LITE [SLH22] proposes to retrieve knowledge for a visual concept using external knowledge bases, then use the enriched concepts for image classification or object detection; similar techniques have also been proposed by [MV23, YPZ23, YWZ23]. DetCLIP [YHW22] builds a large-scale concept dictionary, based on which they provide definitions from WordNet. Different from these studies, our methods show that simply presenting the descriptions at training or inference time is not enough; we propose a simple technique to force models to focus on the provided descriptions (Section 5.2.2.2). Our work relates to a line of work that seek to reveal and fix failure patterns of image-text matching models (e.g., CLIP) by creating hard negative examples [TJB22, YBK23, DAH23, RKK23].

5.2 Approach

In this section, we first briefly introduce language-based object detection models, then illustrate the details of our proposed approach.

5.2.1 Background

We give an introduction to *language-based* object detection models [KSL21, LZZ22, DKG22], which take a language query and an image as inputs, and predict bounding boxes and their alignment to phrases in the language query. In the following, we use GLIP as an example.

Baseline: Grounded Language-Image Pre-training (GLIP). At the center of these approaches is "reformulating any task-specific fixed-vocab classification problem as a task-agnostic open-vocabulary vision-language matching problem" [ZZH22]. The best example is CLIP which reformulates image classification as image-text matching. Similarly, GLIP unifies training data into a *grounding* format: (I, Q, B, T). I is the image; Q is the text query; $B \in R^{N\times 4}$ is the bounding boxes; $T \in \{0,1\}^{N\times K}$ indicates the ground-truth alignment label between the N bounding boxes and K tokens in the query. The key is how to

formulate the *query* with data from two kinds of sources:

- *Detection data*. For object detection data such as Objects365 [SLZ19], the query is the concatenation as a list of object classes, such as "Detect: person. bicycle. car. $\cdots$, toothbrush". Note that negative object classes are included in the query; this makes such query-based detection models similar to classical detection models.
- *Grounding data*. Typically, Q is an image caption, containing entities that can are aligned to annotated object regions [PWC15]. For example, "A toy bear holding a mallet" is the caption; "toy bear" and "mallet" are the "groundable" entities. For densely annotated grounding data (multiple captions for one image) [KZG17], we can concatenate multiple captions into a longer query. Image-caption data (without annotated boxes) can be transferred into grounding data via pseudo labeling with a grounding model [LZZ22].

Given I and Q, we compute the alignment scores S_{ground} between image regions and words in the query:

$$O, L = \text{Enc}(I, Q),\ S_{\text{ground}} = OL^{\top}, \mathcal{L} = \ loss(S_{\text{ground}}, T) + \mathcal{L}_{\text{loc}}$$

where $L \in \mathbb{R}^{K \times d}$ is the contextual token features and $O \in \mathbb{R}^{N' \times d}$ are the regions features. Enc is a vision and language encoder that takes both image and text as inputs and fuses their representations. The training loss contains the region-word matching loss and a localization loss $\mathcal{L}_{\text{loc}}$)as in conventional object detection models.

Inference with language query. At inference time, the model can be used to locate entities/class names appearing in the query. One could simply provide a list of candidate object names (as in the detection data training format). GLIP also shows the promise of using descriptions for generalization to novel concepts; however, we show that while GLIP can be influenced by the description, it does not always take the details in the description into account.

5.2.2 Learning with Language Descriptions

To train object recognition models that fully utilize language descriptions, we propose to generate descriptions with large language models and construct context-sensitive queries during training. The following subsections provide further details.

5.2.2.1 Description Generation With Large Language Models

Fine-grained descriptions could be scarce in image-caption data due to reporting bias. While this problem can be alleviated by scaling up the pre-training corpus, we show that large language models [DCL18, BMR20] can be used to effectively generate the descriptions and thus enrich our training corpus.

We leverage a large language model to transform a query Q into a description-rich query $\text{Llm}(Q)$. In this work, we only focus on generating descriptions for entities mentioned in the original query. We construct a vocabulary consisting of 10K entities appearing frequently in the pre-training corpus. For each entity, we prompt a large language model: `what features should object detection models focus on for {entity}?` We find that large language models give high-quality responses (see examples in Figure 5.1 and Figure 5.4).

5.2.2.2 Context-Sensitive Query Construction

Can we simply add the description-rich data to the pre-training corpus? An intuitive idea is to append the description to the original entity to form a training prompt (e.g., "Detect: mallet. bear$\cdots$" $\rightarrow$ "Detect: mallet, a kind of tool, wooden handle $\cdots$ bear, a kind of animal, $\cdots$"). However, we find that models naively trained with these prompts still do not exhibit "context-sensitivity", i.e., they make predictions solely based on the entity names while ignoring other contexts (see Section 5.3.1 for quantitative analysis). As a result, we observe no evident benefit in incorporating descriptions during inference (Table 5.3). In the following, we elaborate on why the model learns to ignore the descriptions and propose

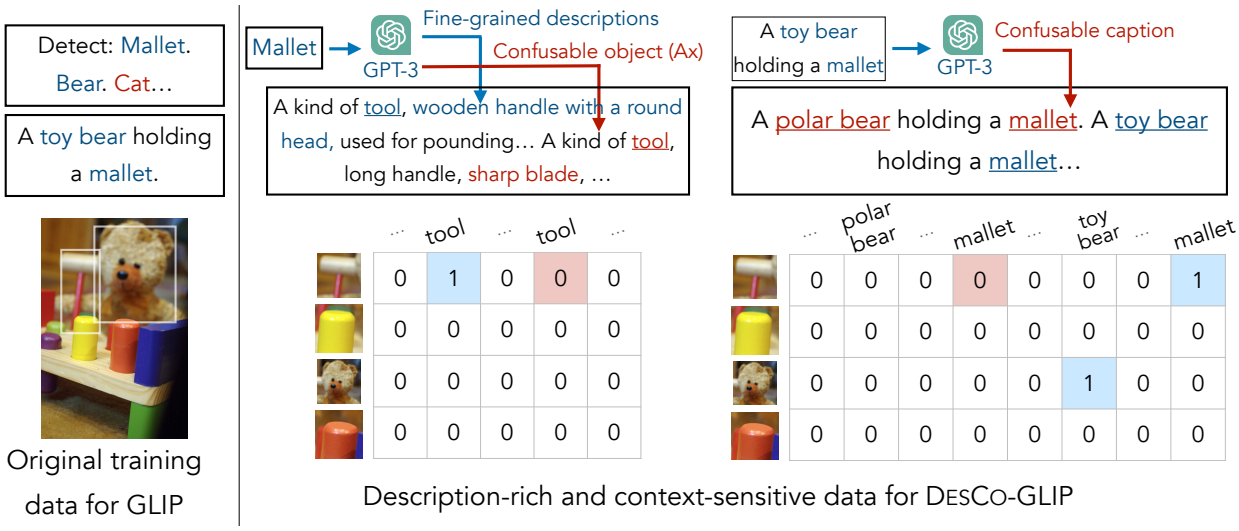

Figure 5.2: Given the original training data of GLIP, we transform it to be description-rich and context-sensitive by: 1) generating descriptions for entities and composing each of them with confusable object descriptions; 2) generating negative captions. We visualize the gold alignment labels (ground truth) between tokens and regions for the new data. Notably, words such as *tools* are assigned both positive (blue block) and negative (red block) labels in alignment with the corresponding object depending on the context of the query. As such, the model requires understanding the description in order to make the correct prediction.

two solutions.

Model learn statistical shortcuts. We first illustrate that without careful design, the model could learn two statistical shortcuts that make them insensitive to descriptive queries.

(1) *Entity shortcut.* The model is trained to align the entities in the query to image regions (this includes predicting "no-alignment" for entities not appearing in the image). Intuitively, if the alignment can be predicted without relying on the context information in the query, then the model is not incentivized to focus on the context information. Figure 5.2 illustrates this issue with an example. The left side shows the training data of GLIP,

where the top query ("Detect: Mallet. Bear. Cat...") comes from detection data and the bottom query ("A toy bear holding a mallet.") comes from grounding data. The problem with such queries is that they can be grounded by only focusing on the entity names and ignoring the context. We denote the gold alignment label of regions as T, the entities in the query as E, and the non-entity part (context) of the query as C. The mutual information $I(T; C|E, I)$ between C and T given E and the image I would be low. That is, the non-entity parts of the queries do not affect the label of the region. Training models on such data will not encourage the model to focus on the descriptions as they provide no additional information. This is similar to the "memorization overfit" issue observed in [YTZck]: the model can simply choose to "memorize" the alignment between the entities and regions.

(2) *Format shortcut (hallucination).* Popularized by GLIP [LZZ22], a line of work adopts a unified view of phrase grounding and object detection: detection can be seen as language-context-free grounding while grounding can be seen as language-context-dependent detection. However, this unification is still *imperfect*: phrase grounding (or referring expression [YPY16b]) traditionally only concerns locating entities in a caption that always exist in the image; thus the model learns to always treat the natural-language queries (in contrast to the template-like queries) as positive and tries to ground every entity mentioned in the sentence. This will result in failure examples as illustrated in the bottom-right picture of Figure 5.1. Such "hallucination" can be commonly seen on models trained on language grounding data [KSL21]; these models are almost incapable of distinguishing positive and negative "natural-language-like" queries.

Constructing context-sensitive queries. This motivates our solution of creating queries that are hard to solve without context (Figure 5.2 and Figure 5.3). We explore two strategies in this study.

(1) We construct training queries similar to the Winograd format. For example, when training on detection data, instead of "Detect: mallet, a kind of tool, $\cdots$", we remove the

entity name "mallet" from the query and sample another description of a "confusable" entity that is also a kind of tool. Pairing the descriptions of the two "confusable" entities creates a strong supervision signal (the middle example in Figure 5.2): the alignment label (0 or 1) of the word "tool" now depends on its context. The confusable entities are obtained by prompting the large language models as well. Similarly, for training on grounding data, we prompt language models to generate confusable (hard negative) captions that differ from the original captions only by a few words (the example on the right in Figure 5.2). Note that the label of the word "mallet" is now affected by the context: the *first* "mallet" is assigned 0 as the caption ("A polar bear holding a mallet") is negative. Mixing in such hard negative captions encourages the model to focus on the context surrounding the entities, such as relations to other entities. To make the confusable caption generation process scalable for image-caption data, we first perform in-context inference and prompt GPT-3 to generate around 50K negative captions based on positive captions; then we distill this knowledge to the open-sourced LLaMA-7B [TLI23] model that is instruction-finetuned using low-rank adaptation[2] [HWA22] and perform inference on large-scale image-caption data.

(2) To resolve the hallucination issue, we generalize the original grounding task: instead of always feeding the model a query that contains at least one description/caption matching the image, we allow the query contain only negative descriptions/captions (Figure 5.3). Thus, the model cannot blindly ground all entities mentioned in the query; implicitly, it needs to perform image-text matching [RKH21] as well as phrase grounding. This was partly done in GLIP, but the query still contains at least one positive entity.

Overview. In Figure 5.3, we summarize the overall data construction algorithm. **Algorithm 1**: $B \in R^{N\times 4}$ are the bounding boxes of an image; E are M positive objects (entities) that appear in the image; V are the descriptions of all candidate objects in a pre-defined

[2]https://github.com/tloen/alpaca-lora

Algorithm 1 Generating Queries for *Detection* Data

Input: B (boxes), T (alignment matrix), E (positive entities), V (vocabulary)

1: $Q \leftarrow \emptyset$
2: **for** $i \leftarrow 1$ to M **do**
3: $q, Q^- \leftarrow$ LLM(prompt, E_i)
4: **if** random() < p_{drop} **then**
5: $q, Q^- \leftarrow$ DropEntity(q, Q^-)
6: $Q \leftarrow Q \cup \{q\} \cup Q^-$
7: $Q \leftarrow Q \cup$ RandSample(V)
8: $q^* \leftarrow$ SubSampleConcat(Q)
9: $q^*, T, B \leftarrow$ LabelAssign(q^*, T, E, B)

Algorithm 2 Generating Queries for *Grounding* Data

Input: B (boxes), T (alignment matrix), E (positive entities), V (vocabulary), C (caption)

1: **if** random() $< p_{\text{des}}$ **then**
2: $Q, T, B \leftarrow$ Algorithm1(B, T, E, V)
3: **else**
4: $Q^- \leftarrow$ LLM(prompt$_{\text{neg}}$, C)
5: $Q \leftarrow \{C\} \cup Q^-$
6: $q^* \leftarrow$ SubSampleConcat(Q)
7: $q^*, T, B \leftarrow$ LabelAssign(q^*, T, C, B)

Figure 5.3: Algorithms for generating queries from detection data and grounding data.

vocabulary; $T \in \{0,1\}^{N \times M}$ denotes the gold alignment between boxes and entities. We first prompt LLM to generate descriptions for the positive entities and propose confusable entities and descriptions (Line 3). The original entities are removed from the descriptions with $p_{\text{drop}} = 0.5$ (DropEntity, Line 4-5). Random negative descriptions from the vocabulary are added to the candicate description set (Line 7). We then randomly subsample the descriptions and concatenate them to form a final query q^*; this is because the total length of all the candidate descriptions is too large (Line 8). Boxes and the mapping relations between boxes and tokens are accordingly adjusted (Line 9). **Algorithm 2**: C is the original caption and E are M positive phrases we extracted from the caption. The last two lines of

both algorithms are important: after *SubSampleConcat*, it is very likely that some positive sub-queries are dropped from Q; then *LabelAssign* would drop boxes that are mapped to the dropped sub-queries. The output B could end with fewer boxes or even no boxes. This is different from the strategy in GLIP or traditional object detection training recipe, where we strive to keep all boxes provided.

5.3 Experiment

In this section, we first investigate whether current models (GLIP) can utilize language descriptions out-of-the-box; then we show that our method allows the model to utilize language descriptions and improves performance on LVIS and OmniLabel significantly.

5.3.1 Do Current Models Utilize Language Descriptions?

As a proof of concept, we first show the GLIP struggles to utilize language descriptions out of the box and analyze the failure patterns.

Model	ΔBox	ΔConf	AP
GLIP [LZZ22]	0.291	0.05	4.7
DesCo-GLIP	**0.381**	**0.11**	**12.4**

Table 5.1: GLIP is insensitive to context changes compared to DesCo-GLIP.

GLIP does not effectively utilize language descriptions. We make an attempt at using descriptions to transfer GLIP to LVIS [GDG19], which contains over 1,200 classes. The process is similar to that of [MV23]. For each category, we prompt a large language model (GPT-3) to give details descriptions (as in Section 5.2) We append the description to the original class name to form a new query. An example of the queries can be seen shown in Figure 5.1 (bottom row). Directly appending the description to the object name at inference time only degrades the performance: GLIP-T achieves 20.8 AP on rare categories while appending the descriptions makes the performance drop to 12.2 AP. This is likely due to model hallucination on natural-language-like queries.

GLIP is insensitive to context changes. Examining the model predictions, we find that the model not only does not utilize language descriptions; it ignores the descriptions and tends to only focus on entity names, as we hypothesized. To quantitatively verify the phenomenon, we introduce a *context-sensitivity* test, inspired by the WinoGround [TJB22] benchmark. For each image, we provide the model with a positive query q^+ describing an object that appears in the image and a negative query q^- describing a confusable object. The original object names are removed from the query. An example of the test is shown in Figure 5.1 (bottom left), where the model is challenged to distinguish "mallet" and "ax". q^+ and q^- describe objects from the same general category (e.g., both are "a kind of tool") while differing in other aspects, similar to the Winograd test.

Intuitively, if a model can effectively utilize the descriptions, it should exhibit two properties: 1) it should give higher alignment scores to entities in q^+ compared to q^-; 2) even if the model cannot "guess" the hidden entity, at least, the model predictions should change drastically when given two different descriptions. We thus introduce two metrics. 1) AP, which measures how accurate the model's predictions are. 2) ΔBox and ΔConf, which are the differences between the model's predictions for q^+ and q^-. ΔBox measures the changes in box coordinates while ΔConf measures the changes in alignment scores of boxes.

We find that the baseline model not only cannot identify the correct description (low AP); but it effectively ignores the language context (low ΔBox and ΔConf) (Table 5.1). On average, the confidence of the predicted boxes changes only 0.05 between q^+ and q^-. One could see the examples in Figure 5.1. GLIP models make almost identical predictions for two different queries. Such insensitivity to language context makes it infeasible and unreliable to use descriptions to control model predictions.

5.3.2 Setup

In this section, we apply our approach to two vision-language object detection models GLIP [LZZ22] and FIBER [DKG22].

Model	Backbone	LVIS MiniVal [KSL21]				OmniLabel [SSD23]			
		APr	APc	APf	AP	AP	APc	APd	APd-P
MDETR [KSL21]	RN101	20.9	24.9	24.3	24.2	-	-	4.7	9.1
MaskRCNN [KSL21]	RN101	26.3	34.0	33.9	33.3	-	-	-	-
RegionCLIP [ZYZ22]	ResNet-50	-	-	-	-	2.7	2.7	2.6	3.2
Detic [ZGJ22]	Swin-B	-	-	-	-	8.0	15.6	5.4	8.0
K-LITE [SLH22]	Swin-T	14.8	18.6	24.8	21.3	-	-	-	-
GroundingDINO-T [LZR23]	Swin-T	18.1	23.3	32.7	27.4	-	-	-	-
GroundingDINO-L [LZR23]	Swin-L	22.2	30.7	38.8	33.9	-	-	-	-
GLIP-L [LZZ22]	Swin-L	28.2	34.3	41.5	37.3	25.8	32.9	21.2	33.2
GLIP-T [LZZ22]	Swin-T	20.8	21.4	31.0	26.0	19.3	23.6	16.4	25.8
DesCo-GLIP	Swin-T	**30.8**	**30.5**	**39.0**	**34.6**	**23.8**	**27.4**	**21.0**	**30.4**
FIBER-B [DKG22]	Swin-B	25.7	29.0	39.5	33.8	25.7	30.3	22.3	34.8
DesCo-FIBER	Swin-B	**34.8**	**35.5**	**43.9**	**39.5**	**29.3**	**31.6**	**27.3**	**37.7**

Table 5.2: Zero-shot transfer to LVIS and OmniLabel. Numbers that are grayed out are supervised models. DesCo-GLIP and GLIP-T are directly comparable; DesCo-FIBER and FIBER-B are directly comparable; the rest are listed for reference and not directly comparable.

Models. The visual backbon of GLIP and FIBER is Swin Transformer [LLC21] and the text backbones are BERT [DCL18] for GLIP and RoBERTa [LOG19] for FIBER. Both models use Dynamic Head [DCX21] as the detection architecture. Built upon the two models, we introduce two model variants: **DesCo-GLIP** and **DesCo-FIBER**.

Datasets. Following GLIP [LZZ22], we train the models on 1) O365 (Objects365 [SLZ19]), consisting of 0.66M images and 365 categories; 2) GoldG that is curated by MDETR [KSL21] and contains 0.8M human-annotated images sourced from Flickr30k [PWC15], Visual Genome [KZG17], and GQA [HM19a]; 3) CC3M [SDG18]: the web-scraped Conceptual Captions dataset with the same pseudo-boxes used by GLIP. We down-sample CC3M to

around 1.4M images to save training costs, based on whether high-confidence boxes exist in the image. As illustrated in Section 5.2, we convert the text caption of each instance into a detailed language description to construct description-rich data.

To evaluate how well the models generalize to novel concepts, we perform a zero-shot evaluation on the LVIS [GDG19] and OmniLabel [SSD23] datasets. LVIS is a popular dataset that has over 1,200 object categories with a challenging long tail of rare objects; OmniLabel is recently proposed and focuses on object detection with diverse and complex object descriptions in a naturally open-vocabulary setting. For evaluation on LVIS, for each category, we append the GPT-3 generated description to the category name; we group several descriptions into one query to save inference time. For OmniLabel evaluation, we follow the original evaluation protocol without modifications. We also verify that the models still possess the ability to perform the conventional detection and grounding tasks as GLIP and FIBER, on COCO [LMB14b] and Flickr30K [PWC15].

Implementation details. We initialize DesCo-GLIP from the GLIP-T checkpoint and DesCo-FIBER from the FIBER-B checkpoint. We fine-tune the models on both the original data and the new description-rich data. For DesCo-GLIP, we fine-tune with a batch size of 16 and a learning rate of 5×10^{-5} for 300K steps; for DesCo-FIBER, we fine-tune with a batch size of 8 and a learning rate of 1×10^{-5} for 200K steps. Experiments can be replicated with 8 GPUs each with 32GB memories.

5.3.3 Zero-shot Transfer to LVIS and OmniLabel

LVIS. Our method shows notable improvements over the baselines on the LVIS MiniVal dataset (Table 5.2). The improvement is particularly prominent for rare object categories (APr), with an increase of 10.0 for GLIP and 9.1 for FIBER.

OmniLabel. Our method also shows improvements over baselines on the OmniLabel dataset (Table 5.2). OmniLabel assesses model performance using plain categories (APc),

Row	Model	LVIS MiniVal [KSL21]				OmniLabel COCO [SSD23]				Context Sensitivity		
		APr	APc	APf	AP	AP	APc	APd	APd-P	ΔBox	ΔConf	AP
1	GLIP-T	20.8	21.4	31.0	26.0	18.7	45.7	11.7	31.2	0.291	0.05	4.7
2	+ Description w/ Entity Name	20.5	23.9	35.5	29.2	23.6	47.4	14.7	36.0	0.293	0.06	5.7
3	+ Description w/o Entity Name	25.6	25.9	**35.9**	30.7	24.0	46.8	16.0	**37.0**	**0.382**	0.10	**10.7**
4	+ Description w/o Name + Neg Cap	**26.5**	**27.1**	35.8	**31.3**	**24.7**	**48.2**	**16.6**	36.2	0.381	**0.10**	10.5

Table 5.3: Ablation study. Directly appending the description does not improve performance on rare categories (Row 1 v.s. Row 2, LVIS APr). Constructing context-sensitive queries is crucial.

free-form descriptions (APd), and positive descriptions (APd-P). Because our models are trained with description data, they naturally excel in supporting such queries, leading to substantial increases in APd and APd-P compared to the baselines. Specifically, DesCo-GLIP achieves a notable improvement of +4.6, while DesCo-FIBER achieves an even more impressive improvement of +5.0. Furthermore, our model's effectiveness extends beyond free-form descriptions to plain categories as well, as illustrated in the table. This highlights the robustness of our method across different evaluation settings and its ability to achieve improvements in various types of queries. Our method wins the 1st place in the Omnilabel challenge 2023 on all three tracks.

5.3.4 Ablation Study

In this section, all ablation models are initialized from GLIP-T and trained for 100K steps.

Directly appending descriptions. We examine the impact of directly adding language descriptions to text queries, without incorporating context-sensitive query construction. The results are presented in Row 2 of Table 5.3. The performance on rare categories (APr) sees no improvement but decreases. To further evaluate the sensitivity of the model to contextual changes, we conduct the same context sensitivity analysis as the one de-

scribed in Section 5.3.1. The context sensitivity of the model almost remains unchanged (Row 1-2): ΔBox changes only 0.002 and ΔConf changes only 0.01. The results indicate that the model remains as insensitive to context changes as the baseline model. This suggests that the model struggles to accurately interpret and effectively utilize the provided language descriptions when context-sensitive query construction is removed.

Dropping the entity name. As in Section 5.2.2.2, we hypothesize that randomly removing the entity name can force the models to concentrate on the contextual information. Remarkably, the results presented in Table 5.3 (Row 2-3) demonstrate that this simple and intuitive approach proves to be highly effective. It significantly enhances the model's contextual sensitivity while concurrently improving object detection performance.

GPT	APr	APc	APf	AP
ada	19.9	23.2	33.7	28.0
babbage	24.2	26.7	36.5	31.3
curie	24.7	28.4	38.2	32.8
davinci	30.8	30.5	39.0	34.6

Table 5.4: Detection performance improves when language model size grows.

Negative captions. We also investigate the effectiveness of using language models to generate hard negative captions. As shown in Row 4 of Table 5.3, including negative captions can improve the model detection performance across datasets while preserving its robust contextual comprehension. These results indicate that this technique effectively enhances the model's ability to grasp the subtleties embedded in the given language descriptions.

Language description quality. We explore the effect of the language model size on detection performance. We evaluate the pre-trained DesCo-GLIP on LVIS with descriptions generated from the GPT families[3]. As presented in Table 5.4, higher-quality language models improve object detection performance. This finding highlights the importance of employing strong language models, as they possess the ability to embed valuable visual information through extensive pre-training. We showcase two examples in Figure 5.4.

[3]https://platform.openai.com/docs/models

Figure 5.4: Detection performance of DesCo-GLIP improves when given better descriptions. GPT-Curie is a smaller model than GPT-Davinci; it gives less accurate descriptions for objects.

5.4 Conclusion and Limitations

In this study, we introduced a new paradigm of learning object detection models from language supervision. We show that large language models can be used to generate rich descriptions and the necessity to construct context-sensitive queries. We hope that our method sheds light on empowering vision models with the ability to accept flexible language queries.

While we greatly improve the models' ability to understand flexible language queries, our method has several limitations that can be addressed in future work. 1) We use a large language model to automatically generate the descriptions, which inevitably introduces noise as not all generated descriptions are accurate or beneficial for representation learning. Future work could consider automatically selecting useful descriptions sampled from the language model, similar to [YPZ23]. 2) The format of the descriptions we explored is still limited (e.g., "{entity}, a kind of {type}, {list of simple features}"); it might be useful to consider more diverse descriptions by prompting the language model with more diverse

prompts. 3) Similar to large language models, querying our model also requires a certain amount of prompt engineering. Future work could explore how to make the model more robust to different kinds of queries.

CHAPTER 6

Conclusion and Future Directions

In general, my research studies the alignment between vision and language, how to build representations that encodes such alignment, and how such representations can be useful for downstream tasks. While great progress has been made, there are many unanswered research questions. Below I list some interesting future directions.

Multimodal large language models. Since the earliest pre-trained vision-language models such as VisualBERT, vision-language models have evolved into Multimodal Larga Language Models (MLLMs) [LLW24]. They typically consist of a vision encoder to embed images into grid features, which are fed into a Large Language Model for processing and reasoning alongside a text input. They need to pre-determine how many tokens an image is worth, and set a fixed number for all images. Finding a flexible number that adaptively strikes a balance between efficiency and performance is difficult. In our recent work [HDL24], we introduce a simple way to train a single MLLM that supports adaptively changing the number of visual tokens at inference time. Future work could consider how to adapt the approach for processing videos, where visual tokens will dominate the computational cost.

Visual grounding beyond objects. In prior work, we have largely focused on recognizing objects in images, as they are basic building blocks of the visual world. As the models improve, it is time to extend our approaches beyond objects. I envision teaching the model to understand a broader range of visual commonsense concepts, such as actions, relations, social interactions, using approaches outlined in this thesis. Language

will be the form of supervision in all these cases, and approaches we developed in DesCo will useful for making sure the model does not take shortcuts.

Unification between perception and generation. We have witnessed great progress on the development of generative models [RBL22]. An interesting question is whether generative models should also have learned perception as well. It seems natural to assume that to generate an image, the model must possess the ability to understand it as well. There has been growing research in this direction. For example, MAE [HCX22] finds that by pre-training a model to predict masked pixels, we can obtain a good visual backbone. How to seamlessly unify perceptions models, which are usually pre-trained with a contrastive loss or language modeling loss, and visual generation models, which are typically modeled using diffusion, remains an active and promising research direction.

References

[AHB18] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[BKL09] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009.

[BMR20] 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." *arXiv preprint arXiv:2005.14165*, 2020.

[BSS18] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. "Zero-shot object detection." In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 384–400, 2018.

[CFL15] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. "Microsoft COCO captions: Data collection and evaluation server." *arXiv preprint arXiv:1504.00325*, 2015.

[CGC20a] Jize Cao, Zhe Gan, Yu Cheng, Licheng Yu, Yen-Chun Chen, and Jingjing Liu. "Behind the Scene: Revealing the Secrets of Pre-trained Vision-and-Language Models." *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[CGC20b] Liqun Chen, Zhe Gan, Yu Cheng, Linjie Li, Lawrence Carin, and Jingjing Liu.

"Graph optimal transport for cross-domain alignment." *Proceedings of the International Conference on Machine Learning (ICML)*, 2020.

[CKL19] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. "What Does BERT Look At? An Analysis of BERT's Attention." *BlackboxNLP*, 2019.

[CKN20] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. "A simple framework for contrastive learning of visual representations." *arXiv preprint arXiv:2002.05709*, 2020.

[CKR22] Zhaowei Cai, Gukyeong Kwon, Avinash Ravichandran, Erhan Bas, Zhuowen Tu, Rahul Bhotika, and Stefano Soatto. "X-DETR: A versatile architecture for instance-wise vision-language tasks." In *ECCV*, 2022.

[CLY19] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. "UNITER: Learning universal image-text representations." *arXiv preprint arXiv:1909.11740*, 2019.

[CLY20] 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." ECCV, 2020.

[CMS20] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. "End-to-end object detection with transformers." In *European Conference on Computer Vision*, pp. 213–229. Springer, 2020.

[CPK17] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs." *IEEE transactions on pattern analysis and machine intelligence*, **40**(4):834–848, 2017.

[CPW19] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. "Hybrid task cascade for instance segmentation." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4974–4983, 2019.

[CSD21] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. "Conceptual 12M: Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts." *arXiv preprint arXiv:2102.08981*, 2021.

[CWL20] Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. "Emerging Cross-lingual Structure in Pretrained Language Models." In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6022–6034, 2020.

[DAH23] 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.

[DBK20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.", 2020.

[DCL18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805*, 2018.

[DCL19a] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." In *Proceedings of the 2019 Conference of the North American Chapter of the*

Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186, 2019.

[DCL19b] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Multilingual BERT Readme Document." https://github.com/google-research/bert/blob/master/multilingual.md, 2019.

[DCX21] Xiyang Dai, Yinpeng Chen, Bin Xiao, Dongdong Chen, Mengchen Liu, Lu Yuan, and Lei Zhang. "Dynamic Head: Unifying Object Detection Heads with Attentions." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7373–7382, 2021.

[DDS09a] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. "ImageNet: A Large-Scale Hierarchical Image Database." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2009.

[DDS09b] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "ImageNet: A large-scale hierarchical image database." In *CVPR*, 2009.

[DGE15] Carl Doersch, Abhinav Gupta, and Alexei A Efros. "Unsupervised visual representation learning by context prediction." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[DKG22] Zi-Yi Dou, Aishwarya Kamath, Zhe Gan, Pengchuan Zhang, Jianfeng Wang, Linjie Li, Zicheng Liu, Ce Liu, Yann LeCun, Nanyun Peng, Jianfeng Gao, and Lijuan Wang. "Coarse-to-Fine Vision-Language Pre-training with Fusion in the Backbone." In *NeurIPS*, 2022.

[DLH16] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. "R-fcn: Object detection via region-based fully convolutional networks." In *Advances in neural information processing systems*, pp. 379–387, 2016.

[DM08] Marie-Catherine De Marneffe and Christopher D Manning. "Stanford typed dependencies manual." Technical report, 2008.

[DM17] Timothy Dozat and Christopher D Manning. "Deep biaffine attention for neural dependency parsing." *ICLR*, 2017.

[FML19] Yang Feng, Lin Ma, Wei Liu, and Jiebo Luo. "Unsupervised Image Captioning." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[FSB20] Zihao Fu, Bei Shi, Lidong Bing, and Wai Lam. "Unsupervised KB-to-Text Generation with Auxiliary Triple Extraction using Dual Learning." In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pp. 258–268, 2020.

[GCL20] Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. "Large-Scale Adversarial Training for Vision-and-Language Representation Learning." *arXiv preprint arXiv:2006.06195*, 2020.

[GDG19] Agrim Gupta, Piotr Dollar, and Ross Girshick. "LVIS: A dataset for large vocabulary instance segmentation." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5356–5364, 2019.

[GGC22] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. "Open-vocabulary image segmentation." In *ECCV*, 2022.

[GGN18] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. "AllenNLP: A Deep Semantic Natural Language Processing Platform." In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pp. 1–6, 2018.

[GGZ21] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. "CLIP-Adapter: Better Vision-Language Models with Feature Adapters." *arXiv preprint arXiv:2110.04544*, 2021.

[GJC19] Jiuxiang Gu, Shafiq Joty, Jianfei Cai, Handong Zhao, Xu Yang, and Gang Wang. "Unpaired image captioning via scene graph alignments." In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2019.

[GJY19] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[GKK22] Tanmay Gupta, Amita Kamath, Aniruddha Kembhavi, and Derek Hoiem. "Towards general purpose vision systems: An end-to-end task-agnostic vision-language architecture." In *CVPR*, 2022.

[GKS17a] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. "Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering." In *CVPR*, 2017.

[GKS17b] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. "Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[GLK21] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. "Zero-Shot Detection via Vision and Language Knowledge Distillation." *arXiv preprint arXiv:2104.13921*, 2021.

[GLK22] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. "Open-vocabulary Object Detection via Vision and Language Knowledge Distillation." In *ICLR*, 2022.

[GRG18] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. "Detectron." `https://github.com/facebookresearch/detectron`, 2018.

[HCH16] Ji He, Jianshu Chen, Xiaodong He, Jianfeng Gao, Lihong Li, Li Deng, and Mari Ostendorf. "Deep Reinforcement Learning with a Natural Language Action Space." In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1621–1630, 2016.

[HCX22] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. "Masked autoencoders are scalable vision learners." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

[HDL24] Wenbo Hu, Zi-Yi Dou, Liunian Harold Li, Amita Kamath, Nanyun Peng, and Kai-Wei Chang. "Matryoshka Query Transformer for Large Vision-Language Models." *arXiv preprint arXiv:2405.19315*, 2024.

[HFW20] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. "Momentum contrast for unsupervised visual representation learning." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[Hir81] Graeme Hirst. "Anaphora in natural language understanding: a survey." 1981.

[HKL22] Dat Huynh, Jason Kuen, Zhe Lin, Jiuxiang Gu, and Ehsan Elhamifar. "Open-vocabulary instance segmentation via robust cross-modal pseudo-labeling." In *CVPR*, 2022.

[HM19a] Drew A. Hudson and Christopher D. Manning. "GQA: a new dataset for compositional question answering over real-world images." In *CVPR*, 2019.

[HM19b] Drew A Hudson and Christopher D Manning. "Gqa: A new dataset for real-world visual reasoning and compositional question answering." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6700–6709, 2019.

[HWA22] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. "LoRA: Low-Rank Adaptation of Large Language Models." In *ICLR*, 2022.

[HYH21] Xiaotian Han, Jianwei Yang, Houdong Hu, Lei Zhang, Jianfeng Gao, and Pengchuan Zhang. "Image Scene Graph Generation (SGG) Benchmark.", 2021.

[HYL20] Xiaowei Hu, Xi Yin, Kevin Lin, Lijuan Wang, Lei Zhang, Jianfeng Gao, and Zicheng Liu. "VIVO: Surpassing Human Performance in Novel Object Captioning with Visual Vocabulary Pre-Training." *arXiv preprint arXiv:2009.13682*, 2020.

[HZL20] Zhicheng Huang, Zhaoyang Zeng, Bei Liu, Dongmei Fu, and Jianlong Fu. "Pixel-BERT: Aligning Image Pixels with Text by Deep Multi-Modal Transformers." *arXiv preprint arXiv:2004.00849*, 2020.

[IZF20] Gabriel Ilharco, Rowan Zellers, Ali Farhadi, and Hannaneh Hajishirzi. "Probing Text Models for Common Ground with Visual Representations." *arXiv preprint arXiv:2005.00619*, 2020.

[JKF19] Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. "Action Genome: Actions as Composition of Spatio-temporal Scene Graphs.", 2019.

[JNC18] Yu Jiang, Vivek Natarajan, Xinlei Chen, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. "Pythia v0. 1: the winning entry to the VQA challenge 2018." *arXiv preprint arXiv:1807.09956*, 2018.

[JYX21] 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 *International Conference on Machine Learning (ICML)*, 2021.

[KB15] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization." In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2015.

[KCG22] Amita Kamath, Christopher Clark, Tanmay Gupta, Eric Kolve, Derek Hoiem, and Aniruddha Kembhavi. "Webly supervised concept expansion for general purpose vision models." In *ECCV*, 2022.

[KDA17] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Andreas Veit, et al. "Openimages: A public dataset for large-scale multi-label and multi-class image classification." *Dataset available from https://github. com/openimages*, **2**(3):18, 2017.

[KF15] Andrej Karpathy and Li Fei-Fei. "Deep visual-semantic alignments for generating image descriptions." In *CVPR*, 2015.

[KFM20] Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. "The Hateful Memes Challenge: Detecting Hate Speech in Multimodal Memes." *arXiv preprint arXiv:2005.04790*, 2020.

[KH09] Alex Krizhevsky, Geoffrey Hinton, et al. "Learning multiple layers of features from tiny images." 2009.

[KJZ18] Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. "Bilinear attention networks." In *NeurIPS*, 2018.

[KLW19] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. "Few-shot object detection via feature reweighting." In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8420–8429, 2019.

[KRA18] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, et al. "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale." *arXiv preprint arXiv:1811.00982*, 2018.

[KRA20] 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." In *International Journal of Computer Vision (IJCV)*, 2020.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks." In *Proceedings of the Conference on Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1097–1105, 2012.

[KSL21] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. "MDETR-modulated detection for end-to-end multi-modal understanding." In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1780–1790, 2021.

[KZB19] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. "Revisiting self-supervised visual representation learning." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[KZG17] 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." *International Journal of Computer Vision (IJCV)*, 2017.

[LAC21] Brian Lester, Rami Al-Rfou, and Noah Constant. "The power of scale for parameter-efficient prompt tuning." *arXiv preprint arXiv:2104.08691*, 2021.

[LCD18] Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. "Unsupervised Machine Translation Using Monolingual Corpora Only." In *International Conference on Learning Representations (ICLR)*, 2018.

[LCH18] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. "Stacked cross attention for image-text matching." In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[LDD19] Jiasen Lu, Batra Dhruv, Parikh Devi, and Lee Lee. "ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks." *arXiv preprint arXiv:1908.02265*, 2019.

[LDF19] Gen Li, Nan Duan, Yuejian Fang, Daxin Jiang, and Ming Zhou. "Unicoder-VL: A universal encoder for vision and language by cross-modal pre-training." *arXiv preprint arXiv:1908.06066*, 2019.

[LDF20] Gen Li, N. Duan, Yuejian Fang, Daxin Jiang, and M. Zhou. "Unicoder-VL: A Universal Encoder for Vision and Language by Cross-modal Pre-training." In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2020.

[LDP23] Liunian Harold Li*, Zi-Yi Dou*, Nanyun Peng, and Kai-Wei Chang. "DesCo: Learning Object Recognition with Rich Language Descriptions." *NeurIPS*, 2023.

[LGB19] Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. "Linguistic Knowledge and Transferability of Contextual Representations." In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 1073–1094, 2019.

[LGG17] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. "Focal loss for dense object detection." In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

[LGY23] Chunyuan Li, Zhe Gan, Zhengyuan Yang, Jianwei Yang, Linjie Li, Lijuan Wang, and Jianfeng Gao. "Multimodal foundation models: From specialists to general-purpose assistants." *arXiv preprint*, 2023.

[LLC21] 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." *arXiv preprint arXiv:2103.14030*, 2021.

[LLL22] Chunyuan Li, Haotian Liu, Liunian Harold Li, Pengchuan Zhang, Jyoti Aneja, Jianwei Yang, Ping Jin, Yong Jae Lee, Houdong Hu, Zicheng Liu, et al. "ELEVATER: A Benchmark and Toolkit for Evaluating Language-Augmented Visual Models." *NeurIPS Datasets and Benchmarks Track*, 2022.

[LLW23] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. "Visual instruction tuning." *arXiv preprint*, 2023.

[LLW24] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. "Visual instruction tuning." *Advances in neural information processing systems*, **36**, 2024.

[LMB14a] 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 *European Conference on Computer Vision*. Springer, 2014.

[LMB14b] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. "Microsoft COCO: Common Objects in Context." In *ECCV*, 2014.

[LOG19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar S. Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke S. Zettlemoyer, and Veselin Stoyanov. "RoBERTa: A Robustly Optimized BERT Pretraining Approach." *arXiv*, 2019.

[LRN19] Iro Laina, Christian Rupprecht, and Nassir Navab. "Towards unsupervised image captioning with shared multimodal embeddings." In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2019.

[LSD15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

[LT20] Zhao Li and Kewei Tu. "Unsupervised Cross-Lingual Adaptation of Dependency Parsers Using CRF Autoencoders." In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 2127–2133, 2020.

[LWB22] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. "Language-driven Semantic Segmentation." In *ICLR*, 2022.

[LWL20] Yikuan Li, Hanyin Wang, and Yuan Luo. "A comparison of pre-trained vision-and-language models for multimodal representation learning across medical images and reports." In *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2020.

[LYF21] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing." *arXiv preprint arXiv:2107.13586*, 2021.

[LYL20] 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 *European Conference on Computer Vision*, pp. 121–137. Springer, 2020.

[LYW21] Liunian Harold Li, Haoxuan You, Zhecan Wang, Alireza Zareian, Shih-Fu Chang, and Kai-Wei Chang. "Unsupervised Vision-and-Language Pre-training Without Parallel Images and Captions." In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5339–5350, 2021.

[LYY19] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. "Visualbert: A simple and performant baseline for vision and language." 2019.

[LZR23] 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*, 2023.

[LZS23] Feng Li, Hao Zhang, Peize Sun, Xueyan Zou, Shilong Liu, Jianwei Yang, Chunyuan Li, Lei Zhang, and Jianfeng Gao. "Semantic-SAM: Segment and recognize anything at any granularity." *arXiv preprint*, 2023.

[LZZ22] Liunian Harold Li*, Pengchuan Zhang*, Haotian Zhang*, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang,

Kai-Wei Chang, and Jianfeng Gao. “Grounded Language-Image Pre-training.” *CVPR*, 2022.

[MCL14] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. “The role of context for object detection and semantic segmentation in the wild.” In *CVPR*, 2014.

[MGS22] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, et al. “Simple open-vocabulary object detection with vision transformers.” In *ECCV*, 2022.

[MH08] Laurens van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE.” *Journal of machine learning research*, 2008.

[MV23] Sachit Menon and Carl Vondrick. “Visual Classification via Description from Large Language Models.” In *ICLR*, 2023.

[NF16] Mehdi Noroozi and Paolo Favaro. “Unsupervised learning of visual representations by solving jigsaw puzzles.” In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[OKB11a] Vicente Ordonez, Girish Kulkarni, and Tamara L. Berg. “Im2Text: Describing Images Using 1 Million Captioned Photographs.” In *Neural Information Processing Systems (NIPS)*, 2011.

[OKB11b] Vicente Ordonez, Girish Kulkarni, and Tamara L. Berg. “Im2Text: Describing Images Using 1 Million Captioned Photographs.” In *Proceedings of the Conference on Advances in Neural Information Processing Systems (NeurIPS)*, 2011.

[PAR21] Cory Paik, Stéphane Aroca-Ouellette, Alessandro Roncone, and Katharina Kann. "The World of an Octopus: How Reporting Bias Influences a Language Model's Perception of Color." In *EMNLP*, 2021.

[PKD16] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. "Context encoders: Feature learning by inpainting." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[PNI18] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. "Deep Contextualized Word Representations." In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2227–2237, 2018.

[PSG19] Telmo Pires, Eva Schlinger, and Dan Garrette. "How Multilingual is Multilingual BERT?" In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4996–5001, 2019.

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher Manning. "GloVe: Global Vectors for Word Representation." In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.

[PWC15] 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*, 2015.

[RBL22] 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*, pp. 10684–10695, 2022.

[RDG16] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. "You only look once: Unified, real-time object detection." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

[RHG15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. "Faster R-CNN: Towards real-time object detection with region proposal networks." In *NeurIPS*, 2015.

[RKB20] Shafin Rahman, Salman Khan, and Nick Barnes. "Improved visual-semantic alignment for zero-shot object detection." In *34th AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

[RKH21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. "Learning transferable visual models from natural language supervision." In *International Conference on Machine Learning (ICML)*, 2021.

[RKK23] Karsten Roth, Jae Myung Kim, A Koepke, Oriol Vinyals, Cordelia Schmid, and Zeynep Akata. "Waffling around for Performance: Visual Classification with Random Words and Broad Concepts." *ICCV*, 2023.

[RKP20] Shafin Rahman, Salman H Khan, and Fatih Porikli. "Zero-shot object detection: Joint recognition and localization of novel concepts." *International Journal of Computer Vision*, **128**(12):2979–2999, 2020.

[RRH16] Anna Rohrbach, Marcus Rohrbach, Ronghang Hu, Trevor Darrell, and Bernt Schiele. "Grounding of textual phrases in images by reconstruction." In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[SDG18] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. "Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning." In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2556–2565, 2018.

[SGM13] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. "Zero-shot learning through cross-modal transfer." *NeurIPS*, 2013.

[SLH22] Sheng Shen, Chunyuan Li, Xiaowei Hu, Yujia Xie, Jianwei Yang, Pengchuan Zhang, Zhe Gan, Lijuan Wang, Lu Yuan, Ce Liu, et al. "K-LITE: Learning transferable visual models with external knowledge." In *NeurIPS*, 2022.

[SLZ19] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. "Objects365: A large-scale, high-quality dataset for object detection." In *Proceedings of the IEEE international conference on computer vision*, pp. 8430–8439, 2019.

[SNS19] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. "Towards VQA models that can read." In *CVPR*, 2019.

[SRL20] Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. "AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts." In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4222–4235, 2020.

[SSD23] Samuel Schulter, Yumin Suh, Konstantinos M Dafnis, Zhixing Zhang, Shiyu Zhao, Dimitris Metaxas, et al. "OmniLabel: A Challenging Benchmark for Language-Based Object Detection." *arXiv preprint*, 2023.

[SSS17] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. "Revisiting unreasonable effectiveness of data in deep learning era." In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2017.

[SXL19] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. "Deep high-resolution representation learning for human pose estimation." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5693–5703, 2019.

[SZC19] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. "VL-BERT: Pre-training of generic visual-linguistic representations." *arXiv preprint arXiv:1908.08530*, 2019.

[SZZ19] Alane Suhr, Stephanie Zhou, Iris Zhang, Huajun Bai, and Yoav Artzi. "A corpus for reasoning about natural language grounded in photographs." *ACL*, 2019.

[TB19] Hao Tan and Mohit Bansal. "LXMERT: Learning Cross-Modality Encoder Representations from Transformers." In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5100–5111, 2019.

[TB20] Hao Tan and Mohit Bansal. "Vokenization: Improving Language Understanding with Contextualized, Visual-Grounded Supervision." In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2066–2080, 2020.

[TJB22] 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.

[TL18] Trieu H Trinh and Quoc V Le. "A Simple Method for Commonsense Reasoning." *Arxiv*, 2018.

[TL19] Mingxing Tan and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." In *International Conference on Machine Learning*, pp. 6105–6114. PMLR, 2019.

[TLI23] 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." *arXiv preprint*, 2023.

[TMC21] Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. "Multimodal few-shot learning with frozen language models." *NeurIPS*, 2021.

[TSC19] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. "Fcos: Fully convolutional one-stage object detection." In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9627–9636, 2019.

[VSP17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." In *Proceedings of the Conference on Advances in Neural Information Processing Systems (NeurIPS)*, pp. 5998–6008, 2017.

[VTM19] Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. "Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned." In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5797–5808, 2019.

[WCC23] Wenhai Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, et al. "VisionLLM: Large language model is also an open-ended decoder for vision-centric tasks." *arXiv preprint*, 2023.

[WD19] Shijie Wu and Mark Dredze. "Beto, Bentz, Becas: The Surprising Cross-Lingual Effectiveness of BERT." In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 833–844, 2019.

[WHD20] Xin Wang, Thomas E Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu. "Frustratingly simple few-shot object detection." *arXiv preprint arXiv:2003.06957*, 2020.

[WRH19] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. "Meta-learning to detect rare objects." In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9925–9934, 2019.

[WTS20] Qinxin Wang, Hao Tan, Sheng Shen, Michael Mahoney, and Zhewei Yao. "MAF: Multimodal Alignment Framework for Weakly-Supervised Phrase Grounding." In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 2020.

[XDL22] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. "GroupViT: Semantic segmentation emerges from text supervision." In *CVPR*, 2022.

[XSJ17] Fanyi Xiao, Leonid Sigal, and Yong Jae Lee. "Weakly-supervised visual grounding of phrases with linguistic structures." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[XWW18] Bin Xiao, Haiping Wu, and Yichen Wei. “Simple baselines for human pose estimation and tracking.” In *Proceedings of the European conference on computer vision (ECCV)*, pp. 466–481, 2018.

[XZC17] Danfei Xu, Yuke Zhu, Christopher B. Choy, and Li Fei-Fei. “Scene Graph Generation by Iterative Message Passing.” *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017.

[XZH21] 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.” *arXiv preprint arXiv:2106.09018*, 2021.

[YBK23] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. “When and why vision-language models behave like bag-of-words models, and what to do about it?” In *ICLR*, 2023.

[YCX19] Xiaopeng Yan, Ziliang Chen, Anni Xu, Xiaoxi Wang, Xiaodan Liang, and Liang Lin. “Meta r-cnn: Towards general solver for instance-level low-shot learning.” In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9577–9586, 2019.

[YHW22] 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.” In *NeurIPS*, 2022.

[YLS18] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. “Mattnet: Modular attention network for referring expression comprehension.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[YLY19] Jun Yu, Jing Li, Zhou Yu, and Qingming Huang. "Multimodal Transformer with Multi-View Visual Representation for Image Captioning." *arXiv preprint arXiv:1905.07841*, 2019.

[YLZ21] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. "Focal self-attention for local-global interactions in vision transformers." *arXiv preprint arXiv:2107.00641*, 2021.

[YPY16a] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. "Modeling context in referring expressions." In *ECCV*, 2016.

[YPY16b] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. "Modeling Context in Referring Expressions." In *ECCV*, 2016.

[YPZ23] Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. "Language in a Bottle: Language Model Guided Concept Bottlenecks for Interpretable Image Classification." In *CVPR*, 2023.

[YTY20] Fei Yu, Jiji Tang, Weichong Yin, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. "ERNIE-ViL: Knowledge Enhanced Vision-Language Representations Through Scene Graph." *arXiv preprint arXiv:2006.16934*, 2020.

[YTZck] Mingzhang Yin, George Tucker, Mingyuan Zhou, Sergey Levine, and Chelsea Finn. "Meta-Learning without Memorization." In *ICLR*, 2020.

[YWZ23] An Yan, Yu Wang, Yiwu Zhong, Chengyu Dong, Zexue He, Yujie Lu, William Yang Wang, Jingbo Shang, and Julian McAuley. "Learning concise and descriptive attributes for visual recognition." In *ICCV*, 2023.

[YYC19] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. "Deep Modular Co-Attention Networks for Visual Question Answering." In *CVPR*, 2019.

[ZBF19] Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. "From recognition to cognition: Visual commonsense reasoning." In *CVPR*, 2019.

[ZCS23] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. "MiniGPT-4: Enhancing vision-language understanding with advanced large language models." *arXiv preprint*, 2023.

[ZDY21] Pengchuan Zhang, Xiyang Dai, Jianwei Yang, Bin Xiao, Lu Yuan, Lei Zhang, and Jianfeng Gao. "Multi-scale vision longformer: A new vision transformer for high-resolution image encoding." *arXiv preprint arXiv:2103.15358*, 2021.

[ZDY23] Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Behl, Jianfeng Wang, Lu Yuan, et al. "Generalized Decoding for Pixel, Image, and Language." In *CVPR*, 2023.

[ZGJ22] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. "Detecting twenty-thousand classes using image-level supervision." In *ECCV*, 2022.

[ZGL20] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le. "Rethinking Pre-training and Self-training." *Advances in Neural Information Processing Systems*, **33**, 2020.

[ZKZ15] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books." In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2015.

[ZLH21] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. "Vinvl: Revisiting visual representations in vision-language models." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5579–5588, 2021.

[ZLL21] Wangchunshu Zhou, Qifei Li, and Chenle Li. "Learning from Perturbations: Diverse and Informative Dialogue Generation with Inverse Adversarial Training." In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 694–703, 2021.

[ZLL22] Tiancheng Zhao, Peng Liu, Xiaopeng Lu, and Kyusong Lee. "OmDet: Language-Aware Object Detection with Large-scale Vision-Language Multi-dataset Pre-training." *arXiv preprint*, 2022.

[ZLZ23] Hao Zhang, Feng Li, Xueyan Zou, Shilong Liu, Chunyuan Li, Jianwei Yang, and Lei Zhang. "A simple framework for open-vocabulary segmentation and detection." In *ICCV*, 2023.

[ZPZ20a] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J Corso, and Jianfeng Gao. "Unified Vision-Language Pre-Training for Image Captioning and VQA." In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2020.

[ZPZ20b] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J Corso, and Jianfeng Gao. "Unified Vision-Language Pre-Training for Image Captioning and VQA." *AAAI*, 2020.

[ZRH21] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. "Open-vocabulary object detection using captions." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14393–14402, 2021.

[ZSL20] 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.

[ZWS20] Pengkai Zhu, Hanxiao Wang, and Venkatesh Saligrama. "Don't Even Look Once: Synthesizing Features for Zero-Shot Detection." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[ZYZ22] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, et al. "RegionCLIP: Region-based language-image pretraining." In *CVPR*, 2022.

[ZYZ23] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Gao, and Yong Jae Lee. "Segment everything everywhere all at once." *arXiv preprint*, 2023.

[ZZH22] 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." *NeurIPS*, 2022.

[ZZP17] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. "Scene parsing through ade20k dataset." In *CVPR*, 2017.