Automating Annotation Pipelines by leveraging Multi-Modal Data

Anish Madan

CMU-RI-TR-24-39

July 2

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Professor Deva K. Ramanan, *chair*
Professor Katerina Fragkiadaki
Neehar Peri

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

To my parents.

Abstract

The era of vision-language models (VLMs) trained on large web-scale datasets challenges conventional formulations of "open-world" perception. In this work, we revisit the task of few-shot object detection (FSOD) in the context of recent foundational VLMs. First, we point out that *zero*-shot VLMs such as GroundingDINO significantly outperform state-of-the-art *few*-shot detectors (48 vs. 33 AP) on COCO. Despite their strong zero-shot performance, such foundational models may still be sub-optimal. For example, `trucks` on the web may be defined differently from `trucks` for a target application such as autonomous vehicle perception. We argue that the task of few-shot recognition can be reformulated as *aligning* foundation models to target concepts using a few examples. Interestingly, such examples can be multi-modal, using both text and visual cues, mimicking *instructions* that are often given to human annotators when defining a target concept of interest. Concretely, we propose Foundational FSOD, a new benchmark protocol that evaluates detectors pre-trained on any external datasets and fine-tuned on multi-modal (text and visual) K-shot examples per target class. We repurpose nuImages for Foundational FSOD, benchmark several popular open-source VLMs, and provide an empirical analysis of state-of-the-art methods. Lastly, we discuss our recent CVPR 2024 Foundational FSOD competition and share insights from the community. Notably, the winning team significantly outperforms our baseline by 23.9 mAP!

Acknowledgments

My time at CMU over the past two years has been an incredible journey marked by a steep learning curve that pushed my limits both professionally and personally. I am extremely fortunate to have met and interacted with so many remarkable individuals, whose support and guidance made this journey possible.

First and foremost, I would like to express my deepest gratitude to my advisor, Deva Ramanan, for his unwavering support, wisdom, and technical expertise. His keen intuition in Computer Vision and invaluable insights into conducting research and communicating ideas have been immensely inspiring and educational. One of the most significant lessons I have learned from him is the importance of thoroughly understanding the question or problem before even considering potential answers. Gaining a deep understanding of the question often means more than half the battle is won. This principle has proven extremely helpful in conducting research, processing feedback and reviews, and discussing ideas with collaborators and friends. I am deeply thankful to Deva for taking a chance on me and providing guidance over the past couple of years.

I am also grateful to my other committee members, Prof. Katerina Fragkiadaki and Neehar Peri, for taking time out of their busy schedules to attend my thesis talk and provide insightful feedback. A special mention goes to my collaborator and post-doc alumnus, Shu Kong, who helped refine our project ideas and experiments and was always available near conference deadlines to resolve any issues.

I would be remiss not to mention the incredible lab mates I had the chance to work and interact with. A special shoutout to Neehar for being an awesome mentor, entertaining all my questions, and helping me learn how to conduct research and design experiments. He always had my best interests at heart, whether preparing for conference submissions, talks, or grad school applications. I have thoroughly enjoyed our impromptu chatting sessions, which often started with project updates and ended on topics far from work. I am immensely thankful to my super smart and hardworking lab members (listed alphabetically) from whom I've learned a lot and had fun at various social events: Amy, Andrew(s), Anirudh, Arun, Erica, Gautam, Gengshan, Haithem, Jeff, Jenny, Jason, Kangle,

Khiem, Meghana, Nate, Neehar, Nikhil, Sally, Tarasha, Zihan, and Zhiqiu.

Additionally, I would like to highlight some of the Smith Hall folks who made the countless hours spent working more bearable. I think I had the most chatty table in Smith Hall, with Bharath across from me, sharing food cravings and chatting about any topic while also grinding out all-nighters together on conference deadlines. Our table and conversations often included folks from all ends of Smith Hall, including Poorvi, Shagun, Moneish, Prachi, Sriram, Vishnu, Himangi, Swami, Z, Yanbo, Jeff, Judy, and countless others. Whenever I felt like not working, I used to look to the other end of the hall and see Shivam deeply engrossed in his work, which motivated me to get my work done.

I have been incredibly fortunate to have a solid group of friends who became my second family here. Moving to the U.S. was daunting, as I knew no one, but I was lucky to have my undergrad labmates and friends, Sarthak and Shagun, in the same program. From working with the same undergrad advisor to now living as flatmates, I couldn't have asked for better company. At CMU, I made some amazing friends. Shreya, an endless source of youthful energy, was always ready for fun and became my flatmate in the second year. Dvij, extremely hardworking yet never said no to any plan. Pranay, a constant source of jokes, tote-bag lover, and occasional gym partner, kept the mood light. Pushkal, always a vibe when having fun, and as clueless as me regarding his career, and Sriram, one of the nicest and most helpful people ever.

Finally, a big shoutout to Poorvi, who has been a constant source of happiness and positivity this past year. She has always been there to help when needed, distract me when I'm stressed with her puns and one-liners, and provide support. From playing table tennis in the Robolounge for the past two years to traveling the country for conferences and pleasure, I couldn't have asked for a better group to spend time with. I am also thankful to some of my undergrad friends in the U.S.: Pranav, Pulkit, Harshita, Aditya, and Brihi, who have been supportive and helpful whenever I needed to talk about anything.

Finally, I want to thank my family for allowing me to pursue my interests and career so far away from them. Given that they knew no one in the U.S., it is incredible that they didn't even hesitate about my plans to study here. I am super grateful for their unconditional support and, of course, the awesome snacks they send from home.

Funding

This work was supported in part by funding from Bosch Research.

Contents

When this dissertation is viewed as a PDF, the page header is a link to this Table of Contents.

List of Figures

List of Tables

Chapter 1

Introduction

Vision-language models (VLMs) trained on (often proprietary) web-scale datasets have disrupted traditional notions of the "open-world," particularly for few-shot recognition. In this paper, we revisit few-shot object detection (FSOD) in the context of these foundational models, propose a new benchmark protocol that allow foundational models to “enter the conversation", and present several simple baselines.

First, we highlight that *zero-shot* VLMs like GroundingDINO demonstrate a remarkable improvement over state-of-the-art *few-shot* detectors (48.3 vs. 33.1 AP) on COCO, as shown in Table 4.1. In hindsight, this is not surprising, as the former is pre-trained on far more data (that may include visual examples of the target concept), while the later is pre-trained on data that is explicitly curated to avoid target concepts of interest. From this perspective, VLMs violate the current training protocol of few-shot benchmarks, suggesting that such protocols need to be rethought in the foundational era.

Concept Alignment. Despite their impressive performance, foundation models used in a zero-shot fashion can still be sub-optimal. For example, `trucks` as defined for a particular target application like perception for autonomous vehicles may differ from `trucks` as found on the web (cf. Fig. 1.1). Indeed, this well-known observation has created the ad-hoc practice of prompt engineering, where users actively search for a textual prompt that elicits the desired zero-shot behaviour. Instead, we argue that one can principally address the challenge of *aligning* foundation models to target concepts through the lens of few-shot recognition, by presenting VLMs with a few

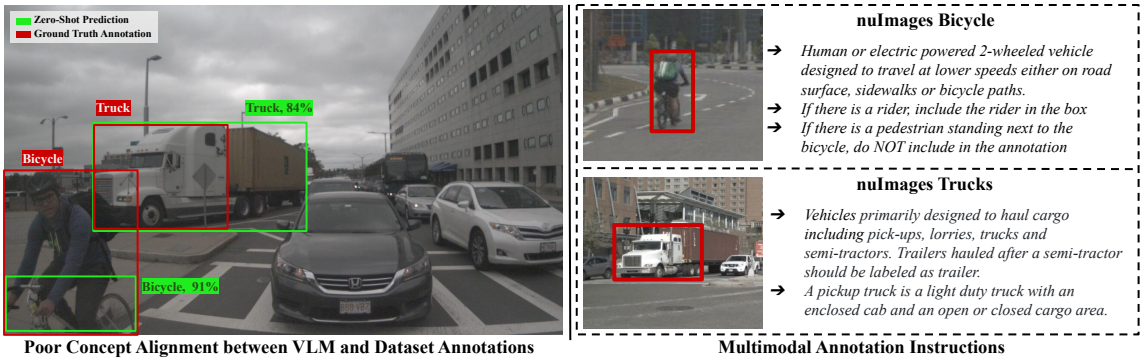

Figure 1.1: **Poor Alignment Between Vision Language Models (VLMs) and Target Concepts**. Although VLMs show impressive zero-shot performance, they struggle when the target class is different from concepts encountered in web-scale training. On the **left**, we see that the nuImages dataset [2] defines the cab of the `truck` as a separate concept from its `trailer` (shown in **red**). In contrast, the VLM predicts the entire vehicle as a `truck` (shown in **green**). Similarly, nuImages annotations dictate that a person riding a bicycle must also be labeled as part of `bicycle` (shown in **red**) unlike the VLM prediction (in **green**). On the **right**, we present the actual *class definitions* given to the nuImages annotators, provided as both textual descriptions and visual examples. Just as human annotators learn concepts from few-shot multi-modal examples, we argue that VLMs should be aligned to K vision-language examples.

examples of the target concept. Crucially, such examples can be multi-modal, using both text and visual cues, mimicking the natural few-shot *multi-modal instructions* that are often given to human annotators when defining a target concept of interest [3]. Before introducing our new protocol, we first review the conventional FSOD setup below.

Conventional FSOD. Existing FSOD benchmarks partition object detection datasets like PASCAL VOC [7] and COCO [30] into `base` and `novel` classes. Detectors pre-train on `base` and then learn `novel` classes given K examples (or K-shots). Current protocols enforce `base` and `novel` to be disjoint to prevent concept leakage, allowing one to evaluate generalization to the "open-world". However, as most detectors are pre-trained on ImageNet, we point out that *concept leakage already occurs in current FSOD protocols*. For example, `cat` and `person` are deemed `novel` for COCO-FSOD but are present in ImageNet data used to pre-train detectors [43]. Moreoever, `car` is deemed `novel`, but similar concepts like `sports car` and `race car` are present in ImageNet, illustrating the difficulty of even defining leakage.

Foundational FSOD. We believe that concept leakage should be embraced. Our

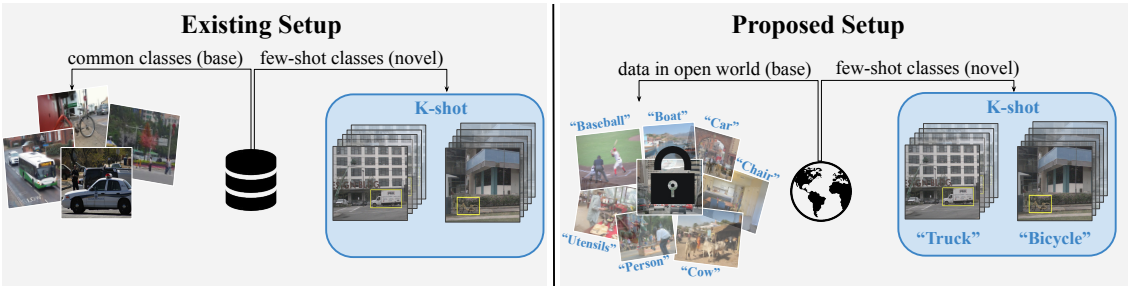

Figure 1.2: **Foundational Few-shot Object Detection (FSOD)**. Conventional FSOD protocols (**left**) allow for pre-training on `base` classes (with many examples per class) and then fine-tuning on K-shots of `novel` classes, where `novel` and `base` are designed to be disjoint. However, we point out that pre-training datasets such as ImageNet often contain classes similar to the `novel` classes, highlighting the issue of concept leakage. As preventing concept leakage is difficult (if not impossible) and appears to be artificial in the foundational era, we propose the setup of *foundational FSOD* (**right**). Our setup allows for pre-training on massively-large (and potentially proprietary) datasets, typical for foundational VLM models. Since these models can process both text and images, one can utilize such multi-modal K-shot examples to *align* VLMs with the target concepts of interest.

Foundational FSOD protocol replaces the `base` pre-training stage with web-scale pre-training, where such data may be proprietary and not fully disclosed [38]. *We argue that pre-training on large-scale data will be the key enabler for generalization to the open-world.* Note that this hypothesis is difficult to even evaluate under current few-shot protocols, motivating our setup. Moreover, another key property is that K-shots may include multi-modal examples spanning both images and text, motivating a multi-modal adaptation stage that aligns the VLM to the target concepts (cf. Fig. 1.2). We repurpose nuImages for our Foundational FSOD benchmark, a challenging dataset due to open-world categories such as `debris` and `pushable-pullable`, which also provides multi-modal annotation instructions.

Contributions. We present three major contributions. First, we modernize FSOD benchmarks by embracing vision-language foundation models that are pretrained on internet-scale data. We highlight the practical challenge of using multi-modal few-shot examples to define the target semantic concept (as shown in Fig. 1.1). Next, we adapt nuImages for Foundational FSOD, evaluate various popular open-source VLMs, and present an empirical analysis of leading methods. Lastly, we highlight the results from our recent CVPR 2024 challenge hosted in conjunction with the Workshop on Visual Perception via Learning in An Open World.

Chapter 2

Related Work

2.1 Few-Shot Object Detection (FSOD)

FSOD aims to detect new categories with limited training data [25]. Recent work explores two primary approaches: meta-learning and transfer learning. Meta-learning-based methods focus on acquiring generalizable features from a set of `base` classes, which can then be applied to identify `novel` classes. For example, [22] proposes a technique that re-weights features from base classes to predict `novel` classes. [50] proposes a framework addressing both few-shot object detection and few-shot viewpoint estimation. [8] introduces a general FSOD network that learns a matching metric between image pairs, while [47] enhances object features using a universal prototype. More recently, [52] proposes a generative approach that is robust to noisy object proposals for `novel` classes. In contrast, transfer learning involves partially freezing network weights pretrained on a `base` dataset to improve a model's ability to generalize to `novel` classes with limited data. Transfer learning approaches often follow a two-stage fine-tuning strategy: first train on `base` classes and then fine-tune the box classifier and regressor with K-shots from `novel` classes. This strategy has historically outperformed meta-learning approaches [43]. Recent work has primarily focused on improving classification performance. [42] utilizes a contrastive proposal encoding loss to encourage instance-level intra-class compactness and inter-class variance. Similarly, [28] applies a class margin loss to balance inter and intra-class margins. Our approach leverages transfer-learning by fine-tuning vision-language

models (VLMs) pre-trained on large-scale datasets.

2.2 Vision-Language Models (VLMs)

VLMs are trained on a large-scale collection of weakly-supervised image-text pairs collected from the web. These models embed images and text into a shared space, enabling open-vocabulary detection. Early works adapt VLMs for object detection by either distilling the model's predictions for specific image regions [12, 13] or directly incorporating detection components into frozen [26] or fine-tuned [6, 35, 36] encoders. In contrast, RegionCLIP [58] employs a multi-stage training approach, which involves generating pseudo-labels from captioning data, conducting region-text contrastive pre-training, and fine-tuning on detection data. GLIP [29] uses a single text query for the entire image and frames detection as a phrase grounding problem. Detic [62] addresses long-tail detection performance by leveraging image-level supervision. In the context of open-vocabulary detection, there may be some overlap between categories seen during training and testing. We use the term "zero-shot inference" to signify that a model has never been trained on the target dataset.

2.3 Foundation Model Fine-Tuning

Fine-Tuning Foundation Models is of significant interest across many application areas [10, 19, 56]. Standard fine-tuning procedures employ both linear probing [4, 16, 17] and full-finetuning [24, 44, 49] to adapt models to downstream tasks. However, such methods may be suboptimal as they can be computationally expensive. Instead, recent works like CLIP-Adapter [10] and Tip-Adapter [57] fine-tune CLIP using parameter-efficient methods [18, 20, 55] which optimize lightweight MLPs while keeping the encoder frozen. Similarly, inspired by the success of prefix-tuning in language models [5, 11, 15, 21], prompt adaptation [32, 51, 59, 63] replaces hand-crafted prompts like "a photo of a `{cls}`" with learned word embeddings. CoOp [60] and other prompting methods [32, 59, 63] finetune CLIP via prefix-tuning. Different from most prior work, we investigate fine-tuning strategies for VLM-based detectors using few-shot *multi-modal* examples.

Chapter 3

Foundational FSOD Benchmark

As shown in Fig 1.2, our proposed Foundational FSOD benchmark utilizes vision-language models (VLMs) pre-trained on diverse, large-scale datasets, which are then aligned to K examples of each target class. We contrast our proposed setup with standard benchmarks and present simple strategies for fine-tuning VLMs below.

3.1 Foundational FSOD Benchmark

Existing FSOD benchmarks repurpose well-established datasets like PASCAL VOC [7] and COCO [30] by partitioning them into `base` and `novel` classes for pre-training and fine-tuning, respectively. For COCO, the 60 categories disjoint with PASCAL VOC are used as `base` classes and the remaining 20 are used as `novel` classes [43]. However, this setup is artificial and does not reflect how FSOD is deployed in practice. First, the FSOD benchmarks construct `novel` classes by including common concepts such as `car` and `person`, and require FSOD methods to detect these common classes by assuming they have only few examples. Importantly, VLMs like GroundingDINO [31] can already detect common categories with high accuracy on COCO *without fine-tuning* (cf. Table 4.1). Therefore, we focus on benchmarking Foundational FSOD on more realistic and challenging datasets like nuImages [2]. Second, existing FSOD benchmarks require that datasets are partitioned into `base` and `novel` classes, which is infeasible for large-scale (often private) foundational datasets. For example, although CLIP's [38] model weights are publicly available, its pre-training dataset is

not. Instead, FSOD methods should only fine-tune VLMs on K-shot annotations for C target classes (or `novel`, as termed in the conventional FSOD benchmark), and also evaluate performance on these C classes.

3.2 Few-Shot Multi-Modal Concept Alignment

Although VLMs achieve strong zero-shot performance on common classes, they struggle when the target class is different from concepts encountered on the web (cf. Fig. 1.1). For example, nuImages [2] defines the cab of a `truck` as a separate concept from its `trailer`. However, Detic detects the entire vehicle as `truck`. This fine-grained distinction is provided to human annotators with visual examples and textual descriptions. We explore seven methods for alignment (either explicitly by updating model weights via fine-tuning or in context via prompting) VLMs below.

Prompt Engineering uses expressive descriptions in the text prompt, adding attributes, synonyms or language context, to manually improve the alignment of foundation model outputs to target concepts of interest. In our case, we prompt VLMs with synonyms of the nuImages classes to improve detection accuracy. For example, we augment the language query for `pushable-pullable` with synonyms like `cart` and `wheel barrow`.

Standard Fine-Tuning updates the last few layers of a model to adapt to new target classes. For two-stage object detectors, this typically requires training the box regression and classifier head. However, we find that standard fine-tuning is sub-optimal, motivating our proposed approach below.

Federated Fine-Tuning leverages a simple but evidently underappreciated observation: few-shot object detection datasets are actually federated datasets [14]. A federated dataset is comprised of smaller mini-datasets, where each mini-dataset is exhaustively annotated for only a single category. For example, `cars` may or may not appear in the background of the K images annotated with `motorcycles`. However, existing FSOD methods incorrectly assume that no `cars` are present in the background of non-`car` images. We devise a simple loss that incorporates this insight, discussed further in the supplement.

Language Prompt Tuning is an established parameter-efficient strategy [27, 41] for updating text embeddings with few-shot examples via fine-tuning. Concretely, for

a given language query (e.g. `stroller`), we first extract a text embedding P^0 and only fine-tune the text embedding [29].

Visual Prompting uses images of target concepts that are difficult to describe through text as prompts to learn novel concepts in-context. For example, although `debris` is a difficult catchall category to define through text, we can use image examples to improve concept alignment. Typically, visual prompts are tokenized and fed as inputs to a frozen VLM.

Multi-Modal Prompting combines language and visual prompting to leverage multi-modal features. Intuitively, multi-modal cues can yield better alignment than uni-modal cues alone; in the above case, ambiguous concepts such as `debris` can be clarified with both textual descriptions (e.g `trash can` and `tree branch`) and visual examples (similar to the multi-modal annotator instructions in Fig. 1.1!). Here, visual and language prompts are tokenized and separately fed as inputs to a frozen VLM. Specifically, MQDet[53] introduces a lightweight module: Gated Class Scalable Perceiver, that fuses visual cues and text descriptions in the text encoder via class-wise cross attention layers.

Multi-Modal Chat Assistants can accomplish many of the same tasks as multi-modal prompting through a multi-modal turn-by-turn conversational interface. However, unlike multi-modal prompting, conversational interfaces allow users to iteratively refine concept definitions, similar to how human annotators often require several rounds of feedback to fully understand the target concept.

Chapter 4

Experiments

We conduct extensive experiments to validate that zero-shot inference from VLMs significantly improves over state-of-the-art FSOD approaches, suggesting that existing benchmarks should be re-framed to allow foundation models to "enter the conversation". Moreover, we demonstrate that leveraging language cues, especially those available for free (e.g. class names), are crucial to improving performance on data-constrained tasks like FSOD.

4.1 Datasets and Metrics.

We repurpose nuImages [2] to support the study of Foundational FSOD. This dataset annotates 18 classes, which are divided into groups with `many`, `medium`, and `few` examples [34, 37]. We report AP for each cohort. Although this dataset is not traditionally used for FSOD, nuImages' open-world categories like `debris` and `pushable-pullable` make it particularly challenging (even for VLMs), and is a realistic benchmark for Foundational FSOD. We follow the K-shot dataset creation process established by [43], described below. To construct a K-shot dataset, we select a target class c and an image at random. If the total annotations for class c in the image are less than or equal to K, we add the image to our dataset. We repeat this process for all classes until we have exactly K annotations per class. Since the specific K examples can have a significant impact on the overall performance, we run each experiment over 3 random data splits and report the average.

Table 4.1: **VLM Zero-Shot Inference Is a Strong FSOD Baseline.** Zero-shot inference with VLMs like GroundingDINO resoundingly outperforms state-of-the-art FSOD methods on the COCO FSOD benchmark, motivating the need to re-frame FSOD to embrace foundation models.

Approach	30-shots		
	AP	Base AP	Novel AP
FRCN-ft-full [54]	18.6	20.6	12.5
FRCN-BCE [54]	30.2	36.8	10.3
TFA w/ fc [43]	29.3	34.5	13.5
TFA w/cos [43]	29.9	35.3	13.6
MPSR [48]	17.1	18.1	14.1
Meta-RCNN [54]	7.8	7.1	9.1
FsDetView [50]	10.0	9.3	12.0
Retentive R-CNN [9]	32.9	39.3	13.8
DiGeo [33]	33.1	39.4	14.2
GroundingDINO (Zero-Shot) [31]	**48.3**	**46.3**	**54.3**

4.2 Zero-Shot Inference Is A Strong FSOD Baseline

We compare state-of-the-art FSOD methods with zero-shot inference from GroundingDINO [31] on COCO in Table 4.1. Surprisingly, GroundingDINO outperforms DiGeo [33] by 16.2% AP averaged across both `base` and `novel` categories despite never being trained on COCO images. GroundingDINO's impressive performance is due to its large-scale multi-modal pre-training on Objects365 [39], GoldG [23] and Cap4M [29]. It is worth noting that GroundingDINO achieves higher AP on `novel` classes than `base`, suggesting that `novel` classes in existing benchmarks (e.g. `car` and `person`) are actually not rare in the real world.

Therefore, FSOD benchmarks should be re-framed to reflect real-world applications, motivating our setup.

4.3 Foundational FSOD with nuImages

In the context of foundational models, we argue that partitioning datasets into `base` and `novel` classes no longer makes sense. Instead, FSOD methods should only train on K-shot annotations for C target classes, and also evaluate performance on these C classes. We pre-train TFA [43] on diverse datasets and fine-tune on K examples per class and highlight model performance in Table 4.2.

We train two variants of TFA trained on `COCO-base` and `LVIS-base` and fine-tune

Table 4.2: **Impact of Large-Scale Pre-Training and Language.** We repurpose nuImages for FSOD following the dataset creation process established by [43]. We group categories by frequency into `many`, `medium` and `few` examples per class [34, 37]. We fine-tune TFA on K examples, but find low performance, $<$ 3AP. However, by simply pre-training on more data (`LVIS`, `COCO` and ImageNet-21K) and leveraging language cues via a CLIP classifier, 5-shot performance improves from 2.02 AP to 15.12 AP. However, rare (or `few`) classes like `strollers`, `pushable-pullable`, and `debris` remain challenging, motivating our task of VLM alignment.

Approach	Average Precision (AP)			
	`All`	`Many`	`Medium`	`Few`
5-shots				
TFA [43] w/ `COCO-base`	1.33	2.78	1.43	0.23
TFA [43] w/ `LVIS-base`	2.02	1.69	4.08	0.58
TFA [43] w/ `LVIS,IN-21K, COCO` + CLIP Classifier	**15.12**	**22.74**	**18.99**	**4.25**
10-shots				
TFA [43] w/ `COCO-base`	1.21	2.55	1.19	0.31
TFA [43] w/ `LVIS-base`	2.27	2.05	4.51	0.58
TFA [43] w/ `LVIS,IN-21K, COCO` + CLIP Classifier	**16.09**	**25.46**	**20.00**	**3.73**
30-shots				
TFA [43] w/ `COCO-base`	1.14	2.81	0.84	0.23
TFA [43] w/ `LVIS-base`	2.23	1.48	4.98	0.45
TFA [43] w/ `LVIS,IN-21K, COCO` + CLIP Classifier	**17.22**	**25.98**	**21.64**	**4.78**

both models on K examples of the nuImages classes. Surprisingly, both variants of TFA achieve less than 3 AP (cf. Table 4.2). We posit that this is largely due to poor classification performance. Since both LVIS and COCO classes do not significantly overlap with nuImages classes, learning a classifier from few examples is extremely difficult. However, we find that simply re-training TFA with a frozen CLIP-based classifier (similar to Detic) dramatically increases performance, reiterating the utility of language and web-scale pre-training in data-constrained settings.

4.4 Empirical Analysis of Results

We evaluate several popular VLMs on the nuImages Foundational FSOD (10-shots) benchmark and present salient insights from Table 4.3 below.

Table 4.3: **Empirical Analysis of Baselines (10-Shots) on our Benchmark**. We evaluate popular VLMs on the nuImages FSOD Benchmark and find that MQ-GLIP performs the best among all baseline models. Notably, it achieves 17.0 mAP zero-shot language-only performance, and achieves 21.4 mAP via zero-shot multi-modal prompting averaged over all classes. Remarkably, our 2024 competition winners further improved performance to 45.4 mAP, beating our best baseline by 24.0%.

Approach	Backbone	Pre-Train Data	Average Precision (AP)			
			All	Many	Med	Few
Zero-Shot Detection						
RegionCLIP [58]	RN50	CC3M	2.50	3.20	3.80	0.40
Detic [62]	SWIN-B	LVIS, COCO, IN-21K	14.40	25.83	16.59	2.32
GroundingDINO [31]	SWIN-T	Objects365, GoldG, Cap4M	12.05	17.29	15.45	3.72
GLIP [29]	SWIN-L	FourODs,GoldG,Cap24M	17.01	23.36	19.86	8.40
MQ-GLIP-Text [53]	SWIN-L	Objects365, FourODs, GoldG, Cap24M	17.01	23.36	19.85	8.41
Prompt Engineering						
Detic [62]	SWIN-B	LVIS, COCO, IN-21K	14.92	26.48	17.29	2.53
GLIP [29]	SWIN-L	FourODs, GoldG, Cap24M	17.15	23.82	19.36	9.02
Standard Fine-Tuning						
RegionCLIP [58]	RN50	CC3M	3.86	6.08	5.13	0.54
Detic [62]	SWIN-B	LVIS, COCO, IN-21K	16.09	25.46	20	3.73
Federated Fine-Tuning (Ours)						
Detic [62]	SWIN-B	LVIS, COCO, IN-21K	17.24	28.07	20.71	4.18
Detic [62] w/ Prompt Engineering	SWIN-B	LVIS, COCO, IN-21K	17.71	28.46	21.14	4.75
Language Prompt Tuning						
GLIP [29]	SWIN-L	FourODs,GoldG,Cap24M	19.41	22.18	**25.16**	**10.39**
Visual Prompting						
MQ-GLIP-Image [53]	SWIN-L	Objects365,FourODs,GoldG,Cap24M	14.07	24.39	15.89	3.34
Multi-Modal Prompting						
MQ-GLIP [53]	SWIN-L	Objects365,FourODs,GoldG,Cap24M	**21.42**	**32.19**	23.29	10.26
Multi-Modal Chat Assistants						
GPT-4o Zero-Shot Classification [1]	*Private*	*Private*	9.95	16.81	12.11	1.71
CVPR 2024 Competition Results						
PHP_hhh	Private	Private	**45.35**	**64.25**	**53.43**	**20.19**
NJUST KMG	SWIN-L	Objects365V2, OpenImageV6, GoldG, V3Det, COCO2014, COCO2017, LVISV1, GRIT, RefCOCO, RefCOCO+, RefCOCOg, gRef-COCO	32.56	50.21	34.87	15.16
zjyd_cxy_vision	SWIN-L	Objects365V2, COCO2017, LVIS, GoldG, VG, OpenImagesV6, V3Det, PhraseCut, RefCOCO, RefCOCO+, RefCOCOg, gRef-COCO	31.57	46.59	33.32	17.03

Zero-Shot Detection. Somewhat unsurprisingly, we find that (i) greater pre-training data scale and diversity, along with (ii) larger backbones result in better zero-shot performance. Notably, GLIP achieves 17.01% zero-shot performance, surpassing all other methods trained with less data and smaller backbones.

Prompt Engineering. We attempt to improve zero-shot performance using synonyms for class names derived from the annotator textual instructions. We see minor improvements (e.g. Detic improves from 14.40 mAP $\rightarrow$ 14.92 mAP), indicating that leveraging rich textual descriptions beyond class names can improve concept alignment.

Federated Fine-Tuning. Standard fine-tuning is sub-optimal for FSOD, as all unannotated classes are treated as negatives. Therefore we use our zero-shot predictions to generate pseudo-labels on training images. We extract pseudo-negatives

based on these pseudo-labels by identifying classes *not* in each image (by using detector confidence scores), and leverage pseudo-negatives in our fine-tuning. Notably, we improve over Detic's standard fine-tuning by 1.15 mAP (16.09 mAP → 17.24 mAP).

Multi-Modal Prompting. We observe that Multi-Modal Prompting (`MQ-GLIP`) achieves the best performance (21.42 mAP) out of all open-source methods in Table 4.3. We attribute this to its large pre-trained dataset, bigger backbone (SWIN-L) and multi-modal prompts used during inference. Notably, the benefit of multi-modal prompts can be seen by comparing `MQ-GLIP` (21.42 mAP) against `MQ-GLIP-Image` (14.07 mAP), which uses visual prompting and `MQ-GLIP-Text` (17.01 mAP), which uses language prompting. Interestingly, `MQ-GLIP` does not require gradient-based fine-tuning, which differs from all existing conventional few-shot methods. Therefore, we posit that future few-shot methods should further explore in-context learning. Just as multi-modal annotator instructions aid human annotator alignment, we find that multi-modal prompting significantly improves VLM concept alignment.

Multi-Modal Chat Agents. Given the strong performance of GPT-4o for general visual understanding, we repurpose it for our task by prompting the model to re-classify image patches from Detic's RPN. Specifically, we ask GPT-4o to predict a class and confidence for each image crop. Surprisingly, we observe reasonable performance (9.95 mAP) despite GPT-4o not being trained as an object detector, emphasizing the importance of the scale of pre-training data. We explore iterative prediction refinement in the supplement.

CVPR 2024 Challenge. Finally, we highlight our top three submissions (out of six participants) from the inaugural Foundational FSOD challenge. Notably all top performers beat our baselines, with the winning team achieving 45.35 AP! We discuss more details in the supplement.

4.5 Analysis of Iconic Few-Shot Images

The specific examples used during few-shot fine-tuning significantly impacts target class performance [43]. However, prior work constructs few-shot splits by randomly sampling K examples per class. In contrast, when creating annotator *instructions*, selecting the right examples to "align" human annotators [3] to subtle aspects of the target concept is carefully considered. To more closely match VLM *concept alignment*

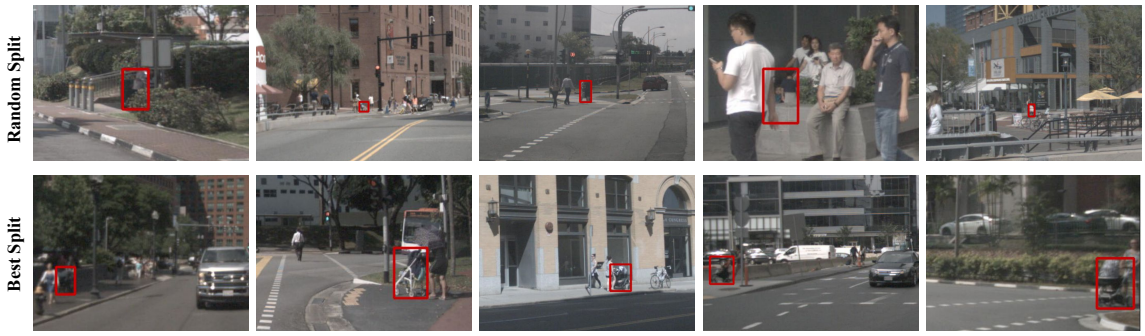

Figure 4.1: **Visualizing Random and Best Split**. In the top row, we visualize the 5-shot training examples of **strollers** from a *random split*. Similarly, we visualize the 5-shot training examples from the *best split* in the bottom row. We observe that strollers in the *random split* are often occluded, small in size and blurry, making few-shot learning harder. On the other hand, the *best split* examples are larger, have better visual quality and are relatively un-occluded. This visual difference directly translates into better few-shot performance. We achieve **13.09 Stroller AP** for the *random split* and **18.54 Stroller AP** for the *best split*. We show a more comprehensive evaluation in Table 4.4.

with human annotator alignment, we design a simple algorithm to construct the best K-shot split for fine-tuning. This allows us to understand which examples are most informative and measure an upper bound in performance.

We construct our *best split* by picking examples corresponding to the best class-wise performance, based on the evaluation of each split on a held-out validation set. For instance, out of 3 random splits for the 5-shot task, one may pick `car` examples from split 1, `bicycle` from split 3 and `debris` from split 2. In Table 4.4, we observe

Table 4.4: **Random Split vs "Best" Split**. We construct the "best" split by selecting per-class few-shot examples that lead to the highest performance on a held-out set. Unsurprisingly, the best split performs better than any random split, especially for very limited data settings (e.g. 5-shot detection). This evaluation setting closely mimics how human annotators are "aligned" to target concepts, since annotator guides are constructed using hand-picked iconic visual examples.

Approach	Average Precision (AP)			
	`All`	`Many`	`Medium`	`Few`
Detic (Zero-Shot) [62]	14.40	25.83	16.59	2.32
Detic w/ Federated Fine-Tuning *(5-shots, Random Split)*	16.58	27.12	19.71	4.13
Detic w/ Federated Fine-Tuning *(5-shots, Best Split)*	**18.30**	**28.66**	**21.81**	**5.56**
Detic w/ Federated Fine-Tuning *(10-shots, Random Split)*	17.24	28.07	20.71	4.18
Detic w/ Federated Fine-Tuning *(10-shots, Best Split)*	**18.24**	**28.63**	**22.00**	**5.19**
Detic w/ Federated Fine-Tuning *(30-shots, Random Split)*	18.64	**29.13**	22.44	5.46
Detic w/ Federated Fine-Tuning *(30-shots, Best Split)*	**18.75**	28.07	**23.18**	**5.81**

that the *best-split* performance is always better than its random counterpart. As expected, the choice of examples in 5-shot case is more important than the 30-shot case (1.72 AP difference for 5-shot vs 0.11 AP for 30-shots). We visualize the difference in the splits for `stroller`s in nuImages (cf. Figure 4.1). Unsurprisingly, iconic examples are large and unoccluded.

4.6 Limitations and Future Work

Despite using VLMs pre-trained on large-scale datasets, we find that performance for rare categories (defined by the cardinality of each class in the original dataset) is considerably lower than for common classes. We posit that VLMs are pre-trained with imbalanced data which includes many examples of common categories like `truck` but few examples of rare categories like `stroller`. Our work does not explicitly improve detection performance on rare classes. Interestingly, since VLMs like Detic [62], GLIP [29], and GroundingDINO [31] are trained with different data sources, each model has dramatically different zero-shot performance on novel categories like `stroller`. Ensembling predictions from different VLMs may yield better detection accuracy for rare categories. In addition, although our work motivates the use of rich textual descriptions found in instructions for multi-modal alignment, our current results use only nouns (class names and synonyms) as text prompts.

Benchmarking in the Era of Foundation Models. Although we argue that pre-training on large-scale data will be the key enabler for generalization to the open-world, understanding how to appropriately benchmark such methods remains challenging. It is readily accepted that in order to accurately evaluate generalization, one should not train on test data. However, it is difficult to guarantee that foundation models have never seen our specific test data. We address this in our challenge by explicitly prohibiting participants from training on nuImages (and nuScenes). However, should we allow participants to train on similar in-domain data (e.g., other autonomous vehicle datasets such as Argoverse [46])? We argue 'yes'! With enough scale, novel test examples may still be similar to the training set.

Out-of-Domain Benchmarks. Another way to address benchmarking is to collect test scenarios that are *designed* to be dissimilar from internet images. For example, out-of-domain images may include medical data (though foundational

performance is still surprisingly effective [45]). We admittedly did not take this route, since urban imagery is similar to images found online and arguably many applications of interest fall under this category. Moreover, there exist ample opportunity for technical innovation in this setting (as suggested by our CVPR 2024 challenge results!). Alternatively, one can manually collect and sequester images that will never be released on the internet. Since ensuring privacy may itself be challenging, yet another approach is to leverage the continual learning paradigm, where new test sets are continually constructed over time.

Comparing Models. Fairly comparing foundation models requires careful consideration. Although accuracy is a valuable metric, it is intrinsically tied to the scale of pre-training data and model architecture. Notably, the LLM community already ranks models via a Pareto frontier of accuracy vs. parameter count. We advocate for a similar approach for Foundational FSOD that considers backbone architecture (e.g. ResNet-50 vs. Swin-B) and pre-training datasets (e.g. CC4M, GoldG, LVIS).

Chapter 5

Conclusion

We revisit few-shot object detection (FSOD) with vision-language models (VLMs) and find that zero-shot inference from web-scale VLMs significantly outperforms leading FSOD methods. However, such foundational models do not fully address few shot recognition because of the *concept alignment* problem; particular concepts in target applications may be different than their use on web-scale datasets. Just as human annotators require concept alignment via multi-modal text and visual examples, we argue that VLMs should be aligned with such few-shot data, formalizing the problem of Foundational FSOD.

Appendix A

Appendix

A.1 Baseline Implementation Details

We repurpose nuImages (CC BY-NC-SA 4.0) for all few-shot experiments in the main paper. We evaluate detection performance using 1600×900 images across 18 classes for all models tested. We create three random splits for each of $K = \{5, 10, 30\}$-shots following the data creation process from [43] and report results averaged across these three seeds. Our test-set is a subset of the (densely annotated) nuImages val-set. We construct our test-set to only include validation images which have at least one annotation from the `Few` or `Medium` cohorts (cf. Fig A.1). We train all baselines with one RTX 3090 GPU. Our baseline code is available on GitHub and dataset splits are available on HuggingFace.

Prompt Engineering: We leverage rich text descriptions provided by the annotator instructions to select synonyms for each nuImages class. We manually identify the best performing synonyms in Table A.1. At test time, we compute the average text embedding of all synonyms to improve classification accuracy.

Language Prompt Tuning We train GLIP (SWIN-L backbone) for our prompt tuning experiments for 60 epochs with a learning rate of 0.025, batch size of 4, and weight decay of 0.25.

Federated Fine-tuning. We use Detic (Swin-B backbone) pre-trained on `LVIS + COCO` and ImageNet-21k data for our federated fine-tuning experiments (described

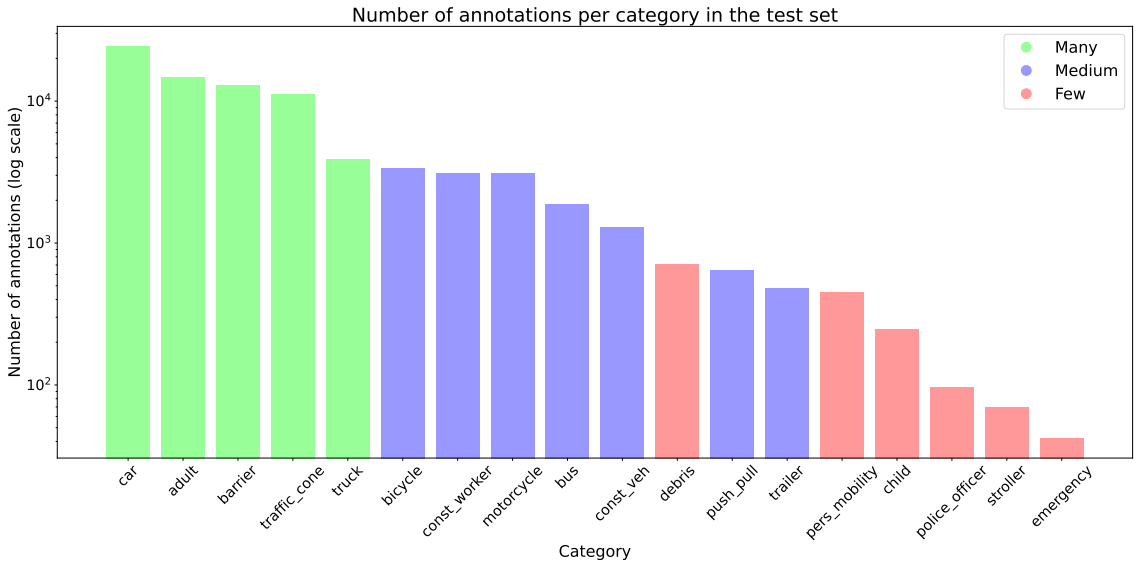

Figure A.1: We visualize the distribution of classes in out test-set compared to the cardinalities of classes in the full nuImages val-set. Notably, our sub-sampling strategy of selecting validation images that have at least one annotation from `medium` or `few` classes does not significantly alter the true distribution.

in detail in the next section). We use a batch size of 8 and an AdamW optimizer with learning rate of $3.75e-6$. We fine-tune this model for 8000 iterations on nuImages. We sample 6 categories for each training image, i.e $|S| = 6$ for the FedLoss and InvFedLoss experiments. We derive negatives from pseudolabels with atleast 20% confidence for the Psuedo-Negative experiment.

Multi-Modal Prompting. We use MQDet (`text-only, vision-only, text + vision` for our in-context learning baselines. Unlike the original code base, we tokenize our few shot examples instead of using random queries. Note that zero-shot results for MQ-GLIP-Text and GLIP-L are the same since these models are identical.

A.2 CVPR 2024 Competition Details.

Our inaugural Foundational FSOD competition (hosted on Eval AI) received submissions from eight teams (some submissions are private) around the world. We present a ranked list of participants at the close of our competition on June 7th AOE in Table A.2. Notably, three teams were able to beat our MQ-GLIP baseline. Unfortunately,

Table A.1: **Synonyms used for Prompt Engineering.** We manually inspect the nuImages annotator instructions to derive a set of synonyms to improve classification accuracy.

Original Classes	Class Names with Synonyms
car	car
truck	truck, pick-up, lorry, semi-tractor
construction_vehicle	construction_vehicle, crane
bus	bus, bendy_bus, rigid_bus
trailer	trailer
emergency	emergency, ambulance, police_car, police_motorcycle
motorcycle	motorcycle
bicycle	bicycle
adult	adult, person
child	child
police_officer	police_officer
construction_worker	construction_worker
personal_mobility	personal_mobility, skateboard, segway, scooter
stroller	stroller
pushable_pullable	pushable_pullable, wheel_barrow, garbage_bin, cart
barrier	barrier, K-rail, fence, bollard, guard_rail
traffic_cone	traffic_cone
debris	debris, trash_bag

the top performing team wasn't willing to publicly share details about their method. We summarize contributions from the other two top teams below.

NJUST KMG presents a method leveraging a vision-language model (VLM) enhanced with a multimodal large language model (MM-LLM) to improve Few-Shot Object Detection (FSOD). To address the challenge of misalignment between VLMs and target concepts, authors propose generating descriptive referential expressions for each category using MM-LLM. This involves annotating images with bounding boxes, prompting ChatGPT to provide descriptive terms for each object, and then creating multiple referential expressions by randomly combining these terms. The VLMs then select the best referential expression for each category by matching the maximum Intersection over Union (IoU) in the training set, and these expressions are used to generate pseudo-labels for all training images, which are combined with original labeled data to fine-tune the VLM. This iterative process of pseudo-label

generation and optimization significantly enhances the VLM's performance.

ZJYD CXY Vision proposes Instruction DINO (ISD), a DETR-based detector architecture and incorporates early fusion of image and text information, using a Swin-L visual backbone and EVA02-CLIP-L text encoder. Pre-training involves two stages using various datasets, transforming grounding data into single-object descriptions with QWen Max. For few-shot fine-tuning, the model adopts a flexible training format and uses VLMs like CLIP, TAP, and LLava for negative sample generation, finding that prompt tuning and text encoder fine-tuning generalize better than visual encoder fine-tuning. The final fine-tuning method combines prompt tuning and negative sampling, significantly improving mAP. To address sparse annotations, the visual encoder is initially fine-tuned to generate pseudo-label annotations, which are then used to complete training with prompt tuning.

A.3 Iterative Prompting with Multi-Modal Chat Assistants

Typically, clients provide human annotators with a set of multi-modal instructions and a corpus of unlabeled data for annotation. Annotators start by first labeling a small subset of the data for review by the client, who acts as a domain expert and provides feedback on erroneous annotations (highlighting *concept misalignment*). Annotators use this feedback to annotate another subset of the data. This iterative process continues until the client is satisfied with the annotators' ability to accurately label the entire dataset.

Table A.2: **CVPR 2024 Foundational FSOD Competition Results**.

Team Name	Average Precision (AP)			
	`All`	`Many`	`Medium`	`Few`
PHP_hhh	45.35	64.25	53.43	20.19
NJUST KMG	32.56	50.21	34.87	15.16
zjyd_cxy_vision	31.57	46.59	33.32	17.03
Baseline (MQ-GLIP)	21.51	32.25	23.35	10.41
team_anon	17.36	25.29	21.93	5.42
youyouqiu	13.16	11.29	19.20	7.68
zhao	11.38	11.16	16.76	5.30
zjdcxy	7.80	5.44	13.43	3.20

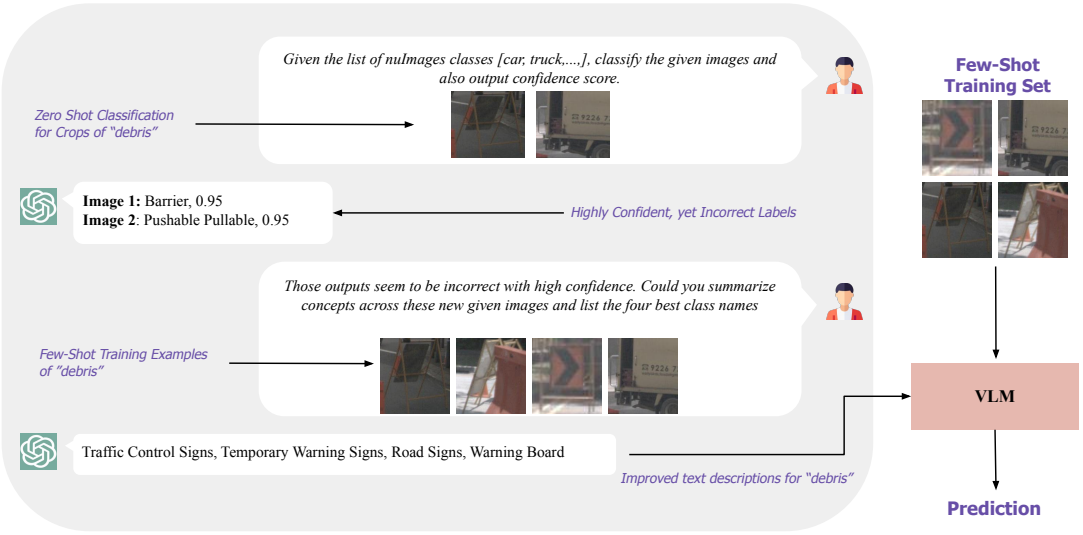

Figure A.2: **Iteratively Prompting ChatGPT.** Despite its large-scale pre-training, multi-modal models like ChatGPT-4o also suffers from concept alignment. Specifically, GPT-4o makes highly confident but incorrect predictions for `debris`. We propose an iterative prompting strategy to better align the model to a target concept. Given a few visual examples per-class from the training-set, we query ChatGPT to use its "web-scale knowledge" to generate text descriptions. We then augment the input to MQDet to incorporate this additional context for zero-shot evaluation.

As shown in Figure A.2, we explore the idea of iteratively prompting multi-modal chat assistants like ChatGPT to mimic the real-world workflow of human annotators. We start by asking GPT-4o to classify image crops of `debris` (derived from the few-shot training split). Notably, GPT-4o incorrectly classifies these training examples with high confidence. Therefore, we prompt GPT-4o to generate its own text descriptions of the few-shot examples according to its "web-scale knowledge". Finally, we use the class names, generated text descriptions for `debris`, and few-shot visual examples to MQDet to predict instances of `debris` in the test-set.

We find that prompting MQDet with class names, ChatGPT generated text descriptions, and few-shot visual examples improves performance by 0.67% (21.42 mAP $\rightarrow$ 22.09 mAP) over the baseline. Interestingly, although `debris` does not change when prompted with generated text descriptions, `pushable pullable` (3.6 AP $\rightarrow$ 15.29 AP) and `barrier` (11.6 AP $\rightarrow$ 15.31 AP) accuracy improve significantly. We posit that this improvement is due to the reduction in confusion (or the over-confident

incorrect predictions) between `debris` and `pushable-pullable` (and `barrier`). Surprisingly, one of the top submissions to our CVPR challenge also use ChatGPT to generate meaningful text descriptions to improve detection *concept alignment*.

A.4 Analysis of Federated Fine-Tuning

Prior works follow the K-shot dataset creation process established by [43]. Importantly, each image in the dataset is exhaustively annotated for a subset of all classes. Recall, a federated dataset is also comprised of images that are exhaustively annotated for a specific category. This suggests that we can leverage existing insights about federated datasets [14, 61] to train better few-shot object detectors.

Fine-Tuning with FedLoss. We fine-tune Detic with Federated Loss (FedLoss) [61] using a subset S of classes C for each training image. Specifically, we use a binary cross-entropy loss on all classes in S and ignore classes outside of S during training. S is comprised of the ground-truth annotation class along with randomly sampled negative classes for each image. We sample these negative classes in proportion to their square-root frequency in the training set. We find that probablistically sampling negatives rather than labeling all unannotated classes as negatives improves fine-tuning results, reliably beating zero-shot performance. Importantly, although FedLoss has been explored in the context of long-tailed detection, applying it to FSOD provides considerable performance improvements, reaffirming that FSOD benchmarks are actually federated datasets.

Fine-Tuning with Pseudo-Negative Federated Loss (Ours). Despite the effectiveness of FedLoss, probablistically sampling negatives using dataset-wide statistics is sub-optimal because it does not consider the content of each image. We can improve the accuracy of sampled negatives with pseudo-labels to determine which classes are likely *not* in a particular image. If the maximal score for any class prediction is less than a threshold, we consider this class to be a negative. Using zero-shot model predictions to identify pseudo-negatives yields better results than simply using dataset-wide statistics. We find that this strategy works the best. We present pseudo-code in Alg. A.1. All federated fine-tuning results in the main paper are trained with psuedo-negative federated loss.

Code Listing A.1: Psuedo-Negative Federated Loss

```
# Inputs

# img: Randomly Sampled Image
# all_classes: All Classes in Dataset
# gt: Ground Truth Annotations for img
# gt_classes: List of Classes in gt

# Outputs
# loss: Psuedo-Negative Federated Loss

# Functions
# filter: Returns All Predictions w/ Confidence > Threshold
# get_neg: Returns List of Classes Not In Pseudo-Positives
# or: Set Union Operation
# BCE: Binary Cross Entropy Loss

#Step 1: Compute Predictions and Filter by Confidence
pred = Detector(img) # predictions
pseudo_pos = filter(pred, thresh=0.2)

#Step 2: Get Pseudo-Negatives for Image
neg_classes = get_neg(pseudo_pos, all_classes)
select_classes = or(neg_classes, gt_classes)

#Step 3: Compute Deterministic Federated Loss w/ Pseudo-Negatives
loss = 0
for cls in select_classes:
    pred_cls = pred[cls] #predictions for cls
    gt_cls = gt[cls] #ground-truth for cls

    loss += BCE(pred_cls, gt_cls)

return loss
```

Oracle Performance Analysis. We empirically validate the effectiveness of our pseudo-negative federated loss by computing the upper bound performance when given access to ground-truth negatives and exhaustive annotations for the few-shot data split. Recall, nuImages is exhaustively annotated, but is repurposed for Foundational

Table A.3: **Analysis of nuImages Upper Bound Performance**. We compare the accuracy of our proposed approach against upper bounds computed for the FSOD task. Our pseudo-negatives strategy approaches the performance of using ground-truth negatives, demonstrating that pesudo-labels can provide a reliable signal about negatives, especially across classes with `many` and `medium` examples. The performance gap between our best method and exhaustive annotations can be attributed to the large number of additional annotations, particularly for classes with `many` and `medium` examples. Compared to the baseline (14.3 AP), our approach (16.7 AP) closes the gap to the (18.5 AP) upper-bound by over 50%.

Approach	**10 Shots**: Average Precision (AP)			
	`All`	`Many`	`Medium`	`Few`
Detic (Zero-Shot) [62]	14.26	27.28	15.15	2.36
+ Standard Fine-Tuning	15.53	26.01	18.02	3.88
w/ FedLoss	15.57	27.20	18.13	2.89
w/ Pseudo-Negatives	**16.67**	**29.15**	**18.71**	**3.90**
w/ True Negatives (*Oracle*)	16.99	29.60	18.94	4.21
w/ Exhaustive Annotations (*Oracle*)	18.51	33.51	20.30	3.93

FSOD.

To compute the set of ground-truth negatives for each image, we use exhaustive ground-truth annotations to determine which categories are not present. Training with ground-truth negatives provides an upper bound on our pseudo-negatives experiment. Next, we train using exhaustive ground-truth annotations to provide an upper bound for the specific set of images used during training. In addition, this experiment highlights the performance gap between having exhaustive negatives and exhaustive annotations.

Table A.3 shows that using pseudo-negatives nearly matches the true negative upper bound (16.67 AP vs 16.99 AP). This demonstrates that we are able to reliably estimate negatives in an image, alleviating the problem of learning with sparse annotations. Training with exhaustive annotations yields significantly better results for `many` and `medium` classes. This is unsurprising because the 10-shot FSOD benchmark includes 10 car annotations, while the exhaustively annotated set includes over 550 car annotations!

Despite strong performance on classes with `many` and `medium`, the upper bound for classes with `few` examples remains low (4.21 AP and 3.93 AP). Given the success of training with pseudo-negatives, a natural next-step is to train with pseudo-positives. Our preliminary results suggest that incorporating pseudo-positives does not provide significant improvement over simply training with pseudo-negatives. We posit that

training with incorrect pseudo-positives may incur a higher penalty than training with incorrect pseudo-negatives. This is a promising direction for future work.

A.5 Impact of Box-Level Supervision for Foundational FSOD

We evaluate the importance of using bounding-box supervised data in pre-training. Unlike Detic, which trains on box-supervised data from `LVIS, COCO` and image-text data from `ImageNet21k`, RegionCLIP[58] only pre-trains on image-text pairs from the Conceptual Captions (CC3M) dataset [40].

We report RegionCLIP's zero-shot and fine-tuning performance on nuImages averaged over 3 random splits in Table A.4. Detic zero-shot outperforms RegionCLIP zero-shot by $\sim$ 12 AP (14.26 vs 2.34). While fine-tuning RegionCLIP improves overall performance, Detic achieves higher accuracy for $K = \{5, 10, 30\}$ shots. This highlights the importance of supervision type (e.g. box-supervised data) and data scale used for pre-training.

Next, we conduct further analysis to diagnose why RegionCLIP zero-shot inference performs so poorly on nuImages (Table A.5). RegionCLIP relies on an RPN pre-trained on box-supervised data like `LVIS-base` to extract regions for pre-training. Notably, RegionCLIP (w/ `LVIS-RPN`: 2.34 AP) suffers from poor foreground-vs-background classification compared to Detic. We validate this hypothesis by evaluating RegionCLIP (w/ `GT-RPN)` to measure classification performance. Surprisingly, RegionCLIP achieves significantly higher accuracy (26.44 AP), confirming that RegionCLIP struggles to distinguish between foreground and background in nuImages. This observation highlights the challenge of working with nuImages categories, further motivating our Foundational FSOD benchmark.

Lastly, we evaluate RegionCLIP's performance with `Detic-RPN`. Notably, we observe that the performance improves over RegionCLIP w/ `LVIS-RPN` demonstrating that reducing the number of false positive proposals yields better performance. Furthermore, we filter out low confidence Detic proposals , i.e < 0.5 objectness score (w/ `Detic-RPN, 0.5`) and find that this doubles RegionCLIP's zero-shot performance to 7.64 AP.

Table A.4: **RegionCLIP Experiments**. RegionCLIP zero-shot inference performs much worse than Detic. While fine-tuning improves RegionCLIP's performance, it still lags far behind Detic. We posit that this performance difference can be attributed to Detic's box-supervised pre-training and use of language cues from CLIP embeddings.

Approach	Average Precision (AP)			
	`All`	`Many`	`Medium`	`Few`
RegionCLIP (*Zero-Shot*) [58]	2.34	3.33	3.45	0.22
Detic (*Zero-Shot*) [62]	**14.26**	**27.28**	**15.15**	**2.36**
RegionCLIP (*Fine-Tuning, 5 shots*) [58]	3.61	6.20	4.63	0.26
Detic (*Fine-Tuning, 5 shots*) [62]	**14.50**	**24.09**	**16.90**	**3.70**
RegionCLIP (*Fine-Tuning, 10 shots*) [58]	3.58	6.10	4.65	0.24
Detic (*Fine-Tuning, 10 shots*) [62]	**15.28**	**26.93**	**18.00**	**3.27**
RegionCLIP (*Fine-Tuning, 30 shots*) [58]	3.57	6.13	4.61	0.22
Detic (*Fine-Tuning, 30 shots*) [62]	**16.65**	**27.45**	**19.46**	**4.02**

Table A.5: **Diagnosing RegionCLIP's Poor Zero-Shot Performance**. RegionCLIP's zero-shot performance lags far behind Detic. Using RegionCLIP's classifier on ground-truth region proposals yields high performance, suggesting that RegionCLIP struggles to accurately distinguish between foreground-vs-background.

Approach	Average Precision (AP)			
	`All`	`Many`	`Medium`	`Few`
Detic *(Zero-Shot)* [62]	14.26	27.28	15.15	2.36
GroundingDINO (Zero-Shot) [31]	11.44	17.42	14.08	3.38
RegionCLIP *(Zero-Shot)* w/ `LVIS-RPN` [58]	2.34	3.33	3.45	0.22
RegionCLIP *(Zero-Shot)* w/ `Detic-RPN` [58]	3.79	6.68	4.01	1.12
RegionCLIP *(Zero-Shot)* w/ `Detic-RPN, 0.5` [58]	7.64	12.81	8.88	1.88
RegionCLIP *(Zero-Shot)* w/ `GT-RPN` [58]	26.44	45.33	32.25	3.92

A.6 NuImages Annotator Instructions

We present an example of the nuImages annotator instructions below. Notably, such annotator instructons are naturally few-shot (e.g. providing a few visual and textual examples describing the target concept), multi-modal, and contain both positive and negative examples. Our proposed Foundational FSOD benchmark, and pseudo-negative federated loss facilitate future work in leveraging rich annotator descriptions, allowing us to "align" VLMs much like how annotators must be "aligned" to subtle aspects of the target class.

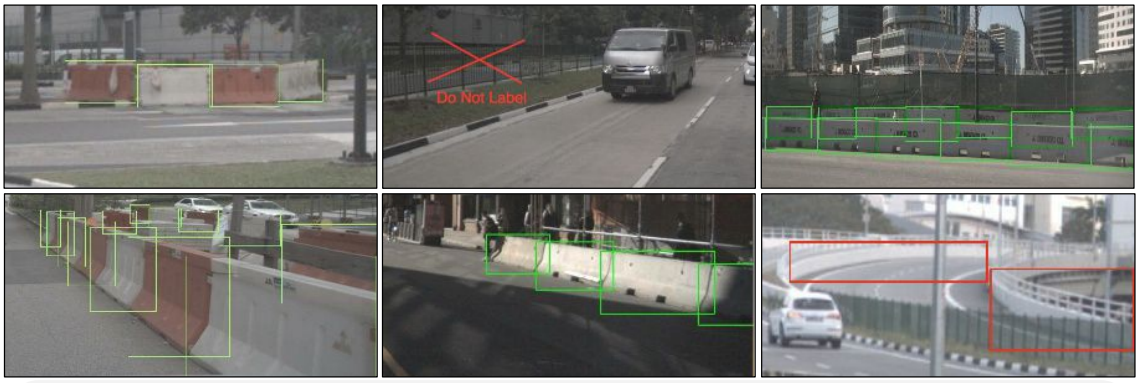

Figure A.3: **NuImages Annotator Instructions.** We include the **multi-modal** annotator instructions `barrier`. Our proposed setup allows FSOD methods to learn such multi-modal examples, similar to how human annotators are taught the labeling policy. Importantly, annotators can also be provided with negative examples (in red) for classes, i.e what **NOT** to label for a certain class. Crucially, our proposed fine-tuning with pseudo-negatives can easily accommodate such negative examples within the proposed setup.

A.7 Empirical Analysis of Baselines (5-Shots and 30-Shots)

We evaluate all baselines for the nuImages experiments with 5-shots and 30-shots in Tables A.6 and 4.3 respectively. We find that trends from the main paper hold. Notably, MQ-GLIP with-multi-modal prompting performs the best. However, we find that adding more examples (e.g. MQ-GLIP 5-shot vs. MQ-GLIP 30-shot) doesn't seem to help in-context learning based methods nearly as much as gradient-based fine-tuning approaches.

Table A.6: **Empirical Analysis of Baselines (5-Shots) on nuImages**.

Approach	Backbone	Pre-Train Data	Average Precision (AP) All	Many	Med	Few
Zero-Shot Detection						
RegionCLIP [58]	RN50	CC3M	2.50	3.20	3.80	0.40
Detic [62]	SWIN-B	LVIS, COCO, IN-21K	14.40	25.83	16.59	2.32
GroundingDINO [31]	SWIN-T	Objects365,GoldG,Cap4M	12.05	17.29	15.45	3.72
GLIP [29]	SWIN-L	FourODs,GoldG,Cap24M	17.01	23.36	19.86	8.40
MQ-GLIP-Text [53]	SWIN-L	Objects365,FourODs,GoldG,Cap24M	17.01	23.36	19.85	8.41
Prompt Engineering						
Detic [62]	SWIN-B	LVIS, COCO, IN-21K	14.92	26.48	17.29	2.53
GLIP [29]	SWIN-L	FourODs,GoldG,Cap24M	17.15	23.82	19.36	9.02
Standard Fine-Tuning						
RegionCLIP [58]	RN50	CC3M	3.84	6.13	5.07	0.49
Detic [62]	SWIN-B	LVIS, COCO, IN-21K	15.12	22.74	18.99	4.25
Federated Fine-Tuning (Ours)						
Detic [62]	SWIN-B	LVIS, COCO, IN-21K	16.58	27.12	19.71	4.13
Detic [62] w/ Prompt Engineering	SWIN-B	LVIS, COCO, IN-21K	16.96	27.89	19.94	4.37
Language Prompt Tuning						
GLIP [29]	SWIN-L	FourODs,GoldG,Cap24M	17.79	21.07	22.87	9.12
Visual Prompting						
MQ-GLIP-Image [53]	SWIN-L	Objects365,FourODs,GoldG,Cap24M	13.42	23.05	15.00	3.54
Multi-Modal Prompting						
MQ-GLIP [53]	SWIN-L	Objects365,FourODs,GoldG,Cap24M	**21.45**	**32.23**	**23.31**	**10.30**
Multi-Modal Chat Assistants						
GPT-4o Zero-Shot Classification [1]	*Private*	*Private*	9.95	16.81	12.11	1.71

A.8 Foundational FSOD with LVIS

Although we use nuImages for Foundational FSOD for benchmarking in the main paper and in our competition, other datasets can still be evaluated under this framework. We include benchmarking results for LVIS below. LVIS [14] re-annotates COCO images using 1,230 fine-grained classes, which are divided into frequent, common and rare based on the cardinality of each class. Frequent and common classes are combined to form `LVIS-base` and is used for pre-training. Rare classes are used for `LVIS-novel`. Following [33, 43], we benchmark with LVIS v0.5 on publicly released data splits and report performance averaged across 3 splits for frequent, common, and rare groups (AP_f, AP_c, AP_r) on the LVIS val-set.

As shown in Table A.8, Detic outperforms all recent FSOD baselines including DiGeo [33] by about ∼6 points on AP_c and AP_f and achieves 16.3 AP_r without ever seeing any rare class data (e.g by prompting Detic (`Base` Only) with the rare class names). Importantly, these performance improvements can be attributed to Detic's CLIP-based classifier, which uses CLIP text embeddings corresponding to class names. Such embeddings are a result of large-scale pre-training, which we can effectively

Table A.7: **Empirical Analysis of Baselines (30-Shots) on nuImages**.

Approach	Backbone	Pre-Train Data	Average Precision (AP) All	Many	Med	Few
Zero-Shot Detection						
RegionCLIP [58]	RN50	CC3M	2.50	3.20	3.80	0.40
Detic [62]	SWIN-B	LVIS, COCO, IN-21K	14.40	25.83	16.59	2.32
GroundingDINO [31]	SWIN-T	Objects365,GoldG,Cap4M	12.05	17.29	15.45	3.72
GLIP [29]	SWIN-L	FourODs,GoldG,Cap24M	17.01	23.36	19.86	8.40
MQ-GLIP-Text [53]	SWIN-L	Objects365,FourODs,GoldG,Cap24M	17.01	23.36	19.85	8.41
Prompt Engineering						
Detic [62]	SWIN-B	LVIS, COCO, IN-21K	14.92	26.48	17.29	2.53
GLIP [29]	SWIN-L	FourODs,GoldG,Cap24M	17.15	23.82	19.36	9.02
Standard Fine-Tuning						
RegionCLIP [58]	RN50	CC3M	3.87	6.05	5.14	0.57
Detic [62]	SWIN-B	LVIS, COCO, IN-21K	17.22	25.98	21.64	4.78
Federated Fine-Tuning (Ours)						
Detic [62]	SWIN-B	LVIS, COCO, IN-21K	18.64	29.13	22.44	5.46
Detic [62] w/ Prompt Engineering	SWIN-B	LVIS, COCO, IN-21K	18.67	29.13	22.43	5.57
Language Prompt Tuning						
GLIP [29]	SWIN-L	FourODs,GoldG,Cap24M	20.73	24.95	**25.60**	**11.54**
Visual Prompting						
MQ-GLIP-Image [53]	SWIN-L	Objects365,FourODs,GoldG,Cap24M	14.26	24.55	16.73	2.79
Multi-Modal Prompting						
MQ-GLIP [53]	SWIN-L	Objects365,FourODs,GoldG,Cap24M	**21.40**	**32.08**	23.31	10.27
Multi-Modal Chat Assistants						
GPT-4o Zero-Shot Classification [1]	*Private*	*Private*	9.95	16.81	12.11	1.71

leverage for the few-shot task. This highlights the role of language in data-constrained settings.

Further, fine-tuning Detic with pseudo-negatives improves overall performance by 1.6 AP (30.0 vs 31.6) over naive fine-tuning. To contextualize the improvement in performance, we note that between TFA (ICML 2020) and DiGeo (CVPR 2023), the community improved on LVIS FSOD by only 0.5 AP (cf. Table A.8). Finally, we note that simply replacing the ResNet-50 backbone with a Swin-B transformer yields a sizeable 12.8 AP improvement for rare classes (19.8 vs. 32.6).

We present fine-tuning results for different variants of Detic on the LVIS 10-shot dataset. Following the standard FSOD protocol, we pre-train Detic on `LVIS-base` (e.g. frequent and common classes) and fine-tune on 10-shots from each class in `LVIS-base` and `LVIS-novel`. Importantly, this means that only results for AP_r are indicative of true few-shot performance. First, we find that naively fine-tuning Detic on `Base + Novel` yields lower performance for AP_f and AP_r. Intuitively, this suggests that ignoring the federated nature of FSOD datasets (e.g. by following the standard practice of assuming common classes are negatives for rare class federated datasets) hurts common class performance (cf. Table A.8). Importantly, simply training with

FedLoss significantly improves over naive fine-tuning, increasing AP_r by 1.9% (15.5 vs. 17.4) and 3.7% (26.7 vs. 30.4) for the ResNet-50 and Swin backbones respectively. Further, leveraging our proposed negative pseudo-labeling strategy provides further improvements over the naive federated loss, increasing AP_r by another 2.4% (17.4 vs. 19.8) and 3.7% (30.4 vs. 32.6) for the ResNet-50 and Swin backbones respectively. Similar to nuImages, we find that multi-modal prompting with MQ-GLIP performs the best of all baselines tested, significantly improving over MQ-GLIP-Text and MQ-GLIP-Image. We attribute MQ-GLIP's strong performance to its bigger backbone and significantly larger pre-training dataset.

LVIS v0.5 Detic Experiment Details. We select Detic with a Resnet-50 backbone for fair comparison with prior work. We pre-train Detic on `LVIS-base` for $90k$ iterations with a batch size of 32 using an AdamW optimizer and a learning rate of $2e-3$. All images are resized to 640×640 and we also enable Repeat Factor Sampling [14]. Following [43], we sample *up to* 10 shots for each class in LVIS (since all classes may not have 10 examples). We use a batch size of 32, learning rate of $2.5e-5$ for $46k$ iterations. We do not use Repeat Factor Sampling for fine-tuning. We sample 50 categories for each training image, i.e $|S| = 50$ for the FedLoss experiments. We derive negatives from pseudolabels with atleast 20% confidence for the Psuedo-Negative experiment.

Table A.8: **LVIS Foundational FSOD Performance**. We present fine-tuning results for different variants of Detic on the LVIS 10-shot dataset. We follow the standard FSOD setup and pre-train Detic on `LVIS-base` for fair comparison with prior work. Detic pre-trained only on `LVIS-base` outperforms specialized methods like TFA and DiGeo by ∼6 AP, *without fine-tuning* on rare classes. Since we keep the model backbone (ResNet-50) and pre-training data same for all methods, these performance improvements can be attributed to Detic's CLIP-based classifier. This demonstrates that concept leakage through language significantly improve FSOD, and leveraging language cues should be embraced in data constrained settings. Naively fine-tuning Detic yields a performance drop of AP_f and AP_c because treating common classes as negatives in rare category federated datasets hurts performance. Instead, we find that embracing the federated nature of FSOD datasets provides consistent improvements in fine-tuning (30.0 vs. 30.8 for ResNet-50). Further, pseudo-labeling negatives in each image provides a modest improvement (30.8 vs. 31.6 for ResNet-50). Similar trends hold for the Swin backbone.

Approach	10-shots			
	AP	AP_f	AP_c	AP_r
ResNet-50 Backbone				
TFA w/ fc [43]	24.1	27.9	23.9	14.9
TFA w/ cos [43]	24.4	27.7	24.3	16.9
DiGeo [33]	24.9	28.5	24.6	17.3
Detic (`Base` Only) [62]	30.0	34.4	30.8	16.3
+ Fine-Tuning (`Base` + `Novel`)	30.0	33.2	31.9	15.5
w/ FedLoss	30.8	33.9	32.7	17.4
w/ Pseudo-Negatives	**31.6**	**34.8**	**32.8**	**19.8**
Swin Backbone				
Detic (`Base` Only, SWIN-B) [62]	35.2	38.7	36.8	21.4
+ Fine-Tuning (`Base` + `Novel`)	35.9	37.1	37.8	26.7
w/ FedLoss	36.5	36.7	38.3	30.4
w/ Pseudo-Negatives	37.2	37.7	38.2	32.6
MQ-GLIP-Text *(SWIN-L)*	35.8	40.2	33.1	33.0
MQ-GLIP-Image *(SWIN-L)*	28.8	33.0	26.6	25.1
MQ-GLIP *(SWIN-L)*	**43.4**	**46.4**	**41.8**	**40.1**

Bibliography

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 4.3, A.6, A.7

[2] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. (document), 1.1, 3.1, 3.2, 4.1

[3] Nadine Chang, Francesco Ferroni, Michael J Tarr, Martial Hebert, and Deva Ramanan. Thinking like an annotator: Generation of dataset labeling instructions. *arXiv preprint arXiv:2306.14035*, 2023. 1, 4.5

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2.3

[5] Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric P Xing, and Zhiting Hu. Rlprompt: Optimizing discrete text prompts with reinforcement learning. *arXiv preprint arXiv:2205.12548*, 2022. 2.3

[6] Yu Du, Fangyun Wei, Zihe Zhang, Miaojing Shi, Yue Gao, and Guoqi Li. Learning to prompt for open-vocabulary object detection with vision-language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14084–14093, 2022. 2.2

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010. 1, 3.1

[8] Qi Fan, Wei Zhuo, Chi-Keung Tang, and Yu-Wing Tai. Few-shot object detection with attention-rpn and multi-relation detector. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4013–4022, 2020.

2.1

[9] Zhibo Fan, Yuchen Ma, Zeming Li, and Jian Sun. Generalized few-shot object detection without forgetting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4527–4536, 2021. 4.1

[10] 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. *International Journal of Computer Vision*, 132(2):581–595, 2024. 2.3

[11] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*, 2020. 2.3

[12] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2021. 2.2

[13] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2021. 2.2

[14] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019. 3.2, A.4, A.8

[15] Adi Haviv, Jonathan Berant, and Amir Globerson. Bertese: Learning to speak to bert. *arXiv preprint arXiv:2103.05327*, 2021. 2.3

[16] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 2.3

[17] 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*, pages 16000–16009, 2022. 2.3

[18] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR, 2019. 2.3

[19] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 2.3

[20] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European*

Conference on Computer Vision, pages 709–727. Springer, 2022. 2.3

[21] Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438, 2020. 2.3

[22] 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*, pages 8420–8429, 2019. 2.1

[23] Gaoussou Youssouf Kebe, Padraig Higgins, Patrick Jenkins, Kasra Darvish, Rishabh Sachdeva, Ryan Barron, John Winder, Donald Engel, Edward Raff, Francis Ferraro, and Cynthia Matuszek. A spoken language dataset of descriptions for speech-based grounded language learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021. 4.2

[24] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. 2.3

[25] Mona Köhler, Markus Eisenbach, and Horst-Michael Gross. Few-shot object detection: A comprehensive survey. *arXiv preprint arXiv:2112.11699*, 2021. 2.1

[26] Weicheng Kuo, Yin Cui, Xiuye Gu, AJ Piergiovanni, and Anelia Angelova. F-vlm: Open-vocabulary object detection upon frozen vision and language models. *arXiv preprint arXiv:2209.15639*, 2022. 2.2

[27] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021. 3.2

[28] Bohao Li, Boyu Yang, Chang Liu, Feng Liu, Rongrong Ji, and Qixiang Ye. Beyond max-margin: Class margin equilibrium for few-shot object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7363–7372, 2021. 2.1

[29] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975, 2022. 2.2, 3.2, 4.2, 4.3, 4.6, A.6, A.7

[30] 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 *European Conference on Computer Vision*, 2014. 1, 3.1

[31] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 3.1, 4.1, 4.2, 4.3, 4.6, A.5, A.6, A.7

[32] Yuning Lu, Jianzhuang Liu, Yonggang Zhang, Yajing Liu, and Xinmei Tian. Prompt distribution learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5206–5215, 2022. 2.3

[33] Jiawei Ma, Yulei Niu, Jincheng Xu, Shiyuan Huang, Guangxing Han, and Shih-Fu Chang. Digeo: Discriminative geometry-aware learning for generalized few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3208–3218, 2023. 4.1, 4.2, A.8

[34] Yechi Ma, Neehar Peri, Shuoquan Wei, Wei Hua, Deva Ramanan, Yanan Li, and Shu Kong. Long-tailed 3d detection via 2d late fusion, 2023. (document), 4.1, 4.2

[35] 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. In *European Conference on Computer Vision*, pages 728–755. Springer, 2022. 2.2

[36] Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. *arXiv preprint arXiv:2306.09683*, 2023. 2.2

[37] Neehar Peri, Achal Dave, Deva Ramanan, and Shu Kong. Towards long-tailed 3d detection. In *Conference on Robot Learning*, pages 1904–1915. PMLR, 2023. (document), 4.1, 4.2

[38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 3.1

[39] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439, 2019. 4.2

[40] 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)*, pages 2556–2565, Melbourne, Australia, 2018. Association for Computational Linguistics. A.5

[41] Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer

Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*, 2020. 3.2

[42] Bo Sun, Banghuai Li, Shengcai Cai, Ye Yuan, and Chi Zhang. Fsce: Few-shot object detection via contrastive proposal encoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7352–7362, 2021. 2.1

[43] Xin Wang, Thomas E. Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu. Frustratingly simple few-shot object detection. In *International Conference on Machine Learning (ICML)*, 2020. (document), 1, 2.1, 3.1, 4.1, 4.1, 4.3, 4.2, 4.5, A.1, A.4, A.8

[44] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Growing a brain: Fine-tuning by increasing model capacity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2471–2480, 2017. 2.3

[45] Zifeng Wang, Zhenbang Wu, Dinesh Agarwal, and Jimeng Sun. Medclip: Contrastive learning from unpaired medical images and text. In *Conference on Empirical Methods in Natural Language Processing*, 2022. 4.6

[46] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021)*, 2021. 4.6

[47] Aming Wu, Yahong Han, Linchao Zhu, and Yi Yang. Universal-prototype enhancing for few-shot object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9567–9576, 2021. 2.1

[48] Jiaxi Wu, Songtao Liu, Di Huang, and Yunhong Wang. Multi-scale positive sample refinement for few-shot object detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pages 456–472. Springer, 2020. 4.1

[49] Wenhao Wu, Zhun Sun, and Wanli Ouyang. Revisiting classifier: Transferring vision-language models for video recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 2847–2855, 2023. 2.3

[50] Yang Xiao, Vincent Lepetit, and Renaud Marlet. Few-shot object detection and viewpoint estimation for objects in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3090–3106, 2022. 2.1, 4.1

[51] Yinghui Xing, Qirui Wu, De Cheng, Shizhou Zhang, Guoqiang Liang, Peng Wang, and Yanning Zhang. Dual modality prompt tuning for vision-language pre-trained model. *IEEE Transactions on Multimedia*, 2023. 2.3

[52] Jingyi Xu, Hieu Le, and Dimitris Samaras. Generating features with increased crop-related diversity for few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19713–19722, 2023. 2.1

[53] Yifan Xu, Mengdan Zhang, Chaoyou Fu, Peixian Chen, Xiaoshan Yang, Ke Li, and Changsheng Xu. Multi-modal queried object detection in the wild. *Advances in Neural Information Processing Systems*, 36, 2024. 3.2, 4.3, A.6, A.7

[54] 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*, pages 9577–9586, 2019. 4.1

[55] Jeffrey O Zhang, Alexander Sax, Amir Zamir, Leonidas Guibas, and Jitendra Malik. Side-tuning: a baseline for network adaptation via additive side networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 698–714. Springer, 2020. 2.3

[56] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 2.3

[57] Renrui Zhang, Rongyao Fang, Wei Zhang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv preprint arXiv:2111.03930*, 2021. 2.3

[58] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16793–16803, 2022. 2.2, 4.3, A.5, A.4, A.5, A.6, A.7

[59] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16816–16825, 2022. 2.3

[60] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. 2.3

[61] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Probabilistic two-stage detection. In *arXiv preprint arXiv:2103.07461*, 2021. A.4

[62] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *European Conference on Computer Vision*, pages 350–368. Springer, 2022. 2.2,

4.3, 4.4, 4.6, A.3, A.4, A.5, A.6, A.7, A.8

[63] Beier Zhu, Yulei Niu, Yucheng Han, Yue Wu, and Hanwang Zhang. Prompt-aligned gradient for prompt tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15659–15669, 2023. 2.3