
Developing Object Perception in the Low Data Regime



PRANNAY KAUL

Visual Geometry Group · Wadham College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Supervised by PROFESSORS ANDREW ZISSERMAN
AND WEIDI XIE

Michaelmas 2024

To Mum and Dad, my launchpad and my rocket.

Never get discouraged if you fail.

Learn from it. Keep trying.

Thomas Edison

Acknowledgements

First and foremost, I offer my sincere gratitude to my supervisor Andrew Zisserman for his advice, patience and support. His vast ocean of knowledge and enthusiasm for developing new ideas made this thesis possible. Similarly, I would like to thank my co-supervisor Weidi Xie for his frequent guidance and detailed discussions. I have been extremely fortunate to learn from both of them.

To all the members of VGG: I thank them for making the lab an interesting place to work with even more interesting people. The collegiate atmosphere and many discussions made the long hours in the IEB more enjoyable. In particular to some members who started with me: Rhydian, Sagar, Liliane and Max; sharing a few beers and rambling discussions with them helped me immensely within the lab and outside. And to newer members of VGG: Tim, Luke, Guanqi and Suny; working amongst them rejuvenated the lab for me after the pandemic. I offer special thanks to Ashish Thandavan for answering queries about the cluster and keeping my experiments running.

I gratefully thank the EPSRC AIMS CDT for funding my DPhil and particularly, Wendy Poole for her endless help and prompting with the administrative side of the DPhil.

To Zhizhong Li, Hao Yang, Yonatan Dukler and the rest of my colleagues at Amazon for their generosity during my internship in Seattle.

To Emilia and Berklee: who brought a lot of joy to my life in Oxford, gave me refuge from science and changed my streaming suggestions forever. I hope we are a bit more than just “housemates” now. To my friends from my 1st innings in Oxford, who gave me regular breaks from the lab in London: Matthew and David, for bringing the good times and endless comedy; Packer, for always giving me a place to stay; and to the rest of the gang for the many fun evenings. To my oldest and dearest friend: Fadi, for the many years of friendship and encouragement.

To Lauren: meeting and being with her is without a doubt the best result of my 2nd innings in Oxford.

Finally, to my family: my late Nana, who showed me that passion and ambition has no sunset; my Nani, who keeps me grounded and reminds me who to thank for any success; my Dada and Dadi, whose blessings have never let me down; my brother, who has grown to inspire me in ways I would not have bet on when we were younger; my dad, who sacrificed much to give me the opportunities I have had, picks me up when I fall or have doubts, and sets the example for effort in any endeavour; and my mum, the foundation which never lets me down and the light which keeps me on the right path.

Abstract

Objects are central to human perception and understanding of the world. There is an abundance of images available on the internet covering the vast number of objects in the world, however, labelling these images exhaustively to cover all objects is infeasible—limiting the utility of systems requiring strong supervision through large labelled datasets. To address this issue, this thesis develops methods to enable novel objects to be learnt with limited use of manually labelled data.

First, we consider the problem of few-shot object detection, which is the problem of learning to expand the set of objects which can be detected with only a few manually labelled examples. We show that the few examples available for novel categories can be used to accurately pseudo-label existing data to yield a large number of novel pseudo-annotations for further detector training.

Second, we address the more challenging problem of open-vocabulary object detection, which requires learning to detect novel object categories with no annotated data. We demonstrate the utility of detailed natural language descriptions to provide additional visual information for novel object detection. Moreover, we show that visual exemplars can be aggregated and combined with object descriptions to yield multi-modal classifiers for superior novel object detection.

Finally, we consider the problem of object hallucinations in large vision-language models. We propose an automatic method to evaluate the presence of object hallucinations in detailed natural language descriptions of images generated by large vision-language models. We make use of language models and labelled detection data to automatically and robustly analyse the presence of object hallucinations in generated descriptions.

Keywords – object detection, deep learning, open-vocabulary learning, multi-modal

This thesis is submitted to the Department of Engineering Science, University of Oxford, in fulfilment of the requirements for the degree of Doctor of Philosophy. This thesis is entirely my own work, and except where otherwise stated, describes my own research.

Prannay Kaul, January 2024.

Contents

1	Introduction and Motivation	9
1.1	Motivation	10
1.1.1	Learning New Categories from Limited Examples	10
1.1.2	Learning New Categories from Natural Language	11
1.1.3	Beyond Object Detection and Towards Scene Understanding	12
1.2	Literature Review	13
1.2.1	Sliding Window and Region-based Methods	13
1.2.2	Two-(Multi-)Stage Object Detectors	15
1.2.3	One-Stage Object Detectors	16
1.2.4	Object Detection Datasets	19
1.2.5	Extensions beyond Object Detection	20
1.3	Thesis Outline and Publications List	21
2	Label, Verify, Correct: A Simple Few Shot Object Detection Method	23
2.1	Introduction	25
2.2	Related Work	27
2.3	Background and Supervision Collapse	29
2.3.1	Problem Definition	29
2.3.2	Training Strategy	29

2.4	Method	33
2.4.1	Candidate Sourcing	33
2.4.2	Label Verification	34
2.4.3	Box Correction	34
2.5	Experiments	35
2.5.1	Few-Shot Object Detection Benchmarks	35
2.5.2	Implementation Details	36
2.5.3	Ablation Studies	37
2.5.4	Comparison to SotA	39
2.5.5	Qualitative Results	41
2.6	Conclusion	42
2.7	Acknowledgements	43
2.A	Supervision Collapse	44
2.A.1	On Generalisability of RPNs	44
2.A.2	On Transferability of Base Features	44
2.B	Ignore Regions	47
2.C	More Qualitative Results	49
2.C.1	Label Verification	49
2.C.2	Box Correction	49
2.D	Quantitative Results	52
2.E	Limitations	56
2.F	Ethical Concerns	56
3	Multi-Modal Classifiers for Open-Vocabulary Object Detection	58
3.1	Introduction	60
3.2	Related Work	63
3.3	Method	65
3.3.1	Preliminaries	66

3.3.2	Text-based Classifiers from Language Descriptions	67
3.3.3	Vision-based Classifiers from Image Exemplars	69
3.3.4	Constructing Classifiers via Multi-Modal Fusion	71
3.4	Experiments	71
3.4.1	Datasets and Evaluation Protocol	72
3.4.2	Implementation Details	73
3.4.3	Open-Vocabulary Detection Results	75
3.4.4	Ablation Study	78
3.5	Conclusion	80
3.A	Ablation Studies	81
3.B	A Closer Look at “rare” Class Performance	84
3.C	Vision-based Classifier Pipeline Implementation Details	85
3.D	Sourcing Image Exemplars	86
3.E	Varying Number of Image Exemplars for Vision-based Classifiers . .	88
3.F	Additional Example Class Descriptions	89
3.F.1	Generated descriptions for “bagpipe” (rare)	89
3.F.2	Generated descriptions for “trench coat” (rare)	90
3.F.3	Generated descriptions for “walrus” (rare)	90
3.F.4	Generated descriptions for “briefcase” (common)	92
3.F.5	Generated descriptions for “coin” (common)	92
3.F.6	Generated descriptions for “waffle” (common)	92
3.F.7	Generated descriptions for “avocado” (frequent)	94
3.F.8	Generated descriptions for “sausage” (frequent)	94
3.F.9	Generated descriptions for “spectacles” (frequent)	94
3.G	Example Image Exemplars	96
3.H	More Qualitative Results	98
3.I	Dugong	99

4	THRONE: A Hallucination Benchmark for the Free-form Generations of Large Vision-Language Models	101
4.1	Introduction	103
4.2	Related Work	106
4.3	THRONE	108
4.3.1	Evaluating Hallucinations with THRONE	110
4.3.2	Ensuring Robustness via Ensembling	111
4.4	Implementation	113
4.4.1	Dataset	113
4.4.2	Language Models	113
4.4.3	Prompt Ensembling	114
4.5	Evaluation Results	115
4.5.1	Models	115
4.5.2	THRONE Results on COCO	115
4.5.3	THRONE Results on Objects365	116
4.5.4	Completing POPE for Type II Hallucinations	118
4.5.5	Ablations	119
4.6	Improved Baselines	120
4.6.1	Visual Instruction Tuning Data Augmentation	120
4.6.2	Improved Baseline Results	120
4.7	Conclusion	121
4.A	Evaluated LVLM Details	123
4.B	Voting Mechanism Ablation	128
4.C	THRONE vs. POPE Sampling	129
4.D	CHAIR Overview	131
4.E	Qualitative Evaluation of THRONE	133
4.E.1	Evaluation Method	133
4.E.2	Discussion	133

4.F	Improved Baseline Implementation Details	135
4.F.1	Object Enumeration Implementation Details	135
4.F.2	Negative Sampling Implementation Details	136
4.F.3	Object Enumeration Data Details	137
4.F.4	Inference Details	137
4.F.5	Ablation Results	137
4.G	Limitations	139
4.H	Ethical Considerations	139
5	Conclusion	142
5.1	Achievements and Impact	142
5.2	Future Work	144
	References	147

Chapter 1

Introduction and Motivation

Objects are the fundamental building blocks of our visual world. This basic unit of perception is the first step in the process of understanding the visual world—scene understanding. Important information about a scene flows from objects such as object parts, object attributes and relationships between objects. Just as humans recognise objects to navigate surroundings, make sense of the environment, and interact with the world, object perception allows computers to do the same. Therefore, injecting object perception into computer vision systems is an essential part of creating useful artificial agents. Object perception is already widely used to avoid hazards in autonomous driving [Janai et al. 2020] and tracking wildlife migrations in satellite imagery [Duporge et al. 2021].

The deep-learning revolution ignited [Krizhevsky et al. 2012] due to the availability of large datasets containing hundreds of object categories each with thousands of manually labelled images [J. Deng et al. 2009]. The vast quantity of images uploaded and shared on the internet are an expansive source of data to train computer vision systems for object perception, likely containing examples of all object categories on Earth. Despite this apparent abundance, there is no possibility of collecting and manually labelling all of these images with useful information. Furthermore, the frequency of images on the internet is highly skewed towards a relatively small number popular categories e.g. *person*, *car*, *dog*, *cat* etc. This makes the *collection* of a useful dataset for all object categories in the visual world impossible. Finally, the labelling process itself grows more expensive when the desired annotations pass beyond simple image-level labels (e.g. a single word) to

structured data (e.g. drawing bounding boxes and categorising multiple objects in an image). These challenges motivate the need to develop methods for object perception under various conditions of data availability.

1.1 Motivation

1.1.1 Learning New Categories from Limited Examples

The revolutionary advances of computer vision in the last decade have been built upon deep learning [Simonyan and Zisserman 2015; Ren et al. 2016; Kaiming He et al. 2016; Ronneberger et al. 2015], however the strongest models require datasets with large numbers of labelled examples [J. Deng et al. 2009; T.-Y. Lin et al. 2014]. Additionally, these models are limited to perform on a closed, small set of categories i.e. the categories in the training dataset. Obtaining these datasets is costly and time consuming, requiring human workers to manually provide the annotations of interest. The cost of data collection is compounded by the fact that the annotation process grows more expensive when structured annotations for each image are needed rather than simple image-level annotations e.g. object detection data collection [Everingham et al. 2010] is more expensive than image classification [J. Deng et al. 2009]. Moreover, in tasks which require specialist knowledge, an expert is required to provide the annotations, greatly increasing the cost of data collection. Finally, some specific tasks require annotations for which vast quantities of data do not exist e.g. medical imaging [Litjens et al. 2017].

In contrast, the human visual system can learn to recognise new objects from only a few examples [Smith and Slone 2017]—a toddler can learn to recognise a new animal from a single cartoon image in a book.

This distinction has motivated the development of *few-shot learning* which aims to solve a task with only a few labelled examples. In computer vision, few-shot learning has mainly been applied to image classification [Koch et al. 2015; Vinyals et al. 2016; Snell et al. 2017; Sung et al. 2018], with fewer works on other tasks like few-shot action recognition [Ju et al. 2022; Bishay et al. 2019; Hongguang Zhang et al. 2020]. These image-level tasks utilise the N -way K -shot protocol for evaluation—the model is required to predict a category from N *novel* cate-

gories given only K labelled examples per category. This protocol is in opposition to how humans learn to recognise objects—humans can use a single example to *incrementally add new categories* to their object vocabulary.

Few-shot object detection (FSOD) [H. Chen et al. 2018; Kang et al. 2019] is more reflective of how humans learn to recognise objects. Object detection is learnt on an initial set of object categories. This initial set of categories is subsequently expanded to include a new set of categories using only a few examples.

In this thesis, we investigate learning new categories from limited examples through few-shot object detection.

1.1.2 Learning New Categories from Natural Language

Natural language is the fundamental medium in which modern humans communicate their perceptions of the world; it serves as a common knowledge base for humans to share their experiences. Therefore it is natural to consider the use of language as a medium to communicate with computer vision systems. Large language models (LLMs) [Touvron et al. 2023a; OpenAI 2023a; Brown et al. 2020; Radford et al. 2019] trained for next word prediction on large corpora of text from the internet have been shown to be capable of generating coherent text relevant to a wide range of tasks such as natural language translation [Vaswani et al. 2017], text summarisation [Raffel et al. 2020] and coding [L. Ouyang et al. 2022].

The training of LLMs instils them with a huge amount of knowledge about the visual world, which can be leveraged to link the visual contents of images with specific and relevant knowledge learnt from the large text corpora. This desire has led to the development of vision-language models (VLMs) which jointly learn common visual and language representations [Radford et al. 2021; Jia et al. 2021] using image-text pairs data from the internet. These models have been shown to be capable of performing a wide range of tasks in a zero-shot manner, such as image classification, action recognition in videos and image-text retrieval.

The combination of LLMs and VLMs provides the opportunity for vision systems to learn new categories (and other visual concepts) without the need for manually collected annotations for the task of interest. As outlined previously, the traditional approach of collecting a large dataset of manual annotations has many

drawbacks and leveraging the vast text corpora and image-text pairs available on the internet is a promising and cheaper alternative. The task of zero-shot object detection [Bansal et al. 2018] and its general form open-vocabulary object detection [Zareian et al. 2021; X. Zhou et al. 2022] investigate the injection of natural language into object detection for the purpose of learning new categories.

In this thesis, we explore the use of LLMs and VLMs to learn new categories through open-vocabulary object detection.

1.1.3 Beyond Object Detection and Towards Scene Understanding

As mentioned previously, an overall goal of computer vision is scene understanding and the most basic units of a scene are objects. Therefore object detection, which localises and classifies multiple objects in an image, is a fundamental task in computer vision and has rightly received a lot of attention. However, there are other tasks to scene understanding which have also received attention in the computer vision community. Instance segmentation [Kaiming He et al. 2017] extends objects detection by computing an object mask for each object instance in an image, rather than just bounding boxes as in object detection. Semantic segmentation [Long et al. 2015] requires applying a category label to *every* pixel in an image, not just the pixels belonging to an object—e.g. *person*, *car*, *dog* etc. and *sky*, *road*, *grass* etc. Panoptic segmentation [Kirillov et al. 2019] is a dense computer vision task (each pixel in an image is assigned a class label) which combines semantic segmentation (assigning a label to every pixel in an image) and instance segmentation (segmenting each object instance in an image). Thus giving a complete understanding of *objects* in an image—countable objects (often referred to as “things”) like person, car, dog, cat etc. are each assigned a unique object ID, object mask and category label (instance segmentation) and non-countable objects (often referred to as “stuff”) like sky, road, grass etc. are each assigned a mask and category label (semantic segmentation).

Scene graph generation [Krishna et al. 2017] is another computer vision task which uses a graph to formalise scene understanding. The nodes of the graph represent objects and the edges represent relationships between the objects with a label

assigned to each edge indicating the type of relationship. This basic graph can be extended to include object attributes, bounding boxes, object parts and other information.

Finally, detailed image captioning [Vinyals et al. 2015] is a task which aims to describe the objects in an image with free-form natural language and is more similar to how a human might express scene understanding.

This thesis explores the detailed image captioning task with respect to object perception as another step towards scene understanding.

1.2 Literature Review

The goal of this thesis is to improve the ability of vision systems to learn new categories without the need for large amounts of manually collected object detection annotations for each new category.

This section covers literature relevant to object detection, without covering details of object detection tasks in the low data regime. Literature for specific tasks in the low data regime is covered in the relevant chapters: few-shot object detection is covered in Chapter 2; open-vocabulary object detection and natural language for image classification are covered in Chapter 3; and large vision-language models are covered in Chapter 4.

In this section, we review object detection—the task of localising and classifying objects in an image. We split this review into five parts: (1) Sliding Window and Region-based Methods; (2) Two-stage Object Detection; (3) One-stage Object Detection—further sub-divided into anchor-based and anchor-free methods; and (4) Object Detection Datasets.

Finally, we review some key milestones in tasks related to object detection such as semantic segmentation and instance segmentation.

1.2.1 Sliding Window and Region-based Methods

Prior to the re-emergence of deep learning, the most popular models for object detection were either based on sliding window methods which exhaustively search an image for objects—the foremost example being deformable part models (DPMs) [P.

Felzenszwalb et al. 2008; P. F. Felzenszwalb et al. 2010; Lim et al. 2013], or region-based methods which used (a combination of) image primitives to propose regions likely to contain objects [Alexe et al. 2010; Uijlings et al. 2013; Zitnick and Dollar 2014; Krähenbühl and Koltun 2014].

DPMs subscribe to the notion that objects can be modelled by parts with a deformable configuration. A representative DPM [P. Felzenszwalb et al. 2008] uses a histogram of orientated gradient (HOG) features [Dalal and Triggs 2005] extracted from an image pyramid. They used a “root filter” to exhaustively search a coarse resolution feature map for matching object HOG features and “part filters” operating at finer resolutions. Dynamic programming is used to compute the best part placements given a root placement and scores are computed between the HOG features and the modelled filters. The root and part filters are solved using a latent support vector machine (SVM). A fundamental drawback of this method was that it used a sliding window exhaustive search for object placement, most of which involves computations on background regions which are not useful for object detection.

To improve computational efficiency, methods which proposed regions likely to contain objects were developed. Before objects were learnt automatically from large human labelled datasets, the “objectness” of a region was first computed using handcrafted rules such as uniqueness/saliency, colour contrast, edge density and straddling superpixels [Alexe et al. 2010].

Selective search [Uijlings et al. 2013] uses a hierarchical approach to perform bottom-up grouping of regions. Given a set of initial regions [P. F. Felzenszwalb and Huttenlocher 2004], pairs of neighbouring regions with the highest similarity (defined by image primitives e.g. colour histograms) are sequentially merged until the whole image is one region. After introducing some randomness (to yield diverse proposals) and combining multiple similarity metrics, a set of bounding box proposals based on the segmented regions are computed. Given these proposals, colour-SIFT descriptors [van de Sande et al. 2010] are used to train SVMs for classifying each proposal, enabling object detection.

Other methods for generating object proposals for detection include EdgeBoxes [Zitnick and Dollar 2014], geodesic object proposals [Krähenbühl and Koltun 2014] and

LPO [Krahenbuhl and Koltun 2015]. EdgeBoxes [Zitnick and Dollar 2014] found object proposals by computing the number of edges *wholly* contained in a bounding box and subtracting the number of edge contours *intersecting* the bounding box. This was computed efficiently using integral images [Viola and Jones 2001] and non-maximum suppression removed highly correlated proposals. Geodesic object proposals [Krähenbühl and Koltun 2014] used a graph-based approach to compute object *segmentation* proposals. Foreground and background masks were computed for a set of superpixels placed throughout an image. Computing the level sets of a signed geodesic distance transform over the image yields multiple image regions. Regions corresponding to certain critical level sets were selected as object segmentation proposals. LPO [Krahenbuhl and Koltun 2015] used low-level image features to train a global segmentation model jointly with multiple local ones—joint training enabled individual models to specialise on particular object appearances. Various contemporaneous methods including the ones above are compared and analysed in [Hosang et al. 2015], which also introduces the Average Recall (AR) metric for evaluating a given method at a particular number of object proposals at various intersection-over-union (IoU) thresholds—averaging over IoU thresholds additionally rewards localisation performance. Averaging over IoU thresholds is now standard practice in object detection evaluation—mean average precision (mAP) and average recall (AR) [T.-Y. Lin et al. 2014].

1.2.2 Two-(Multi-)Stage Object Detectors

Canonical works in two-stage object detectors are the R-CNN series, namely, R-CNN [Girshick et al. 2014], Fast R-CNN [Girshick 2015], Faster R-CNN [Ren et al. 2016]. R-CNN used Selective Search [Uijlings et al. 2013] to propose class-agnostic bounding boxes (first stage), these proposals were then warped to a fixed size, related features computed by a CNN and class-specific linear SVMs used to perform classification (second stage). Fast R-CNN contains a more efficient second stage. Deep features are computed for the whole image (first stage). For each Selective Search proposal, the deep features are pooled in a unique way (called ROI Pool) to a fixed size before classification and final bounding box placement (second stage) is performed by convolutional layers. Online hard example mining (OHEM) [Shri-

[vastava et al. 2016](#)] improved the training of Fast R-CNN. All regions-of-interest (RoIs) are ranked by their training loss; a computationally inexpensive step. The highest loss RoIs are selected (removing highly correlated ROIs with non-maximum suppression) for updating the Fast R-CNN model. Faster R-CNN [[Ren et al. 2016](#)] improved the first stage of R-CNN/Fast R-CNN, in which a Region Proposal Network (RPN) yields object bounding boxes using a set of heuristics regarding box size and aspect ratio (called anchors). Furthermore, to improve efficiency a common set of deep features were used for both the first and second stages. The issue of detecting objects at multiple scales was addressed with Feature Pyramid Networks (FPNs) [[T.-Y. Lin et al. 2017a](#)] in which proposals and predictions are made at multiple feature resolutions. The use of a FPN (and variants) is common in many current object detectors. Probabilistic two-stage detection [[X. Zhou et al. 2021](#)] views two-stage detection as inferring a joint distribution over objectness (foreground vs. background) *and* class labels.

Two-stage detectors are extended to multi-stage detectors in which the additional stages are used to refine the predictions of the previous stage. Cascade R-CNN [[Z. Cai and Vasconcelos 2018](#)] introduced this idea by cascading multiple Fast R-CNN stages.

1.2.3 One-Stage Object Detectors

Anchor-based Methods

One-stage detectors have a general advantage in terms of inference time over two-stage detectors as they only require a single forward pass and decoding stage. The initial series of YOLO works [[Redmon et al. 2016](#); [Redmon and Farhadi 2017](#); [Redmon and Farhadi 2018](#)] provide sequential improvements to the detector. In YOLOv1 [[Redmon et al. 2016](#)] an image is divided into a regular grid in which anchor bounding boxes (determined by heuristics) are given scores in parallel to predicting class labels for each grid cell. SSD [[W. Liu et al. 2016](#)] removes the need for a grid and places anchor boxes at every location in a feature map. These anchor boxes are matched to ground-truth bounding boxes using an IoU threshold. This allowed each ground truth bounding box to have more than one positive anchor box, resulting in easier training. Furthermore, SSD combined predictions from multiple feature resolutions, however these feature maps were less rich than the

more popular FPN structure [T.-Y. Lin et al. 2017a]. The limitations of YOLOv1 are partially addressed in YOLOv2/YOLO9000 [Redmon and Farhadi 2017], which makes use of an RPN structure as in Faster R-CNN [Ren et al. 2016] but gives objectness and classification scores for anchor boxes at each location in feature space, rather than selecting a limited number as required in two-stage networks. Furthermore, the anchor box dimensions are determined by clustering on the training data rather than using heuristics. Previous YOLO versions are improved [Redmon and Farhadi 2018] through the addition of a FPN [T.-Y. Lin et al. 2017a] to address object scale issues. YOLO models have now reached an eighth iteration [Ultralytics 2023]. RetinaNet [T.-Y. Lin et al. 2017b] introduces “focal loss”, a novel loss function designed to address the heavy imbalance between foreground and background in object detection. This enabled their model to densely classify all anchor boxes in an image during training, rather than processing a small balanced (between foreground and background) subset of object proposals as required in Faster R-CNN [Ren et al. 2016].

Anchor-free Methods

CornerNet [Law and Jia Deng 2018] gives a rather different formulation of the problem which does not rely on predefined anchor boxes. Instead, bounding box corners for each class (top left and bottom right) are regressed separately and matched together based on the euclidean distance between their features. By treating “objects as points”, CenterNet [X. Zhou et al. 2019] revolutionised object detection by simply predicting object centres for each class and regressing a symmetric bounding box around each proposed centre. FCOS [Tian et al. 2019] generalises CenterNet—treating all foreground locations as positive locations and predicting the ideal bounding box from these locations (potentially at the extremity of an object). This drastically reduced the size of model outputs when compared to RetinaNet [T.-Y. Lin et al. 2017b], which used anchor boxes.

A new class of object detectors begin with DETR [Carion et al. 2020] which uses the Transformer architecture [Vaswani et al. 2017] to perform object detection by casting it as a set prediction problem. This method does not use anchors or predict geometric entities like centre points or corners, instead a fixed number of “object queries” are learnt and used to predict the bounding boxes and class labels of

objects in an image. DETR is trained using a bipartite matching loss between the predicted and ground truth bounding boxes, with each ground truth box matched to exactly one predicted box. This removes the need for handcrafted rules for matching predictions to ground truth boxes during training. Furthermore, this removes the need for non-maximum suppression—a common post-processing step in object detection to remove multiple overlapping predictions for the same object, which arises from the use of many anchors or geometric entities to predict object bounding boxes. The drawbacks of DETR are the extremely long training time required for convergence and poor performance on small objects due to limited feature resolution. This is addressed in Deformable-DETR [X. Zhu et al. 2020] which combines local attention [Parmar et al. 2018] and deformable convolutions [J. Dai et al. 2017] to reduce computation requirements to yield faster training convergence and improved feature resolution. Conditional DETR [Meng et al. 2021] also addresses the convergence issue by replacing the fixed learnt object queries in decoder cross-attention with dynamically generated queries based on learnable 2D coordinates and previous decoder layer outputs, thus guiding decoder attention to more relevant image regions. DAB-DETR [S. Liu et al. 2022] reintroduces the concept of anchors to DETR by using a set of *learnable anchors* to replace the learnt object queries. DN-DETR [F. Li et al. 2022] addresses the slow convergence by introducing an auxiliary loss function throughout the Transformer decoder to reconstruct groundtruth bounding boxes from noised versions. DINO-DETR [Hao Zhang et al. 2022] combines elements of DAB-DETR and DN-DETR to improve the performance and efficiency of DETR by introducing a contrastive denoising method to ignore heavily noised bounding boxes and using mixed dynamic and static object queries. Co-DETR (the current state-of-the-art) [Zong et al. 2023] abandons the strict one-to-one matching between predicted and ground truth bounding boxes in DETR and uses a hybrid loss based on separate one-to-many label assignments (like in Faster R-CNN) and one-to-one matching heads to improve encoder training. Finally, a recent work [Ouyang-Zhang et al. 2022] uses one-to-many label assignments exclusively and reintroduces non-maximum suppression, showing improved performance over a one-to-one baseline.

The trend in object detection is towards models based on DETR derivatives but with the addition of methods (such as anchors and NMS) from earlier anchor-based

methods.

1.2.4 Object Detection Datasets

Early works in object detection focus on the detection of a single object class, therefore the “classification” part of the object detection is a simple foreground/background binary classification. The common single object classes used were pedestrians [Dalal and Triggs 2005; Papageorgiou and Poggio 2000] and faces [Rowley et al. 1998].

The PASCAL Visual Object Classes (VOC) [Everingham et al. 2010] dataset was the first large scale dataset for object detection with multiple object classes. When published, to distinguish it from single object class datasets (e.g. for pedestrians or faces) the task was termed “object category detection”. In current literature (and this thesis) the term “object detection” is used for multiple object class detection as established in the PASCAL VOC dataset. Initially starting with 4 classes and bounding boxes only, the dataset grew to 20 classes in the final version and included instance masks for objects. Moreover, PASCAL VOC set the standard for evaluation metrics in object detection, using intersection-over-union (IoU) thresholds to define positive detections and average precision from retrieval methods to evaluate object detection performance. The Microsoft Common Objects in Context (COCO) [T.-Y. Lin et al. 2014] built on the PASCAL VOC dataset by increasing the number of object classes to 80 and using images with more object instances and more varied object scale. The COCO dataset contained bounding box annotations and instance masks for objects and additional masks for various “stuff” classes (e.g. sky, grass, road) which were simply grouped as background previously. Objects365 [Shao et al. 2019] is a more recent dataset which contains 365 object classes and 2M images.

The three aforementioned datasets are the primary datasets which provide *exhaustive* object detection annotations. In each case, the three datasets were collected in a manner to ensure all object instances belonging to the relevant classes were annotated. Given this methodology, the datasets implicitly defined negative classes as all classes not labelled in an image.

Due to issues of cost and scale, this exhaustive annotation methodology is not

feasible for larger datasets with larger object vocabularies. Federated datasets such as OpenImages [Krasin et al. 2017] and LVIS [A. Gupta et al. 2019] are created by combining smaller exhaustively annotated datasets (e.g. exhaustive for a single category on a subset of images). These datasets guarantee that all instances of a set of classes are annotated, but not all instances of all classes. Moreover, a subset of classes are guaranteed to be negative classes i.e. not present in an image. This methodology sacrifices the exhaustiveness of annotations for a larger object vocabulary and larger dataset scale and require evaluation metrics which account for the guaranteed negative classes when determining false positives.

Finally, annotations which only provide weak supervision in terms of object detection have been used to improve object detection performance [X. Zhou et al. 2022; Liangyu Chen et al. 2021]. For example, OpenImages [Krasin et al. 2017] provides additional point-level annotations for 5.8k+ classes and image-level labels for 20k+ classes.

1.2.5 Extensions beyond Object Detection

Image classification is arguably the simplest computer vision task—given an image, predict a single class label. The other extreme of this is semantic segmentation [Long et al. 2015], in which each pixel in an image is assigned a class label. The task of object detection is in between these two extremes and has a closely related cousin—instance segmentation [Kaiming He et al. 2017], in which an object mask is predicted for each object in an image instead of bounding box coordinates (as in object detection).

Canonical works in semantic segmentation are FCN [Long et al. 2015] and U-Net [Ronneberger et al. 2015]. FCN uses a CNN to compute a downsampled feature map, which is then upsampled to the original image size using transposed convolutions/deconvolutions to give a class label for each pixel. U-Net [Ronneberger et al. 2015] is a fully convolutional network which uses skip connections to combine features from downsampling and upsampling paths to improve the quality of segmentation masks. This balances the need for high resolution features (for accurate segmentation) and low resolution features (for global context and efficiency).

Mask R-CNN [Kaiming He et al. 2017] is an extension of Faster R-CNN [Ren et al.

2016] to instance segmentation in which an additional fully convolutional head is used to predict a class-agnostic binary mask for each RoI, alongside the bounding box and class label. This separates the mask prediction task from the classification task. HTC [Kai Chen et al. 2019] combines Cascade R-CNN and Mask R-CNN by interleaving the mask and box prediction heads in a cascade fashion.

In semantic segmentation, each pixel is assigned a class label, but there is no notion of object instances. Whereas in instance segmentation, only pixels belonging to an object instance are assigned a class label. These two tasks are unified in panoptic segmentation [Kirillov et al. 2019].

The recent work Mask2Former [Cheng et al. 2022] creates a unified architecture based on the Transformer for instance segmentation, semantic segmentation and panoptic segmentation.

1.3 Thesis Outline and Publications List

Chapters 2 to 4 each contain a research paper. Chapters 2 and 3 have been peer reviewed and published at a conference. Chapter 4 is currently under review. Chapter 2 addresses the problem of few-shot object detection (FSOD)—learning to detect novel object categories with only a few annotated examples. Chapter 3 addresses the problem of open-vocabulary object detection (OVOD)—learning to detect novel object categories with no annotated examples. Chapter 4 assesses the prevalence of object hallucinations in text generated by large vision-language models. We make no modifications to the published papers except for formatting changes to make them consistent with the rest of the thesis. For each paper, we provide a statement of authorship at the end of the relevant chapter. The research papers included in this thesis are as follows:

Chapter 2: “Label, Verify, Correct: A Simple Few Shot Object Detection Method” Prannay Kaul, Weidi Xie, Andrew Zisserman. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Chapter 3: “Multi-Modal Classifiers for Open-Vocabulary Object Detection” Prannay Kaul, Weidi Xie, Andrew Zisserman. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Chapter 4: “THRONE: A Hallucination Benchmark for the Free-form Generations of Large Vision-Language Models” Prannay Kaul, Zhizhong Li, Hao Yang, Yonathan Dukler, Ashwin Swaminathan, CJ Taylor, Stefano Soatto. Under submission to *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Publications Not Included in this Thesis

“RSS-Net: Weakly-supervised Multi-Class Semantic Segmentation with FMCW Radar” Prannay Kaul, Daniele Di Martini, Matthew Gadd, Paul Newman. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020.

Chapter 2

Label, Verify, Correct: A Simple Few Shot Object Detection Method

We start the technical contributions of this thesis with work on few-shot object detection. This considers the problem of learning to detect novel object categories with only a few annotated examples. This task is the simplest form of object detection in the low data regime.

The paper has been published at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2022.

Label, Verify, Correct: A Simple Few Shot Object Detection Method

Prannay Kaul¹ Weidi Xie^{1,2} Andrew Zisserman¹

¹Visual Geometry Group, University of Oxford

²CMIC, Shanghai Jiao Tong University

<http://www.robots.ox.ac.uk/~vgg/research/lvc/>

Abstract

The objective of this paper is few-shot object detection (FSOD) – the task of expanding an object detector for a new category given only a few instances for training. We introduce a simple pseudo-labelling method to source high-quality pseudo-annotations from the training set, for each new category, vastly increasing the number of training instances and reducing class imbalance; our method finds previously unlabelled instances.

Naïvely training with model predictions yields sub-optimal performance; we present two novel methods to improve the precision of the pseudo-labelling process: first, we introduce a verification technique to remove candidate detections with incorrect class labels; second, we train a specialised model to correct poor quality bounding boxes. After these two novel steps, we obtain a large set of high-quality pseudo-annotations that allow our final detector to be trained end-to-end. Additionally, we demonstrate our method maintains base class performance, and the utility of simple augmentations in FSOD. While benchmarking on PASCAL VOC and MS-COCO, our method achieves state-of-the-art or second-best performance compared to existing approaches across all number of shots.

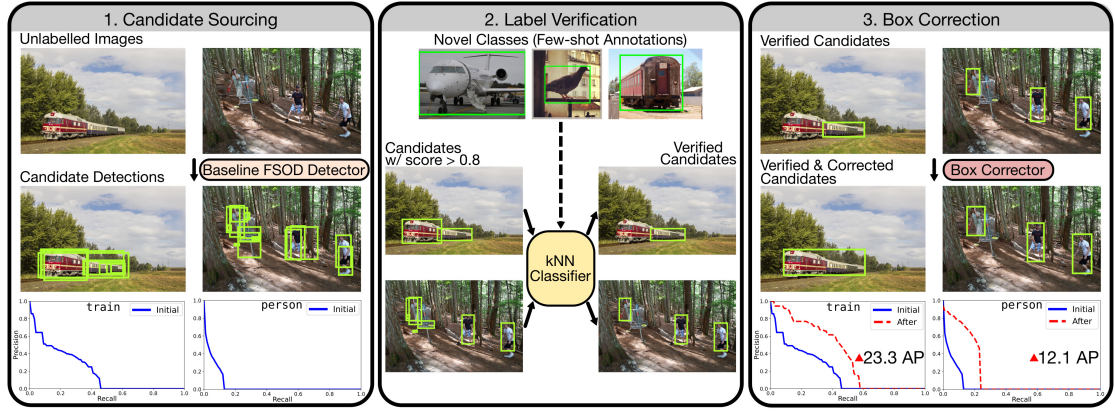


Figure 2.1: **Label, Verify, Correct:** Object detectors naïvely trained with K -shot annotations perform poorly on novel classes (bottom left). We propose to expand the novel class annotation set and re-train *end-to-end*. 1. Given a baseline few-shot object detector, noisy candidate detections are sourced from unlabelled images (left). 2. Labels for each candidate detection are verified by a k NN classifier, constructed from a self-supervised model using the same few-shot annotations, removing large numbers of false positives (centre). 3. A specialised box corrector, drastically improves the remaining bounding boxes, yielding high-quality pseudo-annotations (right). Re-training end-to-end with our pseudo-labelling method yields a large performance boost on novel class detection, improving precision and recall (bottom right).

2.1 Introduction

Object detection refers to the task of determining if an image contains objects of a particular category, and if so, then localising them. In recent years, the community has seen tremendous successes in object detection by training computational models for a set of pre-defined object classes [Ren et al. 2016; T.-Y. Lin et al. 2017b; Tan et al. 2020; M. Xu et al. 2021; X. Dai et al. 2021; Qiao et al. 2021; Han et al. 2021], with large numbers of human annotated labels, e.g. MS-COCO [T.-Y. Lin et al. 2014], and PASCAL VOC [Everingham et al. 2010]. However, such training paradigms have limited the model to only perform well on a closed, small set of categories for which large training data is available.

In contrast, humans can continuously expand their vocabularies, learning to detect a much larger set of categories, even with access to only a few examples [Smith and Slone 2017]. This is also a desirable ability for modern computer vision systems and is studied in the task of few-shot object detection (FSOD) [Kang et al. 2019; X. Yan et al. 2019; Z. Fan et al. 2021; Jiayi Wu et al. 2020; B. Sun et al. 2021]. The goal of our work is FSOD: given an existing object detector that has been trained

on abundant data for some categories, termed *base* categories, we wish to learn to detect *novel* categories using only a few annotations, e.g. 1–30 per category, whilst maintaining performance on the original base categories.

In this paper, we carefully examine the training procedure of the Faster R-CNN two-stage detector, and identify two critical factors that limit its performance on FSOD. *First*, training on base categories leads to a form of “supervision collapse” [Doersch et al. 2020]; this model is naturally trained against detecting instances from novel classes, as they were unlabelled and treated as background; *second*, the FSOD problem involves learning from extremely unbalanced data – only K instances ($K \leq 30$) are available per novel class for training, so the number of training samples for *base* categories is far larger than that for *novel* ones. A model that overfits to a small number (K) of novel instances naturally lacks generalisation ability.

We adopt a simple *pseudo-labelling* technique (see Figure 2.1) and address both the factors that limit performance: we show that the Region Proposal Network (RPN) can be modified to successfully propose regions for the novel categories and use a detector trained with the few-shot *novel* data to label these regions over the images of the large training dataset, producing a set of candidate detections for each novel category. The novelty of our approach is in the two steps used to improve the precision of this candidate set: *first*, we build a classifier for novel categories to verify the candidate detections using features from a network trained with self-supervision (see Figure 2.1, centre); *second*, we train a specialised box regressor that improves the quality of the bounding box of the verified candidates (see Figure 2.1, right). The two steps together yield a large set of high precision pseudo-annotations for the novel categories, removing the class imbalance in the training data. This enables the detector to be trained end-to-end using the pseudo-annotations for the novel categories together with the original groundtruth annotations for base categories, and simultaneously avoids the impact of “supervision collapse” for *all* detector features.

To summarise, our contributions are as follows: (i) we carefully examine the problem of few-shot object detection with the modern two-stage object detector, *e.g.* Faster R-CNN, and identify the issue of “supervision collapse”; (ii) we introduce a novel verification and correction procedure to pseudo-labelling, which

significantly improves the precision of pseudo-annotations, both class labels and bounding box coordinates; (iii) we analyse several critical components of data augmentation and conduct thorough ablation studies to validate their necessity; (iv) with the combination of pseudo-labelling and aggressive data augmentations, we set state-of-the-art (SotA) or comparable performance using a standard Faster R-CNN, for both the challenging MS-COCO benchmark and the PASCAL VOC benchmark.

We discuss limitations of our work and potential ethical concerns in Appendix 2.E and 2.F, respectively. Code and pre-trained models are available from the project webpage.

2.2 Related Work

Object Detection is one of the classical problems in computer vision, which makes it impossible to present a full overview here. We therefore only outline some key milestones here. In general, recent object detection methods can be cast into two sets: one-stage and two-stage detectors. *One-stage* detectors attempt to directly classify *and* regress bounding boxes by, either densely classifying a set of predefined anchor boxes [Redmon et al. 2016; Redmon and Farhadi 2017; Redmon and Farhadi 2018; W. Liu et al. 2016; T.-Y. Lin et al. 2017b; Tan et al. 2020] or densely searching for geometric entities of objects e.g. corners, centres or regions [Law and Jia Deng 2018; X. Zhou et al. 2019; Tian et al. 2019]. Conversely, most *two-stage* detectors propose class-agnostic bounding boxes using a Region Proposal Network (RPN), with predefined sizes and aspect ratios, filtering out many negative (background) locations. These bounding box proposals are pooled to region-of-interest (RoI) features and are classified by a multilayer perceptron (MLP) in the second stage of the detector [Girshick et al. 2014; Girshick 2015; Ren et al. 2016; Yanghao Li et al. 2019].

Few-Shot Object Detection aims to expand the vocabulary of an object detector by only annotating a handful of samples. Several works [H. Chen et al. 2018; Kang et al. 2019; Y.-X. Wang et al. 2019; X. Yan et al. 2019; Y. Xiao and Marlet 2020; Xin Wang et al. 2020; Hanzhe Hu et al. 2021; Bohao Li et al. 2021;

Yiting Li et al. 2021; A. Li and Zhenguo Li 2021; B. Sun et al. 2021; Z. Fan et al. 2021; Chenchen Zhu et al. 2021; Han et al. 2021; Qiao et al. 2021; A. Wu et al. 2021; Z. Yang et al. 2020; Karlinsky et al. 2019; Q. Fan et al. 2020] have been proposed in the recent literature. The meta-learning method, FSRW [Kang et al. 2019] conditions dense query image features from a YOLOv2 network [Redmon and Farhadi 2017] with a separate network operating on a support set. Recently, a simple two-phase fine-tuning approach (TFA) is proposed [Xin Wang et al. 2020], in which a Faster R-CNN model is initially trained on the base data. In the second training phase, only the final classification layer is finetuned on a few samples of novel classes, with the rest of the model fixed. This work initiates a shift away from meta-learning based methods for few-shot object detection. FSCE [B. Sun et al. 2021], alleviates class confusion between novel classes by training a separate supervised contrastive learning [Khosla et al. 2020] head on RoI features. A recent work, DeFRCN [Qiao et al. 2021], decouples the training of RPN features and RoI classification. Another recent work, SRR-FSD [Chenchen Zhu et al. 2021] combines vision and natural language, projecting image features into semantic class embeddings learnt from a large text corpus.

Semi-supervised Object Detection belongs to another related research area. Such a problem setup can be traced back to the pre-deep learning era [Rosenberg et al. 2005], where the goal is to train detectors with a combination of labelled, weakly-labelled and unlabelled data. In the recent literature, the idea of exploiting consistency and self-training has been widely adopted, for example, [Jeong et al. 2019] proposed to enforce the predictions of an input image and its flipped version to be consistent; [Sohn et al. 2020; Y.-C. Liu et al. 2021] pre-trained a detector using a small amount of labelled data and generated pseudo-labels on unlabelled data for further fine-tuning. Generally speaking, these methods aim to train a detector on two separate subsets; one contains images with all objects of all classes being *exhaustively* annotated, and the other subset is fully unlabelled. Therefore, the model trained on the labelled set does not suffer the same issue as in FSOD, where a large number of *novel* object instances are wrongly treated as background during base training. Hence the scarce data issue of FSOD *does* exist in semi-supervised object detection, but “supervision collapse” *does not*.

Self-Training is a method for gaining noisy pseudo-labels which has gained renewed interest since it was initially proposed [Scudder 1965]. In recent years, self-training has been used to improve image classification by using a teacher-student training regime [Hinton et al. 2015; Chollet 2017; Xie et al. 2020]. This idea is extended to general object detection in [Zoph et al. 2020], however, their considered scenario is semi-supervised object detection.

2.3 Background and Supervision Collapse

In this section, we first outline the few-shot object detection task in Section 2.3.1. Next, in Section 2.3.2, we carefully examine the various components of the popular two-stage detector (Faster R-CNN), identify the critical issues that limit its use in the few-shot scenario and draw conclusions to ameliorate such critical issues.

2.3.1 Problem Definition

In this paper, we consider the same problem setup as in TFA [Kang et al. 2019]. Specifically, assuming we are given an image dataset, $\mathcal{D}$, and two annotation sets. First, $\mathcal{Y}_{\text{BASE}}$, with exhaustive annotations on a set of base categories, $\mathcal{C}_{\text{BASE}}$. Second, $\mathcal{Y}_{\text{NOVEL}}^K$, with only K annotations on a set of novel categories, $\mathcal{C}_{\text{NOVEL}}$. Note, the annotations on base categories are exhaustive, but for novel categories most instances are *unlabelled* as only K annotations are provided for the image dataset, $\mathcal{D}$, under the few-shot setting.

2.3.2 Training Strategy

In this section, we start by describing a baseline two-stage detector for the problem of few-shot object detection, following that of TFA [Xin Wang et al. 2020]. In general, a Faster R-CNN detector, $\Phi_{\text{DET}}(\cdot)$, can be formulated as:

$$\Phi_{\text{DET}}(\cdot) = \Phi_{\text{CLS}} \circ \Phi_{\text{ROI}} \circ \Phi_{\text{RPN}} \circ \Phi_{\text{ENC}}(\cdot)$$

where, each input image is sequentially processed by a set of operations: an image encoder, (Φ_{ENC}); a Region Proposal Network, (Φ_{RPN}); a region of interest feature module, (Φ_{ROI}); and a classification layer on the RoI features, (Φ_{CLS}), mapping to a

set of bounding boxes and classes. Note that, each module here contains the same number of convolutional or MLP layers as the standard Faster R-CNN [Ren et al. 2016].

Training few-shot object detectors involves a two-phase training procedure, as detailed below:

Base Training: refers to the standard training for a Faster R-CNN model [Ren et al. 2016; Y. Wu et al. 2019] using *base class annotations* only, $\mathcal{Y}_{\text{BASE}}$. In this work, we do not modify this training regime.

Novel Training: requires extending the base detector, such that it can additionally detect instances from novel categories, i.e. $\mathcal{C}_{\text{BASE}} \cup \mathcal{C}_{\text{NOVEL}}$. In recent works, this is usually done by only training (relatively few) layers on novel and base class data, $\mathcal{Y}_{\text{BASE}}^K \cup \mathcal{Y}_{\text{NOVEL}}$; the detector is not trained end-to-end on novel class data. For example, in TFA [Xin Wang et al. 2020], the fewest possible number of parameters are trained on novel class data, namely, Φ_{CLS} only.

Such a two-phase training strategy naturally leads to two questions: (i) Does an RPN trained on base categories actually generalise, i.e. are regions proposed for instances of the novel categories? (ii) How well do features trained only on base categories actually generalise, in other words, will the RoI features be discriminative for classifying *novel* categories? We aim to answer these two questions on the MS-COCO 30-shot object detection benchmark (these benchmarks are detailed in Section 2.5.1). Specifically, we follow the same data split as in TFA [Xin Wang et al. 2020], with 60 categories being treated as base categories, and 20 as novel categories.

On Generalisability of RPNs

In standard two-stage object detectors, an RPN is considered as a necessary condition for high-performance detections, as classification and box coordinate regression will only act on the proposed regions. Here, we aim to evaluate the quality of an RPN for FSOD, based on recall with respect to the novel categories.

Specifically, we consider the following three settings: *first*, to understand whether

Novel Training?	# Proposals	nAR _{IoU=0.5}	min(R50 _{IoU=0.5})
$\times$	100	49.7	16.2
$\checkmark$	100	71.0	40.0
Ideal RPN [Ren et al. 2016]	100	84.0	64.5
$\times$	1000	82.1	55.9
$\checkmark$	1000	88.5	77.0
Ideal RPN [Ren et al. 2016]	1000	95.0	79.2

Table 2.1: RPN recall evaluation on 20 novel categories from the MS-COCO few-shot object detection benchmark, using an IoU=0.5 criterion. nAR50 – novel class average recall, min(nR50) – minimum novel class recall. Please refer to the text for detailed discussion.

an RPN trained on 60 base categories can directly propose novel object instances, we evaluate the recall of the RPN from the base detector; *second*, we finetune the RPN (composed of 2 convolutional layers) on both base and given novel categories, i.e. $\mathcal{Y}_{\text{BASE}} \cup \mathcal{Y}_{\text{NOVEL}}^{30}$; *third*, the Ideal RPN which is inherited from an off-the-shelf Faster R-CNN trained on exhaustive data for all categories.

Discussion: Table 2.1 presents the large performance gap between an RPN from the base detector, and the Ideal RPN with respect to recalling instances from novel categories. However, finetuning the RPN specific parameters, Φ_{RPN} , on a handful of instances ($K = 30$), yields a substantial increase in average recall (49.7 vs. 71.0, 82.1 vs. 88.5, for 100 and 1000 proposals respectively), largely bridging the performance gap to the Ideal RPN. The minimum class recall also increases substantially (16.2 vs. 40.0, 55.9 vs. 77.0, for 100 and 1000 proposals respectively).

On Transferability of Base Features

In this section, we aim to measure the transferability of the visual features trained on base categories. Specifically, we keep the encoder (Φ_{ENC}) fixed, and finetune, individual or combinations of, subsequent modules ($\Phi_{\text{CLS}}, \Phi_{\text{ROI}}, \Phi_{\text{RPN}}$) during **Novel Training**.

Discussion: As shown in Table 2.2, we compare TFA [Xin Wang et al. 2020] (Setting A1) with finetuning more layers (A3), which tends to be beneficial (13.0 vs. 14.3 nAP), however, it remains substantially lower than the Ideal Faster R-

	Novel Training			Metrics		
Setting	Φ_{CLS}	Φ_{ROI}	Φ_{RPN}	nAP	nAP50	nAP75
A1	✓	✗	✗	13.0	24.7	12.3
A2	✓	✓	✗	13.3	25.6	12.6
A3	✓	✓	✓	14.3	27.5	12.9
A4 (ALL)	✓	✗	✗	18.3	33.3	17.7
Ideal Faster R-CNN [Ren et al. 2016]				43.5	67.4	46.6

Table 2.2: Evaluation on the transferability of base features to 20 novel categories from the MS-COCO few-shot object detection benchmark. During **Novel Training**, we jointly finetune the different modules with the base and novel category data (only the given few-shot annotations, $\mathcal{Y}_{\text{NOVEL}}^K$). As an Oracle test, we also consider to finetune the classifier with *all* MS-COCO annotations, e.g. A4 (ALL). nAP – novel class average precision. Please refer to the text for discussion.

CNN (14.3 vs. 43.5 nAP). To remove the factor caused by insufficient data annotation, we compare TFA (A1) with an Oracle test (A4-ALL) that finetunes the final classifier with exhaustive data on all categories. However, the result still largely underperforms the Ideal Faster R-CNN reference (18.3 vs. 43.5nAP), indicating that the feature encoder is heavily biased towards base classes, and hardly contains discriminative information for classifying instances from novel classes. This may be expected, as these categories were treated as background during **Base Training**.

These experiments demonstrate the “supervision collapse” issue present in FSOD detectors. We note that “supervision collapse” manifests in two ways: *first*, many false positives occur due to class confusion and poor bounding box regression (poor detection precision), *second*, there are many false negatives or missing detections, despite an improved RPN (poor detection recall). We further analyse these manifestations in Appendix 2.A.

Summary

After careful evaluations, we draw the following two conclusions: *first*, updating all parameters specific to the RPN (Φ_{RPN}) is essential to improve recall on novel categories; *second*, features trained on base category data are **not** discriminative enough to classify novel instances, leading to severe performance degradation, and so we update all parameters in the RoI feature module (Φ_{ROI}), in addition to

the classification layer (Φ_{CLS}). These two choices constitute our **Novel Training** process, yielding a stronger baseline detector. This allows enough alleviation of “supervision collapse”, such that our baseline detector can be used as a starting point in our pseudo-labelling method, as will be detailed in the next section.

2.4 Method

To address the “supervision collapse” issue, we adopt a simple *pseudo-labelling* method for mining instances of novel categories, effectively expanding their annotation set. However, pseudo-annotations that are naïvely sourced from the detector (after **Novel Training**), are unreliable, containing a large number of false positives. Here, we establish a method for improving the precision of these candidate pseudo-annotations by automatically filtering out candidates with incorrect class labels, and refining the bounding box coordinates for those remaining. Our method yields a large set of high precision pseudo-annotations for novel categories, allowing the final detector to be trained end-to-end on both base and novel category data. We detail the proposed method in the following sections.

2.4.1 Candidate Sourcing

The goal here is to generate a set of candidate detections that are *potentially* valid pseudo-annotations for novel categories. Specifically, the detector from **Novel Training** (as described in Section 2.3.2), is used to perform inference on the training images ($\mathcal{D}$) to generate a set of candidate detections, each containing a class label and predicted bounding box coordinates (see Figure 2.1, left). We limit the size of this set to be 1000s by taking *novel class* detections with high confidence scores, here we use $q > 0.8$, producing $\hat{\mathcal{Y}}_{\text{NOVEL}}$.

As demonstrated in earlier evaluations, the detector from **Novel Training** cannot detect instances from novel categories well, leaving a large number of incorrect predictions in the set of candidates, either misclassification, or imprecise bounding box coordinates. The key question now becomes, *how can we improve the precision of this list?*

2.4.2 Label Verification

In this section we take inspiration from the work on query expansion by Chum *et al.* [Chum et al. 2007], which uses spatial verification to accept or reject new instances during retrieval. The goal here is to verify the predicted class label for each candidate detection. Specifically, we consider to build a classifier for the novel categories with the very limited few-shot annotations ($\mathcal{Y}_{\text{NOVEL}}$).

Building classifiers with only a few annotations is clearly not a trivial task, as it often demands high-quality feature representations. Here, we benefit from the recent development of self-supervised models, e.g. MoCo [Kaiming He et al. 2020], SwAV [Caron et al. 2020], DINO [Caron et al. 2021], and construct k NN classifiers with the high-quality features produced from those models. In practice, this work uses the output CLS token from a ViT model [Dosovitskiy et al. 2021] trained with the self-supervised DINO [Caron et al. 2021] method, where the NN performance is shown to be particularly strong.

To perform Label Verification (see Figure 2.1, centre), we first compute features for each of the given novel class groundtruth annotations, using the self-supervised model. These features are used as the training data in our k NN classifier. Similarly, we compute features for each instance in the set of candidate detections using the same self-supervised model. In detail, to compute the feature of a given annotation/candidate detection, we first use the bounding box to crop the relevant image. This crop is then resized and passed as input to the self-supervised model.

We adopt a simple verification policy: a given candidate detection is accepted (or verified) if our k NN classifier, using cosine similarity, predicts the same class as the predicted class label from the detector. With such a verification step, we obtain a *verified* set of candidate detections with high precision with respect to classification labels.

2.4.3 Box Correction

In addition to verifying classification labels, we consider refining the bounding boxes for all remaining candidate detections in the *verified* set (see Figure 2.1, right). Taking inspiration from the Cascade R-CNN [Z. Cai and Vasconcelos 2018], we build a separate model, containing three *class-agnostic* regressors that gradually

produce a higher-quality bounding box, with each only processing boxes of similar IOUs to the groundtruth.

Specifically, during **Novel Training**, we divide the RPN proposals into three splits using different IoU thresholds, and pass the RoI features through the corresponding regressors. For example, all features pooled from $\text{IoU} > 0.3$ boxes are passed to the first regressor, features pooled from $\text{IoU} > 0.5$ boxes are passed to the second regressor and features pooled from $\text{IoU} > 0.7$ boxes are passed to the third regressor. Once this is trained, the bounding boxes for the *verified* set can be corrected, by feeding their RoI features through the three regressors in succession.

We now possess a large set of *previously unlabelled* novel instances, with high precision class labels and high-quality bounding boxes. This verified and corrected set is then used as pseudo-annotations to re-train our detector *end-to-end* on novel and base class instances.

2.5 Experiments

In this section, we first introduce the standard experimental benchmarks used in the literature [Kang et al. 2019; B. Sun et al. 2021]. After this, in Section 2.5.2 we describe our implementation, training details, and conduct extensive ablation studies on the design choices of this work in Section 2.5.3. Lastly, inheriting the best experience from the ablation studies, we compare to the existing state-of-the-art approaches in Section 2.5.4.

2.5.1 Few-Shot Object Detection Benchmarks

We follow the same benchmarks as in [Kang et al. 2019], evaluating our model on the MS-COCO [T.-Y. Lin et al. 2014] and PASCAL VOC [Everingham et al. 2010] datasets. To maintain a fair comparison, we use the same fixed lists of novel samples and data splits given in [Kang et al. 2019].

MS-COCO has 80 categories in total. In FSOD, the 20 categories present in PASCAL VOC are used as novel classes and the remaining categories are used as base classes. In this case, the benchmarks are designed for testing with $K = 10, 30$ shots, and we report standard MS-COCO metrics, namely Average

Precision (IoU=0.5 : 0.95), Average Precision (IoU=0.5) and Average Precision (IoU=0.75) on novel classes, abbreviated to nAP, nAP50 and nAP75, respectively. Our ablation studies are all conducted on the MS-COCO benchmark.

PASCAL VOC contains 20 classes, in FSOD, the data is randomly split into 15 base classes and 5 novel classes. There are three such splits and for each novel class there are $K = 1, 2, 3, 5, 10$ shots available. For this dataset we report the standard PASCAL VOC metric Average Precision (IoU=0.5) for novel classes (nAP50).

2.5.2 Implementation Details

Experiments are conducted with a standard Faster R-CNN [Ren et al. 2016], with a FPN [T.-Y. Lin et al. 2017a]. All experiments are run on 4 GPUs with batch-size 16. We use a SGD optimiser with momentum 0.9 and weight decay 10^{-4} , except for models with Transformer based backbones, in which we use the AdamW optimiser with standard hyperparameters and weight decay 0.05, following [Ze Liu et al. 2021]. The number of finetuning iterations is scaled depending on the dataset and the number of available shots.

We apply **RandomCrop** and **ColorJitter** augmentations when finetuning our FSOD detector on novel class data, unless stated otherwise. For experiments on MS-COCO, we include the **Mosaic** augmentation introduced in YOLOv4 [Bochkovskiy et al. 2020]. This augmentation helps improve detection performance on “small” objects by stitching 4 images into a 2×2 grid. **Mosaic** is not used on PASCAL VOC experiments as the dataset does not contain the same scale variation as MS-COCO.

The number of neighbours k used in Label Verification (Section 2.4.2) is determined by the number of novel instances, $k = \min\left(\left\lfloor \frac{K}{3} \right\rfloor + 1, 10\right)$. Hence for $K = 1, 2, 3, 5, 10, 30$, $k = 1, 1, 2, 2, 4, 10$. We choose a self-supervised DINO ViT-S/8 [Dosovitskiy et al. 2021; Caron et al. 2021] to construct the kNN classifier for verification, using the output CLS token as the chosen feature. For Box Correction (Section 2.4.3), we train a series of three box regression heads, where positive boxes are defined as those with $\text{IoU} > (0.3, 0.5, 0.7)$, respectively. This choice enables the correction of relatively poor initial bounding boxes.

After Candidate Sourcing, Label Verification and Box Correction, we significantly increase the number of available samples for novel categories, however it is inevitable that many novel instances will remain absent in our pseudo-annotations, and would still be treated as background. To avoid this issue, we introduce “*ignore regions*”, during end-to-end re-training, which are considered as neither foreground nor background, in practice, we treat all *unverified* novel class detections as ignore regions. We analyse these “*ignore regions*” in Appendix 2.B.

Setting	Augmentations			Finetune $\Phi_{\text{RPN}}, \Phi_{\text{ROI}}?$	Φ_{ROI} Dropout?	nAP	nAPs
	Color Jitter	Random Crop	Mosaic				
B1						13.0	5.5
B2	✓					13.2	5.6
B3		✓				13.2	5.6
B4			✓			12.8	6.1
B5	✓	✓	✓			13.8	6.8
C1	✓	✓		✓		14.5	4.3
C2	✓	✓	✓	✓		16.2	6.2
C3	✓	✓	✓	✓	✓	16.6	7.0

Table 2.3: Few-shot object detection performance on *novel* classes on the MS-COCO 30-shot task. Note that, the baseline model B1 is equivalent to a strong baseline proposed in TFA [Xin Wang et al. 2020]. nAP – novel class average precision (nAPs is for small instances only)

2.5.3 Ablation Studies

We conduct ablation studies to investigate our design choices. The following experiments are considered: *first*, we demonstrate the importance of data augmentation to yield a stronger baseline model before any of the pseudo-labelling steps; *second*, we analyse several critical components of our method and conduct thorough

Method	nAP	nAP50	nAP75	bAP	bAP50	bAP75
Baseline	16.6	30.9	15.8	29.5	46.7	32.4
+ Candidate Sourcing	18.4	34.0	18.2	31.7	53.2	35.6
+ Label Verification	21.8	40.3	21.1	31.6	53.4	35.2
+ Box Correction	25.5	42.0	27.3	33.3	53.6	36.1
Ideal Faster R-CNN [Ren et al. 2016]	43.5	67.4	46.6	36.1	56.0	39.2

Table 2.4: Ablation study on our pseudo-labelling method.

Method	Backbone	nAP		nAP50		nAP75		bAP	
		10	30	10	30	10	30	10	30
CGDP+FSCN [Yiting Li et al. 2021]	ResNet-50	11.3	15.1	20.3	29.4	-	-	-	-
Ours (Baseline)		11.4	16.6	21.2	30.9	11.1	15.8	28.4	29.5
Ours (Pseudo-Labelling)		17.6	25.5	30.9	42.0	17.3	27.3	29.7	33.3
TFA w/ cos [Xin Wang et al. 2020]	ResNet-101	10.0	13.7	19.1	24.9	9.3	13.4	32.4	34.2
FCSE [B. Sun et al. 2021]		11.9	16.4	-	-	10.5	16.2	-	-
Retentive R-CNN [Z. Fan et al. 2021]		10.5	13.8	-	-	-	-	39.2	39.3
SRR-FSD [Chenchen Zhu et al. 2021]		11.3	14.7	23.0	29.2	9.8	13.5	-	-
CME [Bohao Li et al. 2021]		15.1	16.9	24.6	28.0	16.4	17.8	-	-
DCNet [Hanzhe Hu et al. 2021]		12.8	18.6	23.4	32.6	11.2	17.5	-	-
TIP [A. Li and Zhenguo Li 2021]		16.3	18.3	33.2	35.9	14.1	16.9	-	-
QA-FewDet [Han et al. 2021]		11.6	16.5	23.9	31.9	9.8	15.5	-	-
FSOD-UP [A. Wu et al. 2021]		11.0	15.6	-	-	10.7	15.7	-	-
DeFRCN [Qiao et al. 2021]		18.5	22.6	-	-	-	-	N/A	N/A
Ours (Baseline)	Swin-T	12.1	17.8	22.0	31.6	11.8	17.7	31.9	31.8
Ours (Pseudo-Labelling)		17.8	24.5	30.9	41.1	17.8	25.0	31.9	33.0
Ours (Baseline)	Swin-T	12.6	19.0	23.3	35.3	12.1	18.3	27.2	31.5
Ours (Pseudo-Labelling)		18.6	26.1	32.1	45.0	18.5	26.8	29.2	31.1
Ours (Baseline)	Swin-S	14.4	20.3	25.8	37.2	14.4	20.2	33.0	36.2
Ours (Pseudo-Labelling)		19.0	26.8	34.1	45.8	19.0	27.5	28.7	34.8

Table 2.5: Few-shot detection performance on the MS-COCO benchmark. We report performance on the 20 novel classes of MS-COCO in the FSOD setting. Best and second-best results, per backbone, are coloured **blue** and **red**, respectively. We also report performance on base classes.

ablation studies to validate their necessity, namely, Candidate Sourcing, Label Verification, Box Correction; *third*, we show that our proposed approach maintains performance for base class detections. Note that, all ablation experiments are conducted on the MS-COCO benchmark with $K = 30$ and a ResNet-50 backbone.

Importance of Augmentations: Given we only have access to a limited number of samples for novel categories at the starting point, maximising data efficiency before any pseudo-labelling is critical. Table 2.3 presents our observations. When comparing to TFA [Xin Wang et al. 2020] as a baseline model (equivalent to Setting B1), applying **ColorJitter**, **RandomCrop** and **Mosaic** augmentations (Settings B2-B4) yields negligible performance improvements. Since almost all layers in TFA have been frozen during **Novel Training**, augmentations can only affect the classification layer of Faster R-CNN, Φ_{CLS} . The combination of these three augmentations (Setting B5), only gives a marginal improvement of 0.8nAP.

As explained in Section 2.3.2, we improve the **Novel Training** stage by also updating all RoI parameters and the RPN, *i.e.* $\Phi_{\text{RPN}}, \Phi_{\text{ROI}}$. With this simple change, all augmentations (Setting C2) substantially improve results, yielding a 3.2nAP

boost over the TFA baseline model (Setting B1). In addition, we observe a noticeable improvement in performance on small novel instances (6.2 vs. 4.3nAPs) from **Mosaic** augmentations, *e.g.* B5 vs. B1-B3, and C2 vs. C1. To further combat the overfitting issue, we add Dropout on RoI activations (Setting C3), yielding a small additional improvement of 0.4nAP.

Ablation of Pseudo-Labelling Steps: Table 2.4 shows the importance of our overall pseudo-labelling method and the contribution of each step. Using the Candidate Sourcing step only is equivalent to treating the naïve detections as pseudo-annotations, as done in other self-training works [Y.-C. Liu et al. 2021; Sohn et al. 2020]; this gives a marginal performance improvement of 1.8nAP. Removing the pseudo-annotations with incorrect class labels by Label Verification, yields an additional 3.4nAP performance boost. Lastly, Box Correction brings a 3.7nAP performance boost, in particular, such improvement is largely attributed to that from the stricter metric, *i.e.* nAP75, clearly showing the reduction of bounding box regression errors. Note that, the number and distribution of class labels for pseudo-annotations are identical before and after Box Correction; only the box coordinates of each pseudo-annotation have been changed.

Effect on Base Class Performance: While reading the performance on base classes from Table 2.4, we observe that, for the **Baseline** model, the improved performance on novel classes comes at the cost of performance on base classes, *e.g.*, bAP drops from 36.1 to 29.5. Our proposed pseudo-labelling method improves the detection of novel classes, while recovering performance on base classes.

2.5.4 Comparison to SotA

Existing methods include the meta-learning approaches: CGDP+FSCN [Yiting Li et al. 2021], CME [Bohao Li et al. 2021], TIP [A. Li and Zhenguo Li 2021], DCNet [Hanzhe Hu et al. 2021], and two-phase training works: TFA [Xin Wang et al. 2020], FCSE [B. Sun et al. 2021], Retentive R-CNN [Z. Fan et al. 2021], SRR-FSD [Chenchen Zhu et al. 2021], DeFRCN [Qiao et al. 2021], FSOD-UP [A. Wu et al. 2021], QA-FewDet [Han et al. 2021]. In particular, we note that very few works report results on the base class detection performance, and methods

like DeFRCN [Qiao et al. 2021], can actually only detect novel classes and does not maintain the ability to detect base classes as with ours.

We report two sets of results for each task, **Baseline** which makes use of augmentations and the improved training outlined in Section 2.3 and **Pseudo-Labeling** which follows the method as outlined in Section 2.4.

MS-COCO Results are shown in Table 2.5. Using a ResNet-50 backbone, our **Baseline** method, which makes extensive use of augmentations and improved **Novel Training**, outperforms many existing works for $K = 30$, reflecting the importance of our findings in Section 2.3.2. When applying our **Pseudo-Labeling** method, we set new SotA performance for $K=30$, with a performance boost of up to 2.9, 6.1 and 10.4 for nAP, nAP50 and nAP75 metrics, respectively. When $K=10$, with a ResNet-101 backbone, our **Pseudo-Labeling** method achieves state-of-the-art or second-best performance in terms of nAP, nAP50, nAP75. In terms of nAP, only DeFRCN outperforms our work. However, note that, DeFRCN is not able to detect base classes after training on novel categories, which strives for a different purpose than ours, we aim to *expand* our detector, rather than to *transfer* the detector.

In addition, while using a more powerful Transformer backbone models (Swin-T, Swin-S [Ze Liu et al. 2021]), our proposed pseudo-labelling provides additional performance boosts.

PASCAL VOC Results are shown for the three standard novel splits in Table 2.6. Incorporating our proposed **Pseudo-Labeling** method is among the top 2 best performing models in all cases (except Novel Split 2 for $K=2$), and sets SotA performance for the majority of cases.

Notably, for Novel Split 3, our **Pseudo-Labeling** method achieves SotA for all K , with a boost of up to 4.7 nAP50, when using a ResNet-101 backbone. In many cases our weaker detection backbone (ResNet-50) already achieves SotA performance. Note that, once again the transferred detector DeFRCN outperforms ours in some cases.

2.5.5 Qualitative Results

In Figure 2.2, we present the qualitative results after each step of our pseudo-labelling procedure. The top row of Figure 2.2 shows examples of Label Verification (Section 2.4.2). The first three examples demonstrate the case in which the predicted class label from our detector matches k NN classification, and so the candidate detection is verified. The last three examples show the opposite case in which candidate detections are correctly rejected. The bottom row of Figure 2.2 shows examples of Box Correction (Section 2.4.3). The first three examples show very poor bounding boxes from verified candidates (dashed blue boxes), which are drastically improved during the Box Correction step. The last three examples show acceptable bounding boxes from verified candidates (dashed blue boxes), also being improved with Box Correction. This demonstrates the ability of our Box Correction model to deal with a wide range of bounding box quality with respect to input candidate detections.

In Figure 2.3, we show precision-recall curves for some novel classes on the MS-COCO benchmark for $K=30$ using the stricter IoU=0.75 criterion. Our Pseudo-Labelling method substantially improves novel class performance, with improved precision and novel class recall. We note that for many novel classes, baseline models suffer from poor recall due to limited novel class annotations. This poor recall is improved by our pseudo-labelling method, however the poor recall of the baseline detector puts a limit on the diversity of pseudo-annotations for end-to-end retraining.

Method/Shot	Backbone	Novel Split 1					Novel Split 2					Novel Split 3				
		1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
CGDP+FSCN [Yiting Li et al. 2021] CVPR 21	ResNet-50	40.7	45.1	46.5	57.4	62.4	27.3	31.4	40.8	42.7	46.3	31.2	36.4	43.7	50.1	55.6
Ours (Baseline)		37.7	42.0	50.3	57.0	58.0	19.8	22.8	35.6	42.7	44.2	33.9	36.0	38.6	49.8	51.6
Ours (Pseudo-Labelling)		50.5	53.1	56.4	61.7	62.7	36.4	33.8	46.1	49.3	48.2	42.4	44.3	49.1	55.2	57.6
TFA w/ cos [Xin Wang et al. 2020] ICML 20	ResNet-101	39.8	36.1	44.7	55.7	56.0	23.5	26.9	34.1	35.1	39.1	30.8	34.8	42.8	49.5	49.8
FSCE [B. Sun et al. 2021] CVPR 21		44.2	43.8	51.4	61.9	63.4	27.3	29.5	43.5	44.2	50.2	37.2	41.9	47.5	54.6	58.5
Retentive R-CNN [Z. Fan et al. 2021] CVPR 21		42.4	45.8	45.9	53.7	56.1	21.7	27.8	35.2	37.0	40.3	30.2	37.6	43.0	49.7	50.1
SRF-FSD [Chenchen Zhu et al. 2021] CVPR 21		47.8	50.5	51.3	55.2	56.8	32.5	35.3	39.1	40.8	43.8	40.1	41.5	44.3	46.9	46.4
CME [Bohao Li et al. 2021] CVPR 21		41.5	47.5	50.4	58.2	60.9	27.2	30.2	41.4	42.5	46.8	34.3	39.6	45.1	48.3	51.5
DCNet [Hanzhe Hu et al. 2021] CVPR 21		33.9	37.4	43.7	51.1	59.6	23.2	24.8	30.6	36.7	46.6	32.3	34.9	39.7	42.6	50.7
TIP [A. Li and Zhenguo Li 2021] CVPR 21		27.7	36.5	43.3	50.2	59.6	22.7	30.1	33.8	40.9	46.9	21.7	30.6	38.1	44.5	50.9
QA-FewDet [Han et al. 2021] ICCV 21		42.4	51.9	55.7	62.6	63.4	25.9	37.8	46.6	48.9	51.1	35.2	42.9	47.8	54.8	53.5
FSOD-UP [A. Wu et al. 2021] ICCV 21		43.8	47.8	50.3	55.4	61.7	31.2	30.5	41.2	42.2	48.3	35.5	39.7	43.9	50.6	53.5
DeFRCN [Qiao et al. 2021] ICCV 21		53.6	57.5	61.5	64.1	60.8	30.1	38.1	47.0	53.3	47.9	48.4	50.9	52.3	54.9	57.4
Ours (Baseline)		36.0	40.1	48.6	57.0	59.9	22.3	22.8	39.2	44.2	47.8	34.3	43.4	42.9	52.0	54.5
Ours (Pseudo-Labelling)		54.5	53.2	58.8	63.2	65.7	32.8	29.2	50.7	49.8	50.6	48.4	52.7	55.0	59.6	59.6

Table 2.6: Few-shot detection performance across the three splits on the PAS-CAL VOC benchmark. Best and second-best results are coloured **blue** and **red**, respectively. Please refer to the text for discussion.

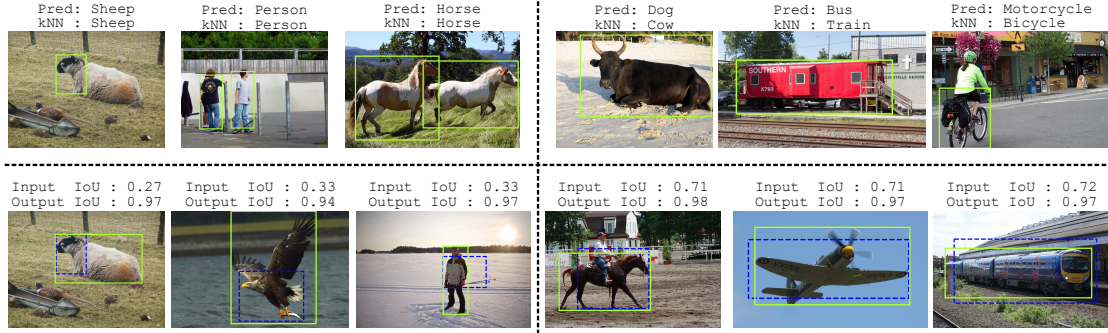


Figure 2.2: Top Left: Predicted instances which are verified during *Label Verification*; the predicted class labels from our baseline detector and our k NN classifier match. Top Right: Predicted instances which are rejected during *Label Verification*; the predicted class labels from our baseline detector (false positive) and the k NN *do not* match. Bottom Left: Verified bounding boxes with very poor quality (blue dashed) are drastically improved (lime solid) during *Box Correction*. Bottom Right: Verified bounding boxes which are acceptable (blue dashed) are further improved (lime solid).

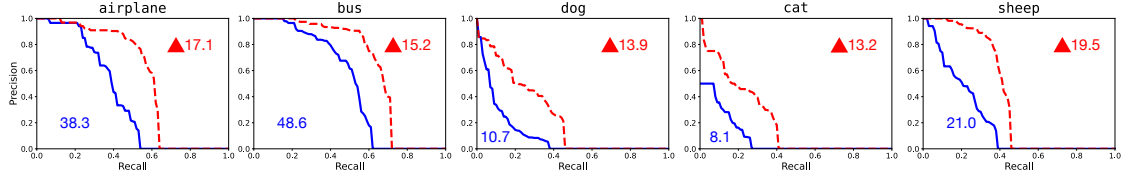


Figure 2.3: Precision-Recall curves (using the stricter $\text{IoU}=0.75$ criterion) for $K=30$ on MS-COCO showing Baseline performance (solid blue) and the substantial performance boost after making use of our Pseudo-Labeling method (dashed red). Our Pseudo-Labeling method yields improved precision and improved recall for novel classes.

2.6 Conclusion

In this paper, we tackle the problem of few-shot object detection by training on pseudo-annotations. We present two novel methods to improve the precision of the pseudo-labelling procedure: *first*, we use the given K few-shot annotations to construct classifiers to verify class labels sourced from a baseline detector; *second*, we train a specialised box correction model to drastically improve the precision of pseudo-annotation bounding box coordinates. Our method generates a large number of high-precision pseudo-annotations with precise bounding boxes, removing the class imbalance issue in FSOD. This enables re-training of our detector *end-to-end*, alleviating the identified issues around “supervision collapse” in few-shot object detectors. Furthermore, we have illustrated the importance of augmentations for FSOD, this was previously under-explored, despite augmentations being a key part of preventing overfitting. Our method achieves state-of-the-art

or second-best performance performance on both PASCAL VOC and MS-COCO benchmarks, across all number of shots.

2.7 Acknowledgements

We thank Rhydian Windsor, Sagar Vaze, Ragav Sachdeva and Guanqi Zhan for help with proofreading. This research was supported by the EPSRC CDT in AIMS EP/L015897/1 and EP/S515541/1, EPSRC Programme Grant VisualAI EP/T028572/1, and a Royal Society Research Professorship RP\R1\191132.

2.A Supervision Collapse

In Section 3 of the main paper, we experimentally demonstrate a form of “supervision collapse” in few-shot object detection; detector features trained *only on base class data* are naturally biased against novel instance detection. In this section we demonstrate two aspects of supervision collapse.

2.A.1 On Generalisability of RPNs

Figure 2.4 qualitatively illustrates the manifestation of “supervision collapse” in the Region Proposal Network (RPN) of a Faster R-CNN detector. We note that an RPN trained only with base class data struggles to propose boxes for the novel instances, due to these novel instances having been treated as background during **Base Training**.

We find that training RPN specific parameters (2 convolutional layers) on limited novel class data is adequate to drastically improve the proposal of object regions containing novel instances. This improvement is shown qualitatively in Figure 2.4 and quantitatively in Table 1 of the main paper.

2.A.2 On Transferability of Base Features

Figure 2.5 qualitatively demonstrates the other manifestation of “supervision collapse”; after training the second stage of Faster R-CNN as in [Xin Wang et al. 2020] i.e. only training the final classification layer on novel class data, features in the second stage of the detector (which take RPN proposals as input), incorrectly classify Regions-of-Interest (RoIs) containing novel instances as “background”, i.e. the detector features in the second stage do not contain discriminative information to classify most novel instances, even in the cases where the object regions proposed by the RPN are very good.

We find that training parameters specific to the second stage of the detector, (Φ_{CLS} , Φ_{ROI}), can moderately alleviate this form of “supervision collapse” such that the resultant detector can be used for Candidate Sourcing in our Pseudo-Labeling method. Figure 2.5 shows examples in which RoIs are correctly classified after *our Novel Training*.



Figure 2.4: **Supervision collapse for the RPN.** Top 100 RPN proposals before and after **Novel Training**. In each pair of rows, the upper row shows the top 100 object proposals (blue) of an RPN trained *only on base class data* and the lower row shows the top 100 object proposals (blue) after RPN specific parameters (Φ_{RPN}) have been trained on limited novel class data. Groundtruth novel instances are shown in dashed lime. We find an RPN trained only on base class data cannot propose many simple novel instances and that training RPN specific parameters on limited novel class data leads to an RPN which is substantially better at proposing object regions containing novel instances.

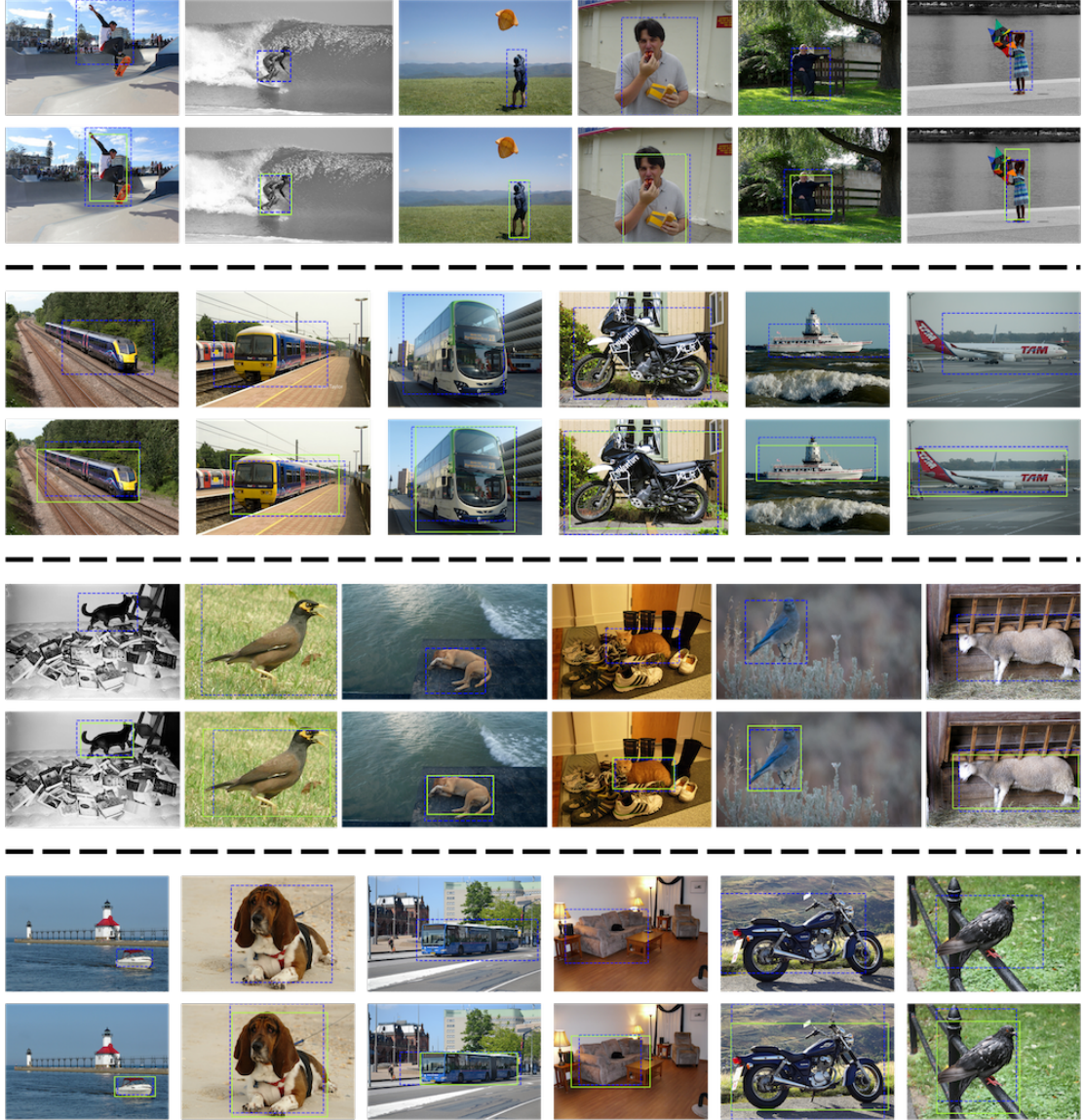


Figure 2.5: **Supervision collapse for RoI classification.** Comparing RoI classification between TFA [Xin Wang et al. 2020], where only Φ_{CLS} is trained on limited novel class data, with *our Novel Training*, which additionally trains Φ_{ROI} on limited novel class data, for cases where the object proposal is very good. In each pair of rows, the upper row shows the best object region (dashed blue) proposed by the RPN in TFA and the lack of foreground detection (false negative) from the second stage, and the lower row shows the best object region (dashed blue) proposed by the RPN after *our Novel Training* and the correct foreground detection (solid lime). We find that TFA incorrectly classifies positive (with respect to novel classes) object regions (dashed blue) as background, whereas *our Novel Training* allows positive object regions (dashed blue) to be correctly classified as a novel instance (solid lime).

2.B Ignore Regions

As mentioned in Section 5.2 of the main paper, we introduce the “*ignore regions*” during end-to-end training after making use of our Pseudo-Labelling method. Figure 2.6 shows the impact of ignore regions. During end-to-end training, we ensure no RoIs which have $\text{IoU} > 0.5$ with an ignore region are sampled for training the second-stage of the detector; RoIs which have $\text{IoU} > 0.5$ with an ignore region are neither pushed into foreground (positive) or background (negative) in the second-stage of the detector. Only with respect to the RPN during end-to-end training, “*ignore regions*” are treated as foreground i.e. the RPN after end-to-end training has learnt to propose regions-of-interest covering ignore regions. Using ignore regions prevents the undiscovered novel instances from being treated as background during end-to-end training (as occurs during **Base Training**) – there are undiscovered novel instances because we use a score threshold of 0.8 during Candidate Sourcing to yield high-precision pseudo-annotations. We use all detections from our baseline detector, with novel class labels, which are *unverified* after Label Verification as ignore regions – this includes detections with score less than 0.8.

An ablation study on the utility of ignore regions is presented in Table 2.7, using the MS-COCO 30-shot object detection benchmark. Using ignore regions improves performance by 4.4, 3.5 and 4.7 on novel categories, *i.e.* nAP, nAP50 and nAP75, respectively, when compared to not using ignore regions.

Method	Ignore Regions?	Metrics		
		nAP	nAP50	nAP75
Ours (Pseudo-Labelling)	✗	21.1	38.5	22.6
Ours (Pseudo-Labelling)	✓	25.5	42.0	27.3

Table 2.7: Ablation study on the utility of ignore regions.



Figure 2.6: Comparing images in the training set with and without ignore regions. In each pair of rows, the upper row shows images with pseudo-annotations found by our Pseudo-Labeling method (solid lime), with many novel instances not covered, the lower row shows the same images additionally showing the ignore regions (solid blue). Using ignore regions covers many novel instances which are not found by our Pseudo-Labeling method, preventing these novel instances being treated as background during end-to-end training.

2.C More Qualitative Results

In this section we present more qualitative results from our Pseudo-Labeling method on the MS-COCO benchmark with $K=30$.

2.C.1 Label Verification

In Figure 2.7, we present additional qualitative results after the *Label Verification* step of our Pseudo-Labeling method. The first four rows in Figure 2.7 show examples of Label Verification (Section 4.2 of the main paper) in which the predicted class label from our baseline detector (after **Novel Training**) matches the class label from our k NN classifier. These candidate detections are *verified* and used as inputs during *Box Correction* after which they are included as pseudo-annotations for end-to-end training. The final two rows in Figure 2.8 show examples of Label Verification in which the predicted class label from our baseline detector *does not* match the class label from our k NN classifier. These candidate detections are *not verified* and rejected from our Pseudo-Labeling method.

2.C.2 Box Correction

In Figure 2.8, we present additional qualitative results after the *Box Correction* step of our Pseudo-Labeling method. The first five rows in Figure 2.8 show examples of Box Correction (Section 4.3 of the main paper) in which the bounding box coordinates of *verified* candidate detections are poor (dashed blue) i.e. they have low intersection-over-union (IoU) with the groundtruth instance. During *Box Correction* these poor bounding boxes are drastically improved to near perfect bounding boxes (solid lime). The bottom row in Figure 2.7 show examples of Box Correction in which the bounding box coordinates of the verified candidate detections (dashed blue) have acceptable IoU with groundtruth instances. During *Box Correction* these acceptable bounding boxes are further improved to near perfect bounding boxes (solid lime).

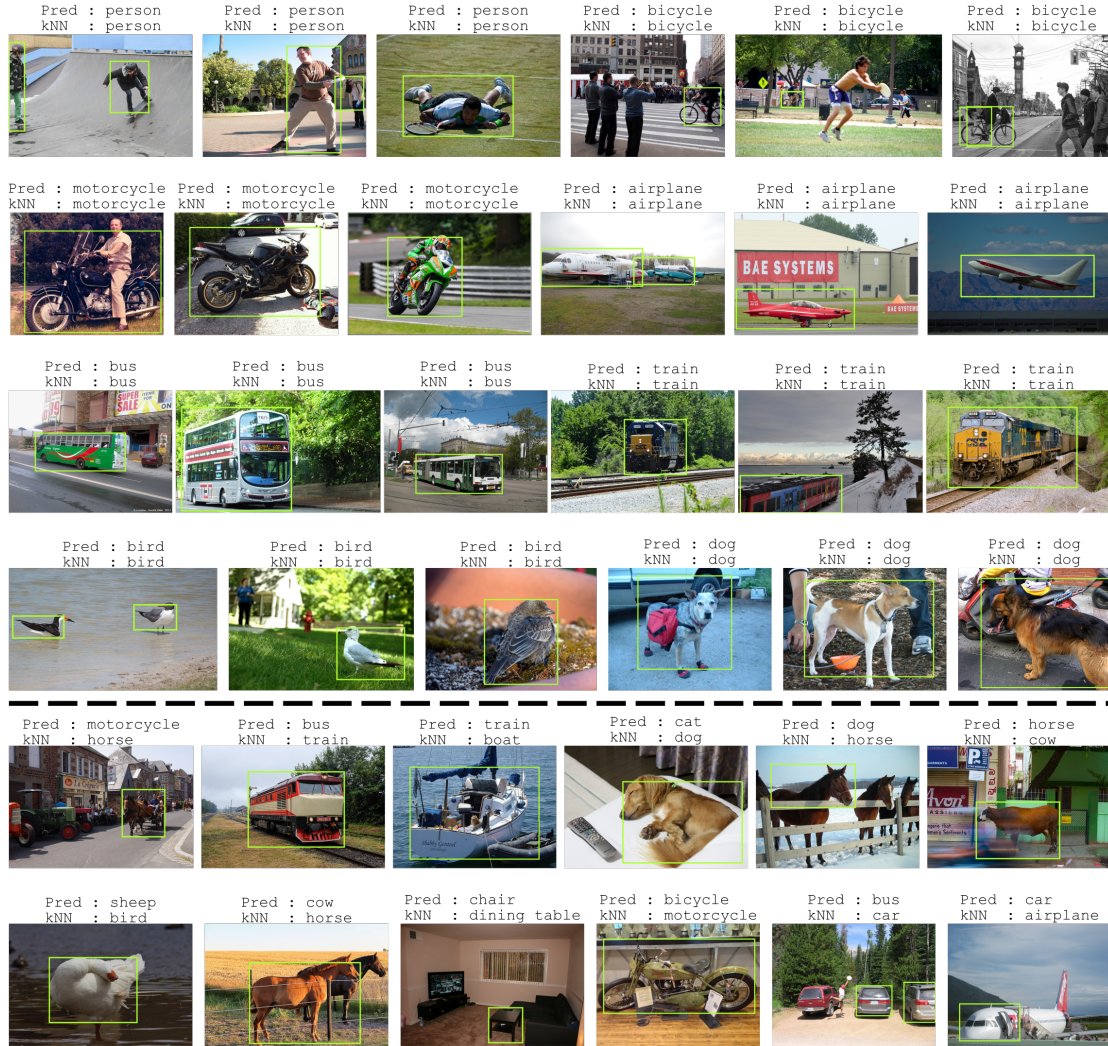


Figure 2.7: More examples of *Label Verification*. Top Four Rows: Candidate detections which *are verified* during *Label Verification*; the predicted class labels from our baseline detector and our *kNN* classifier match. Bottom Two Rows: Candidate detections which *are not verified* (or rejected) during *Label Verification*; the predicted class labels from our baseline detector (false positive) and our *kNN* classifier *do not* match.

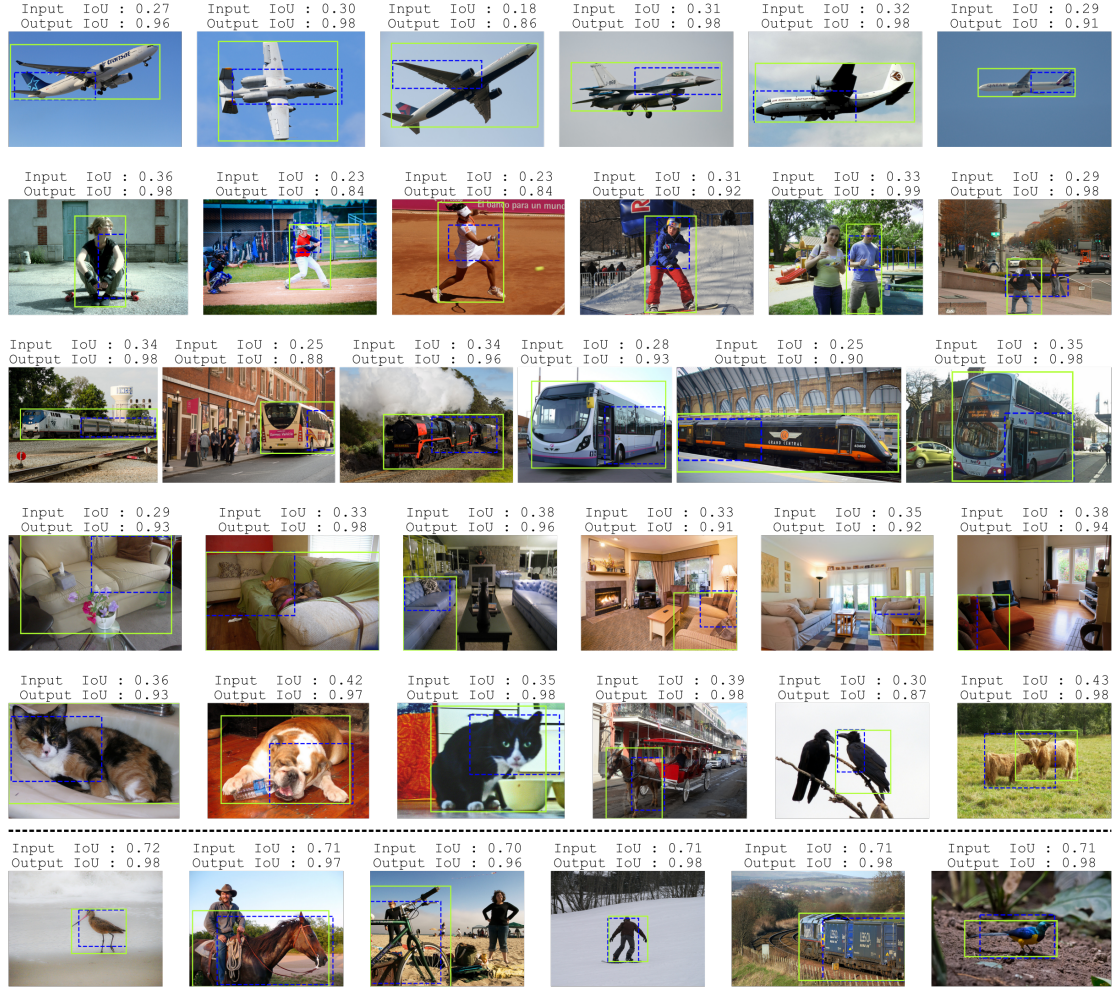


Figure 2.8: More examples of *Box Correction*. First Five Rows: Verified bounding boxes with poor quality/IoU with groundtruth (dashed blue) are drastically improved (solid lime) during *Box Correction*. Bottom Row: Verified bounding boxes with acceptable quality/IoU with groundtruth (dashed blue) are further improved (solid lime) during *Box Correction*.

2.D Quantitative Results

In this section we present quantitative results of each step of our Pseudo-Labeling method on the MS-COCO benchmark with $K=30$. We show precision-recall curves using the various IoU criteria used in MS-COCO.

Precision-recall curves when using the IoU=0.5, IoU=0.75 and IoU=0.5:0.95 MS-COCO criteria are shown in Figure 2.9, 2.10, 2.11, respectively.

For the vast majority of novel classes, adding our Pseudo-Labeling method improves performance over our baseline detector performance, regardless of which MS-COCO criteria is used. However, we note that the effect of each step is not uniform across different novel classes and metrics. For example, considering the **person** class, using Candidate Sourcing only is the best performing model when using IoU=0.5 due to higher recall, but when using IoU=0.75 and IoU=0.5:0.95 our full Pseudo-Labeling method performs best.

In terms of IoU=0.5:0.95, which is the main MS-COCO criterion, Figure 2.9 shows that different classes see the largest performance boost from different steps in our Pseudo-Labeling process. For example, the largest performance boost comes from Label Verification from some classes e.g. **bird**, **sheep**, **cat**; these classes suffer from many false positives with respect to class labels after Candidate Sourcing, which are removed during Label Verification yielding better performance. On the other hand, for other classes the largest performance boost comes from Box Correction e.g. **airplane**, **train**, **person**; these classes suffer from many false positives with respect to bounding box coordinates after Candidate Sourcing, which are corrected during Box Correction yielding better performance.

We note that in some rare cases an individual Pseudo-Labeling step may harm performance, e.g. adding Label Verification for **person** reduces performance by 2.4nAP compared to using Candidate Sourcing only, and adding Box Correction for **bicycle** reduces performance by 1.5nAP compared to using Candidate Sourcing and Label Verification only.

Moreover, we note that our Pseudo-Labeling method yields negligible performance improvements for the **bicycle** and **chair** classes yielding only a 0.1 and 0.3nAP boost, respectively, over our baseline detector.

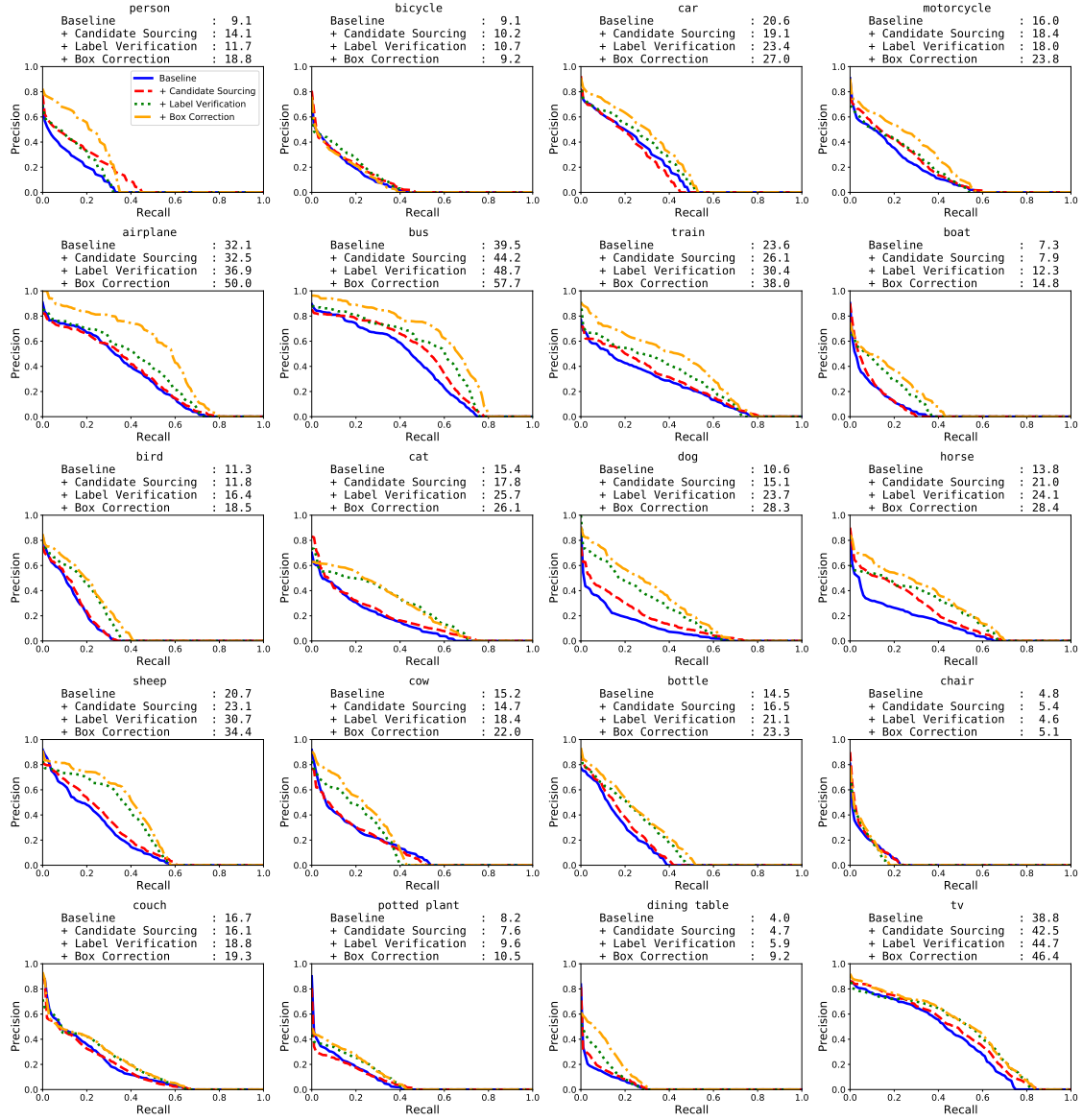


Figure 2.9: **Precision-Recall curves for novel classes with IoU=0.5:0.95.** Baseline curves are shown in solid blue. Curves are shown after adding Candidate Sourcing, Label Verification and Box Correction in dashed red, dotted green and dash-dotted orange, respectively. The nAP value for each step of our method is shown in the title of each subplot.

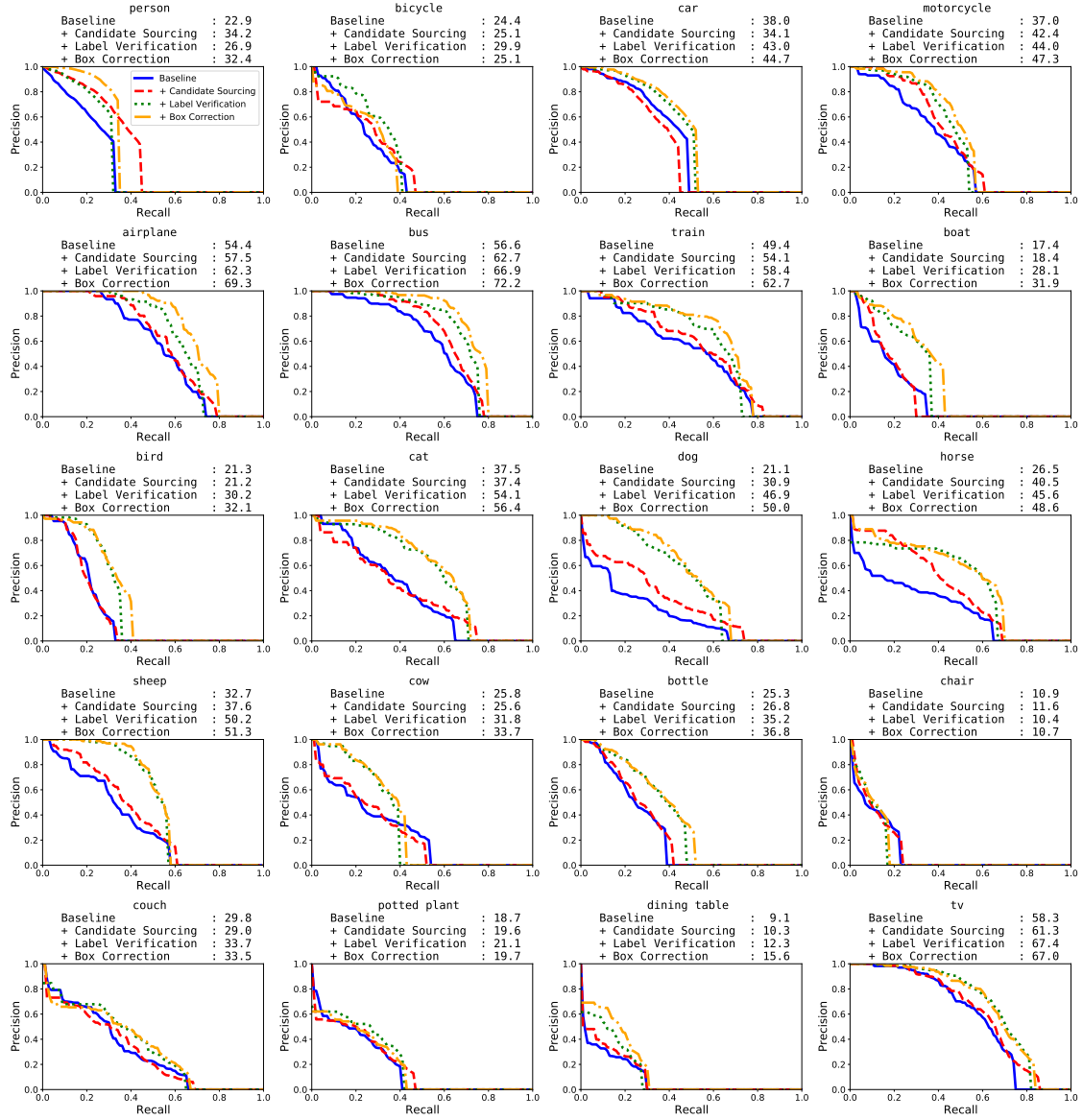


Figure 2.10: **Precision-Recall curves for novel classes with $\text{IoU}=0.5$.** Baseline curves are shown in solid blue. Curves are shown after adding Candidate Sourcing, Label Verification and Box Correction in dashed red, dotted green and dash-dotted orange, respectively. The nAP50 value for each step of our method is shown in the title of each subplot.

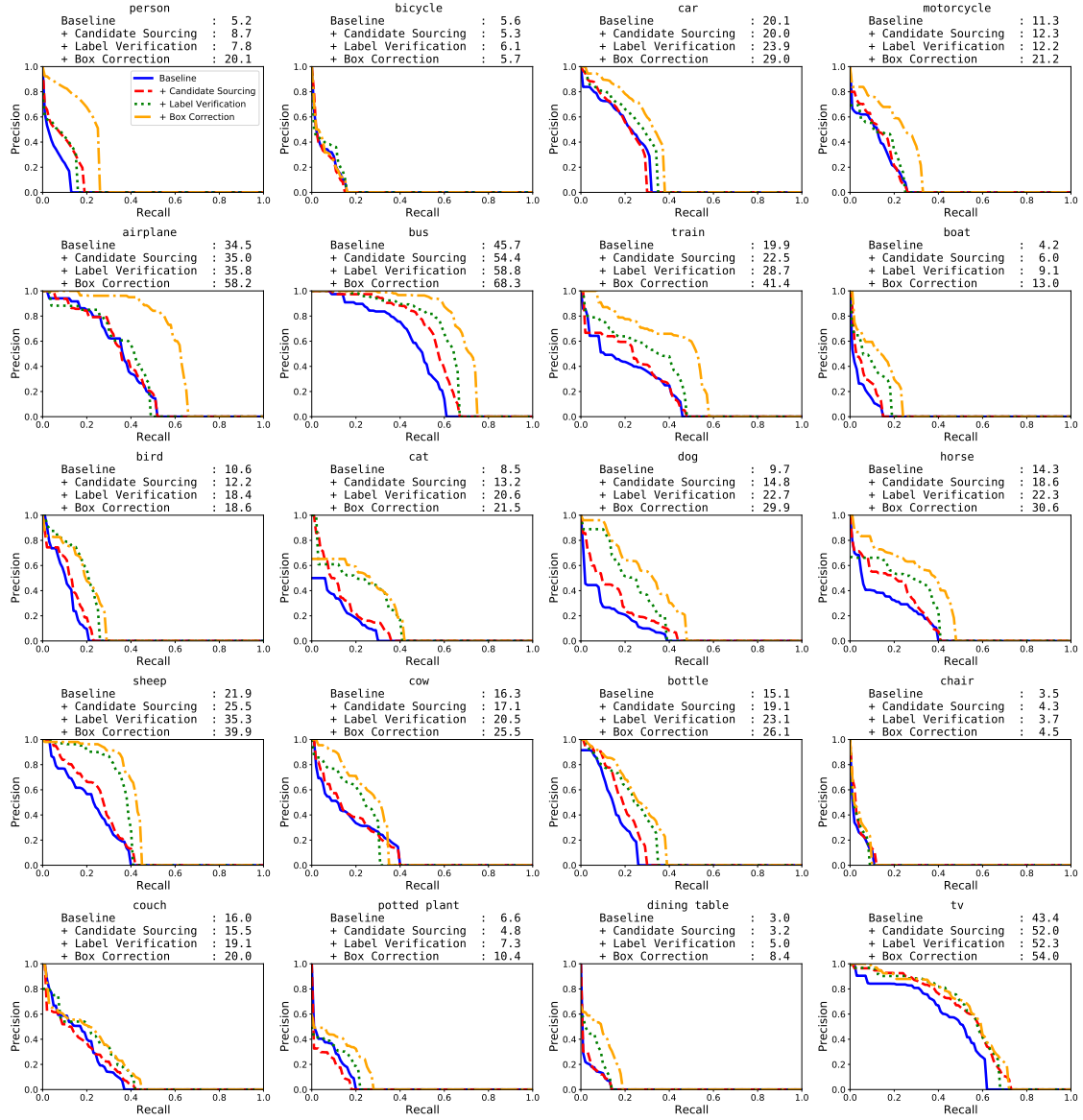


Figure 2.11: **Precision-Recall curves for novel classes with $\text{IoU}=0.75$.** Baseline curves are shown in solid blue. Curves are shown after adding Candidate Sourcing, Label Verification and Box Correction in dashed red, dotted green and dash-dotted orange, respectively. The nAP75 value for each step of our method is shown in the title of each subplot.

2.E Limitations

There are two main limitations of our work: *first*, our work follows prior work and performs benchmark experiments on MS-COCO [T.-Y. Lin et al. 2014] and PASCAL VOC [Everingham et al. 2010], which contain 80 and 20 classes, respectively. In future work, we will benchmark the idea on LVIS [A. Gupta et al. 2019], which contains 1230 classes, and better characterise few-shot object detection methods. *Second*, in this work, we only perform one iteration of pseudo-labelling with k NN classifier. Given our pseudo-annotations are noisy, one can make use of noisy label learning to effectively incorporate our pseudo-annotations as training data for training more powerful classifiers in the Label Verification step and obtaining more pseudo-annotations.

2.F Ethical Concerns

Our proposed method allows object detectors to detect novel categories with limited access to groundtruth annotations. Moreover, our work contains a Pseudo-Labeling method which enables object detectors to effectively re-train on (noisy) pseudo-annotations. Our work could be applied to numerous areas e.g. public surveillance and autonomous driving. While not our intention, our method could be used by agents with malicious purposes for detection with limited data. We think robust public debate and suitable regulations can reduce the risk of malicious applications of our work.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor (**only required where there isn't already a statement of contribution within the paper itself**).

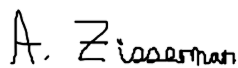
Title of Paper	Label, Verify, Correct: A Simple Few Shot Object Detection Method
Publication Status	☒ Published ☐ Accepted for Publication ☐ Submitted for Publication ☐ Unpublished and unsubmitted work written in a manuscript style
Publication Details	Kaul, P., Xie, W., & Zisserman, A. (2022). Label, verify, correct: A simple few shot object detection method. In <i>Proceedings of the IEEE/CVF conference on computer vision and pattern recognition</i> (pp. 14237-14247).

Student Confirmation

Student Name:	Prannay Kaul		
Contribution to the Paper	First Author Contribution: • conception of research ideas • design and implementation of models • running of large-scale experiments • writing and presentation of the paper		
Signature 	Date	12th January 2024	

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Andrew Zisserman		
Supervisor comments The descriptions of the contributions is accurate.		
Signature 	Date	12th January 2024

This completed form should be included in the thesis, at the end of the relevant chapter.

Chapter 3

Multi-Modal Classifiers for Open-Vocabulary Object Detection

Following the work on few-shot object detection, in which few labelled examples are available for each novel object category, we consider the more challenging problem of open-vocabulary object detection (OVOD). This considers the problem of learning to detect novel object categories with no annotated examples—relying on other methods to specify a novel category of interest.

The paper has been published at the International Conference on Machine Learning (ICML), 2023.

Multi-Modal Classifiers for Open-Vocabulary Object Detection

Prannay Kaul¹ Weidi Xie^{1,2} Andrew Zisserman¹

¹Visual Geometry Group, University of Oxford

²CMIC, Shanghai Jiao Tong University

<http://www.robots.ox.ac.uk/vgg/research/mm-ovod/>

Abstract

The goal of this paper is open-vocabulary object detection (OVOD) — building a model that can detect objects beyond the set of categories seen at training, thus enabling the user to specify categories of interest at inference without the need for model retraining. We adopt a standard two-stage object detector architecture, and explore three ways for specifying novel categories: via language descriptions, via image exemplars, or via a combination of the two. We make three contributions: *first*, we prompt a large language model (LLM) to generate informative language descriptions for object classes, and construct powerful text-based classifiers; *second*, we employ a visual aggregator on image exemplars that can ingest any number of images as input, forming vision-based classifiers; and *third*, we provide a simple method to fuse information from language descriptions and image exemplars, yielding a multi-modal classifier. When evaluating on the challenging LVIS open-vocabulary benchmark we demonstrate that: (i) our text-based classifiers outperform all previous OVOD works; (ii) our vision-based classifiers perform as well as text-based classifiers in prior work; (iii) using multi-modal classifiers perform better than either modality alone; and finally, (iv) our text-based and multi-modal classifiers yield better performance than a fully-supervised detector.

3.1 Introduction

In this paper, we consider the problem of open-vocabulary object detection (OVOD), which aims to localise and classify visual objects beyond the categories seen at training time. One may consider its usefulness from the perspective of online inference, when users want to freely specify categories of interest at inference time without the need or ability to re-train models. To specify categories of interest, three obvious ways exist, namely: (1) text-based, e.g. name the category or describe it in text form; (2) vision-based, e.g. give image examples; (3) multi-modal, e.g. indicate the category jointly with text and image.

Existing works [Bansal et al. 2018; Gu et al. 2022; Zareian et al. 2021; X. Zhou et al. 2022; C. Feng et al. 2022] have explored replacing the learnt classifiers in a traditional detector with text embeddings, that are computed by passing the class name into a pre-trained text encoder with manual prompts, such as “a photo of a dalmatian”, however, this design can be sub-optimal from three aspects: *first*, the discriminative power of generated text embeddings relies entirely on the internal representation of the pre-trained text encoder, potentially leading to lexical ambiguities — e.g. “nail” can either refer to “the hard surface on the tips of the fingers” or “a small metal spike with a flat tip hammered into wood to form a joint” — simply encoding the class name will not be able to distinguish the two concepts; *second*, the class name for objects of interest may be unknown to the user, while exemplar images can be easily acquired — e.g. “dugong” refers to a herbivorous marine mammal with an adorable, plump appearance, a dolphin tail, round head and downward snout (an example is in Section 3.I of the Appendix); *third*, in cases where multi-modal information is preferable to specify the category of interest, e.g. a species of butterfly with a distinctive wing pattern — language descriptions can be unsuitably long to capture all the intricacies of a given category, while an exemplar image can “tell a thousand words” and act as an effective complement to text.

To tackle these limitations, we propose a multi-modal open-vocabulary object detector, with the classifier of the detector for a particular category being constructed via natural language descriptions, image exemplars or a combination of the two. Specifically, we establish an automatic approach for sourcing visual descriptions

of the object categories, by prompting a large language model with questions, e.g. “What does a dalmatian look like?”, yielding “A dalmatian is typically a large dog with a short coat of black spots on a white background”. Such a description provides additional visual cues to enhance the discriminative power of the classifier generated from a text encoder. For cases where collecting suitable informative language descriptions may be difficult or require unnecessarily long descriptions to establish the differences between classes, e.g. the dog breeds “pug” and “bulldog” have similar descriptions, we can generate classifiers from image exemplars — RGB images of the class of interest. Finally, we suggest a simple method to fuse both language descriptions and image exemplars, yielding multi-modal classifiers that perform better than either modality individually.

We explore the issue of how best to combine the set of language descriptions and the set of image exemplars into a classifier, by comparing the performance of aggregation methods on a standard detector architecture. By evaluating on the challenging LVIS [A. Gupta et al. 2019] open-vocabulary object detection benchmark we show that: (i) our automated method for sourcing rich natural language descriptions yields text-based classifiers superior to those of previous work that rely entirely on the class name; (ii) vision-based classifiers can be effectively constructed by a visual aggregator, enabling novel categories to be detected by specifying image exemplars; (iii) natural language descriptions and image exemplars can be simply combined to produce multi-modal classifiers, which perform better than either modality individually, and achieve superior results to existing approaches.

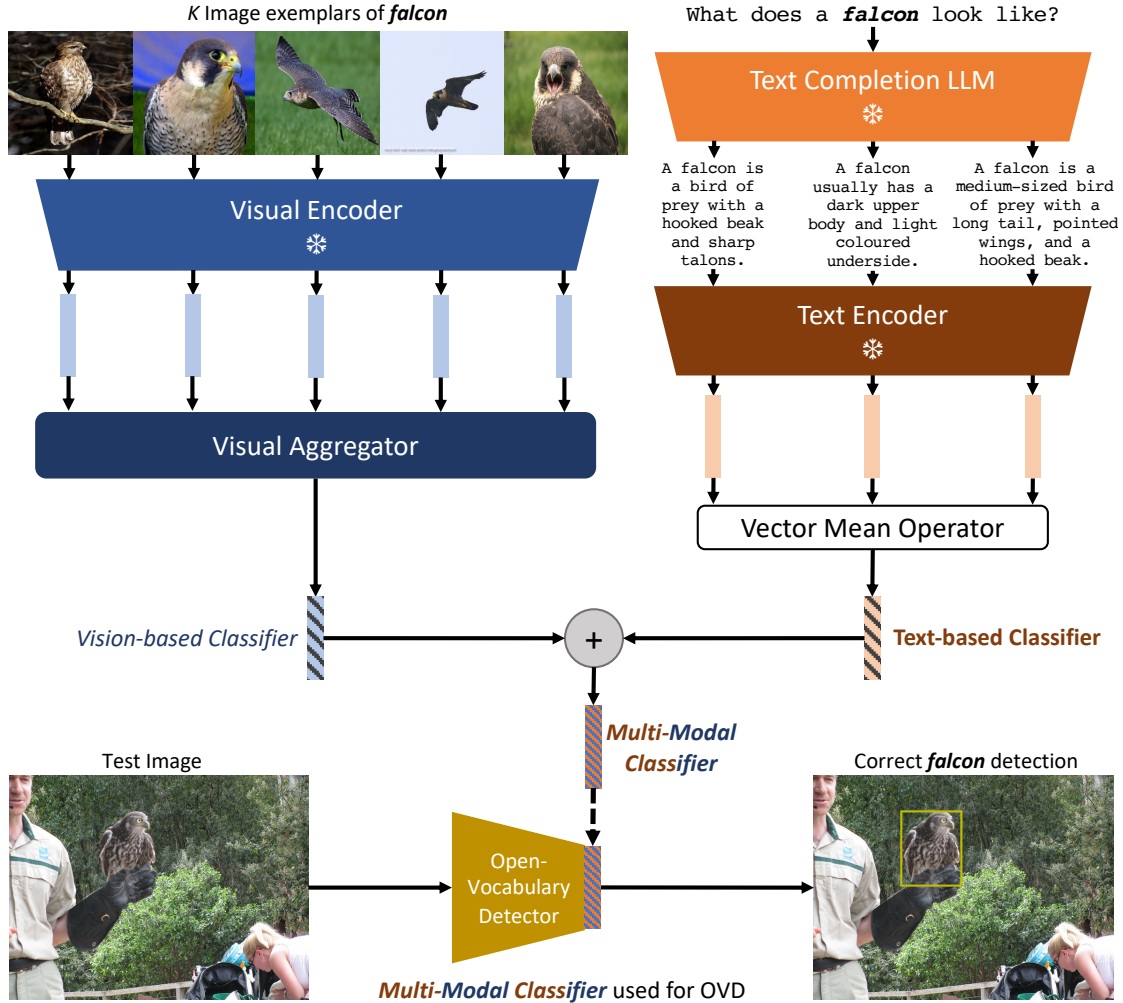


Figure 3.1: Overview of the architecture for generating text-based, vision-based, or multi-modal classifiers for OVOD. Vision (top left): A frozen visual encoder ingests image exemplars of **falcon** producing an embedding per exemplar. A trained aggregator takes these embeddings as input and produces a *vision-based classifier*. Text (top right): A text completion LLM is prompted to give descriptions of a **falcon** which are then encoded by a text encoder and averaged yielding a *text-based classifier*. Multi-Modal (middle): *Multi-Modal classifiers* are generated by adding the vision-based and text-based classifiers together. OVOD (bottom): The multi-modal classifier is used to detect the **falcon** in a standard model. Note, all three types of classifier: vision-based, text-based and multi-modal, can be used on the detector head for OVOD.

3.2 Related Work

Closed-Vocabulary Object Detection is one of the classical computer vision problems, making a full overview here impossible. Therefore, we outline some key milestones. In general, modern object detection methods can be cast into two sets: one-stage and two-(multi-)stage detectors. *One-stage* detectors directly classify and regress bounding boxes by either densely classifying a set of predefined anchor boxes [Redmon et al. 2016; Redmon and Farhadi 2018; W. Liu et al. 2016; T.-Y. Lin et al. 2017b; Tan et al. 2020], each which may contain an object, or densely searching for geometric entities of objects e.g. corners, centre points or boxes [Law and Jia Deng 2018; X. Zhou et al. 2019; Tian et al. 2019]. Conversely, most *two-stage* detectors first propose class-agnostic bounding boxes that are pooled to fixed size region-of-interest (RoI) features and classified by a sub-network in the second stage [Girshick 2015; Ren et al. 2016; Yanghao Li et al. 2019]. Two-stage detectors are extended to *multi-stage* detectors in which the additional stages refine predictions made by the previous stage [Z. Cai and Vasconcelos 2018; Kai Chen et al. 2019; X. Zhou et al. 2021]. A unique work in this area is that of [Carion et al. 2020] which uses the Transformer architecture [Vaswani et al. 2017] to treat object detection as a set prediction problem. **Note that**, the classifiers in these object detectors are jointly learnt on a training set, therefore only objects seen at training time can be detected during inference time, thus termed closed-vocabulary object detection.

Open-Vocabulary Object Detection goes beyond closed-vocabulary object detection and enables users to expand/change the detector vocabulary at inference time, without the need for model re-training. Recently, OVOD has seen increased attention and progress primarily driven by the emergence of large-scale vision-language models (VLMs) e.g. CLIP and ALIGN [Radford et al. 2021; Jia et al. 2021], which jointly learn image and natural language representations.

ViLD [Gu et al. 2022] distills representations from VLMs. First, groundtruth bounding boxes are used to crop an image and an embedding for the box is sourced from a frozen VLM image encoder. An object detection model is learnt by matching overlapping region-of-interest (RoI) features with the embedding for the relevant box from the VLM image encoder using a L1 reconstruction loss.

RegionCLIP [Zhong et al. 2022] uses image-caption data to construct region-wise pseudo-labels, followed by region-text contrastive pre-training before transferring to detection. GLIP and MDETR [L. H. Li et al. 2022; Kamath et al. 2021] use captions to cast detection as a phrase grounding task and use early fusion between the image and text modalities, increasing complexity. OVR-CNN [Zareian et al. 2021] uses large image-caption data to pre-train a detector to learn a semantic space and finetunes on smaller detection data. OWL-ViT [Minderer et al. 2022] follows OVR-CNN but makes use of large transformer models and even larger image-caption data. OV-DETR [Zang et al. 2022] modifies the DETR framework for closed-vocabulary object detection [Carion et al. 2020] to make it suitable for the open-vocabulary setting. We note that OV-DETR can condition object detection on image exemplars, but only provides some qualitative examples, whereas we quantitatively benchmark our method using vision-based classifiers. Detic and PromptDet [X. Zhou et al. 2022; C. Feng et al. 2022] improve open-vocabulary detection by making use of image classification data to provide weak supervision on a large set of categories. Our work uses Detic as a starting point in experiments, and we investigate different methods for constructing the classifiers.

Low-shot Object Detection. Despite garnering less attention in recent literature, some low/few-shot object detection works make use of image-conditioned object detection [Kang et al. 2019; Hsieh et al. 2019; Osokin et al. 2020; D.-J. Chen et al. 2021] in which image exemplars of novel categories are encoded at inference time and used to detect novel category instances. These works focus on architectural advances usually leveraging attention between the novel class image exemplars and the inference time image [D.-J. Chen et al. 2021; Hsieh et al. 2019].

There are an increasing number of low/few-shot object detection works based on finetuning detector parameters on limited groundtruth data for novel categories [Xin Wang et al. 2020; B. Sun et al. 2021; Qiao et al. 2021; Kaul et al. 2022]. These works finetune the detector on limited groundtruth novel instances and so are not related to open-vocabulary object detection using vision-based classifiers, where no novel instances are available for re-training/finetuning.

Natural Language for Classification. Natural language is a rich source of semantic information for classification. CLEVER [Choudhury et al. 2021] matches descriptions of images in simple text form with descriptions from expert databases

e.g. Wikipedia to perform fine-grained classification. ZSLPP [Elhoseiny et al. 2017] extracts visual information from large-scale text to identify parts of objects and perform zero-shot classification. The work by [Menon and Vondrick 2023] uses class descriptions from GPT-3 [Brown et al. 2020] for classification, analysing which parts of the description contribute to classification decisions. CuPL [Pratt et al. 2022] uses a GPT-3 model to provide detailed descriptions enabling improved zero-shot image classification. The learnings from this work inform our use of natural language descriptions sourced from LLMs.

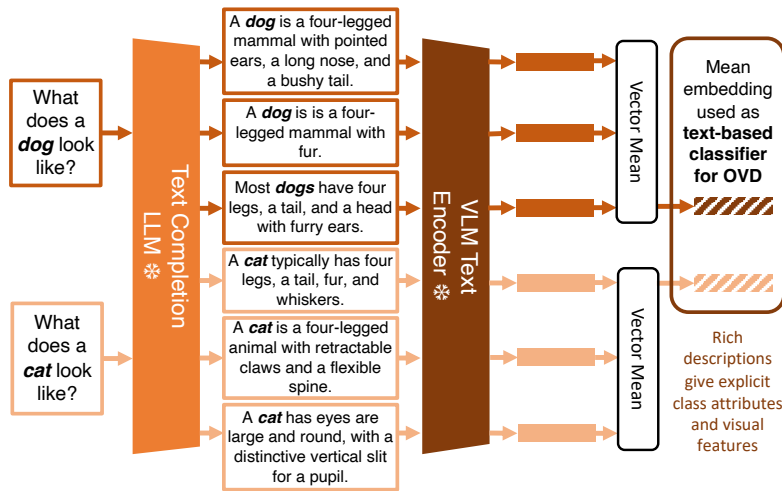


Figure 3.2: Generating powerful text-based classifiers. A LLM (GPT-3) is used to generate multiple rich descriptions of the class of interest. These descriptions are then encoded with the CLIP [Radford et al. 2021] VLM text encoder. The descriptions are more informative than the simple phrases, such as “(a photo of) a **dog**” or “(a photo of) a **cat**”, used in previous work such as Detic and ViLD. Additional examples of class descriptions are given in the Appendix (Section 3.F).

3.3 Method

In this section, we start by providing background on open-vocabulary object detection (OVOD), then outline the proposed methods for constructing classifiers from language descriptions of a category (text-based classifiers, Section 3.3.2) and image exemplars (vision-based classifiers, Section 3.3.3). Our final method combines the classifiers found from language descriptions and image exemplars, yielding multi-modal classifiers (Section 3.3.4).

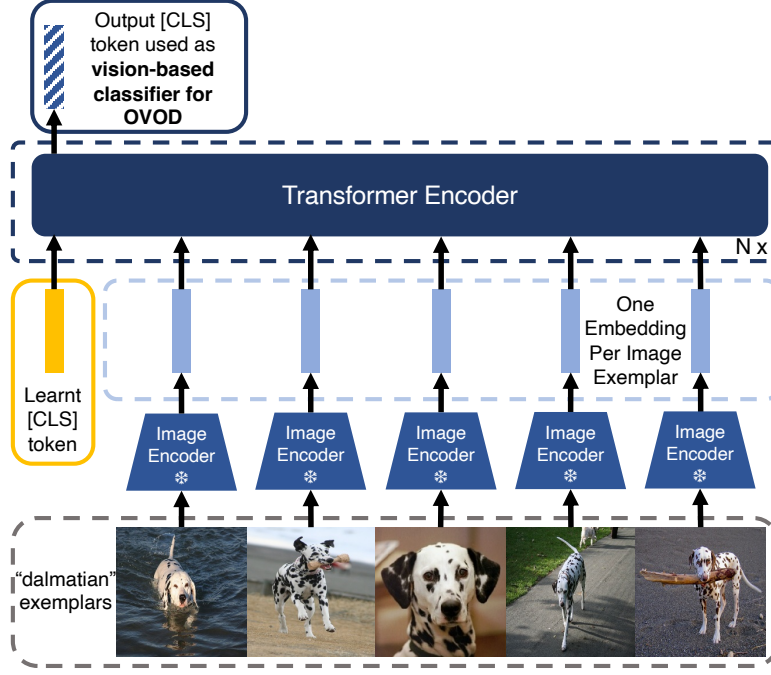


Figure 3.3: Generating an OVOD vision-based classifier from a set of image exemplars. A stack of transformer blocks is used to combine embeddings of multiple exemplars belonging to the same category.

3.3.1 Preliminaries

Problem Scenario. Given an image ($\mathbf{I} \in \mathbb{R}^{3 \times H \times W}$) fed input to an open-vocabulary object detector, two outputs are generally produced: (1) classification, in which a class label, $c_j \in \mathcal{C}^{\text{TEST}}$, is assigned to the j^{th} predicted object in the image, and $\mathcal{C}^{\text{TEST}}$ refers to the category vocabulary desired at inference time; (2) localisation, with bounding box coordinates, $\mathbf{b}_j \in \mathbb{R}^4$, denoting the location of the j^{th} predicted object. In accordance with the setting introduced by Detic [X. Zhou et al. 2022], two datasets are used at training time: a detection dataset, $\mathcal{D}^{\text{DET}}$, containing bounding box coordinates, class labels and associated images, addressing a category vocabulary, $\mathcal{C}^{\text{DET}}$; and an image classification dataset, $\mathcal{D}^{\text{IMG}}$, containing images with class labels only, addressing a category vocabulary, $\mathcal{C}^{\text{IMG}}$. In the most general case there are no restrictions on the overlap or lack thereof between the sets $\mathcal{C}^{\text{TEST}}$, $\mathcal{C}^{\text{DET}}$ and $\mathcal{C}^{\text{IMG}}$.

Architecture Overview. In this work, we make use of a popular multi-stage detector based on CenterNet2 [X. Zhou et al. 2021] as done in Detic. This detector, with outputs $\{c_j, \mathbf{b}_j\}_{j=1}^M$, can be formulated as (for simplicity we consider the two-

stage variant below):

$$\{f_j\}_{j=1}^M = \Phi_{\text{ROI}} \circ \Phi_{\text{PG}} \circ \Phi_{\text{ENC}} (\mathbf{I}) \quad (3.1)$$

$$\{\mathbf{b}_j, c_j\}_{j=1}^M = \{\Phi_{\text{BBOX}}(f_j), \Phi_{\text{CLS}} \circ \Phi_{\text{PROJ}}(f_j)\}_{j=1}^M \quad (3.2)$$

where, each input image is first sequentially processed by a set of operations: an image encoder (Φ_{ENC}); a proposal generator (Φ_{PG}); a region-of-interest (RoI) feature pooling module (Φ_{ROI}), yielding a set of RoI features, $\{f_j\}_{j=1}^M$. The RoI features are processed by a bounding box module (Φ_{BBOX}) to infer position of objects, $\{\mathbf{b}_j\}_{j=1}^M$. Additionally, the RoI features are processed by a classification module, consisting of a linear projection (Φ_{PROJ}), and C classification vectors or classifiers (Φ_{CLS}), yielding a set of class labels, $\{c_j\}_{j=1}^M$ (C is the size of the category vocabulary).

In closed-vocabulary object detection all parameters listed above are learnt during training on $\mathcal{D}^{\text{DET}}$. While in the open-vocabulary scenario, the classifiers (Φ_{CLS}) *are not* learnt during training but are instead generated separately from an alternative source, e.g. a pre-trained text encoder. This allows $\mathcal{C}^{\text{TEST}} \neq \mathcal{C}^{\text{DET}}$, as the classifiers, Φ_{CLS} , for a specific set of user defined classes can be generated at inference time. In the following sections, we describe different options for constructing such classifiers: from natural language, from image exemplars, or from a combination of the two.

3.3.2 Text-based Classifiers from Language Descriptions

Existing OVOD approaches, e.g. Detic [X. Zhou et al. 2022] and ViLD [Gu et al. 2022], make use of simple text-based classifiers by encoding the category name with a manual prompt, e.g. `a photo of a(n) {class name}` or `a(n) {class name}`, using an appropriate encoder — e.g. a CLIP text encoder, thereby yielding a set of classifiers for $\mathcal{C}^{\text{TEST}}$. This method relies on the text encoder to produce a text-based classifier entirely from its internal understanding of `class name`.

Instead, we make use of natural language descriptions of categories sourced from a large language model (LLM). Such a design choice gives additional details like visual attributes, leading to increased discriminative information in the classifier. This alleviates lexical confusion — `class name` may have two different meanings, and effectively prevents the need for human efforts to manually write descriptions

or spend time searching external sources for them.

Figure 3.2 outlines our method for generating informative text-based classifiers. Specifically, we start by prompting an autoregressive language model with a question: “What does a(n) {class name} look like?”, and sample multiple descriptions per class. We use OpenAI’s API for GPT-3 [Brown et al. 2020] and generate 10 descriptions per class with temperature sampling (Figure 3.2 shows 3 descriptions per class for clarity, more examples can be found in Section 3.F of the Appendix), yielding multiple descriptions of the format {class name} is a ... or similar. Given a set of M plain text descriptions $\{s_i^c\}_{i=1}^M$ for class c , we encode each element of the set with a CLIP text encoder [Radford et al. 2021], $f_{\text{CLIP-T}}(\cdot)$, and the text-based classifier for class c is obtained from the mean of these text encodings:

$$\mathbf{w}_{\text{TEXT}}^c = \frac{1}{M} \sum_{i=1}^M f_{\text{CLIP-T}}(s_i^c) \quad (3.3)$$

At detector training time, text-based classifiers for categories of interest ($c \in \mathcal{C}^{\text{DET}}$ and $c \in \mathcal{C}^{\text{IMG}}$) are pre-computed and kept frozen, the rest of detector parameters are updated i.e. all parameters in Equations 3.1-3.2, except Φ_{CLS} . At inference time, classifiers for testing categories are computed similarly to enable open-vocabulary object detection.

Discussion. In this paper, we only consider a single straightforward question as a prompt to the LLM: “What does a(n) {class name} look like?”. However, it is also feasible to use alternative question prompts, e.g. “How can you identify a(n) {class name}?” or “Describe what a(n) {class name} looks like?”, and obtain visual descriptions with the same or similar concepts [Pratt et al. 2022].

We investigated using a transformer architecture to aggregate text embeddings from natural language descriptions, but we found this was not beneficial to OVOD performance over simply using the mean vector. In general, each text embedding of a natural language description summarises the category of interest very well and so the contrastive task, which is used to train the aggregator (explained in detail for the visual case below), is very easy with text embeddings yielding no improvement in text-based classifiers for OVOD.

3.3.3 Vision-based Classifiers from Image Exemplars

In addition to constructing classifiers with natural language descriptions, another natural option is to use image exemplars, especially in cases where a good description of the category is prohibitively long (such as the painted lady butterfly or *Vanessa cardui* which has an intricate wing pattern), or occasionally the class name is not known beforehand, e.g. “Deerstalker cap” refers to the hat often worn by Sherlock Holmes.

In such scenarios, we propose to construct classifiers by using image exemplars, as shown in Figure 3.3. Specifically, given a set of K RGB image exemplars for category c , $\{\mathbf{x}_i^c\}_{i=1}^K$, we encode each exemplar with a CLIP visual encoder, $f_{\text{CLIP-IM}}(\cdot)$, yielding K image embeddings, which are then passed to a multi-layer Transformer [Vaswani et al. 2017] with learnable [CLS] token, $\mathbf{t}_{\text{CLS}}$:

$$\mathbf{w}_{\text{IMG}}^c = \text{Transformer} \left(\{f_{\text{CLIP-IM}}(\mathbf{x}_i^c)\}_{i=1}^K; \mathbf{t}_{\text{CLS}} \right) \quad (3.4)$$

The Transformer architecture acts to best aggregate the K image exemplars, and the output from the [CLS] token is used as the vision-based classifier for OVOD. When training the transformer aggregator, all exemplars are sourced from ImageNet-21k-P [Ridnik et al. 2021]. When generating vision-based classifiers for OVOD, where possible, we source our exemplars from ImageNet-21k [J. Deng et al. 2009] — where this is not possible, we use LVIS and/or VisualGenome training data to source image exemplars. Additional details on how we collate exemplars for training and testing are provided in the Appendix (Section 3.D). This transformer architecture will be referred to as the *visual aggregator* and its training procedure is described next.

Offline Training. The visual aggregator is trained offline i.e. it is not updated during detector training. The training procedure needs to learn an aggregator which combines multiple image exemplars to produce effective vision-based classifiers for OVOD — a classifier for a given class needs to be discriminative w.r.t. other classes. A CLIP image encoder is used to provide initial embeddings for each exemplar. We keep the CLIP image encoder frozen during training to improve training efficiency and prevent catastrophic forgetting in the CLIP representation. To provide discriminative vision-based classifiers, contrastive learning

is utilised. For a given class, the output embedding from the visual aggregator is trained to minimise similarity with the output embedding from other classes and maximise the similarity with an output embedding from the same class. To do this, the contrastive InfoNCE [van den Oord et al. 2018] loss is used. The visual aggregator should generalise well and not be trained for a specific downstream OVOD vocabulary, therefore it is trained with the ImageNet-21k-P dataset [Ridnik et al. 2021] for image classification, which contains $\sim 11\text{M}$ images across $\sim 11\text{K}$ classes. For category c during visual aggregator training, at each iteration, two distinct sets of K exemplars are sampled, augmented and encoded by the frozen CLIP image encoder. The two sets are input separately to visual aggregator, outputting 2 embeddings from the learnable [CLS] token for class c . Given a batch size B , the InfoNCE contrastive loss ensures sets formed from the same class have similar embeddings and those of different classes are separated. Once trained, the visual aggregator and visual encoder are frozen and provide vision-based classifiers for categories in $\mathcal{C}^{\text{DET}} \cup \mathcal{C}^{\text{IMG}} / \mathcal{C}^{\text{TEST}}$ during detector training/testing. Additional details on how the visual aggregator is trained are provided in the Appendix (Section 3.C).

Discussion. Using image exemplars for open-vocabulary detection may share some similarity to few-shot object detection, however, there is a key distinction. In few-shot object detection, the given “novel/rare” annotations (albeit few) are available for training, e.g. recent works have found that finetuning a pre-trained object detector on few-shot detection data yields the best results [Xin Wang et al. 2020; Qiao et al. 2021; Kaul et al. 2022], while in open-vocabulary detection, there are no bounding box annotations for “novel/rare” categories. Image exemplars (i.e. the wholes image without bounding boxes) are used to specify the categories of interest; we do not update any parameters based on “novel/rare” category bounding box data, unlike in few-shot object detection.

During ablation experiments in Section 3.A, we compare the visual aggregator to a simple mean operator, when obtaining a vision-based classifier from multiple image exemplar embeddings, and show the benefit of the trained aggregator in this case.

3.3.4 Constructing Classifiers via Multi-Modal Fusion

To go one step further, a natural extension to the aforementioned methods is to construct classifiers from multi-modal cues; intuitively, natural language descriptions and image exemplars may contain complementary information. For a given class, c , with text-based classifier, $\mathbf{w}_{\text{TEXT}}^c$, and vision-based classifier, $\mathbf{w}_{\text{IMG}}^c$, the multi-modal classifier, $\mathbf{w}_{\text{MM}}^c$, is computed by a simple fusion method based on addition:

$$\mathbf{w}_{\text{MM}}^c = \frac{\mathbf{w}_{\text{TEXT}}^c}{\|\mathbf{w}_{\text{TEXT}}^c\|_2} + \frac{\mathbf{w}_{\text{IMG}}^c}{\|\mathbf{w}_{\text{IMG}}^c\|_2} \quad (3.5)$$

Figure 3.1 demonstrates our entire pipeline for generating text-based, vision-based and multi-modal classifiers, showing how any of the three classifiers can be used with an open-vocabulary detector to detect a “falcon”.

Discussion. Section 3.3.3 provides details of the visual aggregator, which yields our vision-based classifiers, but for multi-modal classifiers we simply compute the vector sum of our l^2 -normalised text-based and vision-based classifiers. We investigated using a unified multi-modal aggregator which ingests both text and visual embeddings, sourced from class descriptions and image exemplars, respectively. Such a model did not generate good multi-modal classifiers for OVOD — distinguishing between sets of text and image embeddings for different classes becomes trivial as the text embeddings alone are sufficient to solve the contrastive learning task, thereby ignoring the visual embeddings altogether. Attempts to modify the training for a unified multi-modal aggregator by using Dropout [N. Srivastava et al. 2014] on the text embeddings were not fruitful.

3.4 Experiments

In this section, we first introduce the standard dataset and benchmark used in the literature [Gu et al. 2022; X. Zhou et al. 2022; C. Feng et al. 2022]. Section 3.4.2 provides implementation and training details for our OVOD models, which use classifiers constructed from natural language descriptions, visual exemplars or the combination of both. We compare our models with existing works in Section 3.4.3, demonstrating state-of-the-art performance. Additionally, Section 3.4.3 provides

results for cross-dataset transfer. Section 3.4.4 provides an ablation study regarding our design choices.

3.4.1 Datasets and Evaluation Protocol

Standard LVIS Benchmark. In this work, most experiments are based on the LVIS object detection dataset [A. Gupta et al. 2019], containing a large vocabulary and a long-tailed distribution of object instances. Specifically, the LVIS dataset contains class, bounding box and mask annotations for 1203 classes across 100k images in the MS-COCO dataset [T.-Y. Lin et al. 2014]. Annotations are collected in a federated manner, *i.e.* manual annotations for a given image are not necessarily exhaustive. The classes are divided into three sets — rare, common and frequent — based on the number of training images containing a given class.

Training Datasets. To develop open-vocabulary object detectors, we follow the same setting as proposed in ViLD [Gu et al. 2022] and used in Detic [X. Zhou et al. 2022]. Specifically, the original LVIS training set (*LVIS-all*) is slightly changed by removing all annotations belonging to the “rare” categories. This removes the annotations of 317 rare classes *but not* the associated images, in other words, objects belonging to rare categories appear in the training set but are unannotated. This subset of LVIS training data containing only “common” and “frequent” annotations is referred to as *LVIS-base*. LVIS-base serves as $\mathcal{D}^{\text{DET}}$ using notation from Section 3.3.1, unless stated otherwise. When using image classification data, $\mathcal{D}^{\text{IMG}}$, as extra weak supervision, we use the subset of categories in ImageNet-21K [J. Deng et al. 2009] that overlap with the LVIS vocabulary and denote this subset as *IN-L*, as in Detic. IN-L covers 997 of the 1203 classes in the LVIS vocabulary.

Evaluation Protocol. For evaluation, previous work evaluates OVOD models on the LVIS validation set (*LVIS-val*) for all categories — treating “rare” classes as novel categories as it is guaranteed that no groundtruth box annotations whatsoever are provided at the training stage. The main evaluation metric is the standard mask AP metric averaged over the “rare” classes and is denoted as APr. The mask AP averaged across *all* classes is also reported, indicating overall class performance and is denoted as mAP. The latter metric is an important consideration as a good model should improve both APr and mAP; a model should not improve APr at

the cost of worse performance in terms of mAP.

3.4.2 Implementation Details

Object Detector Architecture. The architecture we use is almost identical to that in Detic, using the CenterNet2 [X. Zhou et al. 2021] model with a ResNet-50 backbone [K. He et al. 2016] pre-trained on ImageNet-21k-P [Ridnik et al. 2021]. In addition to exploring different ways for constructing classifiers (Φ_{CLS}), as described in Section 3.3, we also add a learnable bias before generating final confidence scores, the effect of this bias term is investigated in the Appendix (Section 3.A).

Detector Training. The training recipe is the same as Detic for fair comparison, using Federated Loss [X. Zhou et al. 2021] and repeat factor sampling [A. Gupta et al. 2019]. While training our OVOD model on detection data only, $\mathcal{D}^{\text{DET}}$, we use a $4\times$ schedule (~ 58 LVIS-base epochs or 90k iterations with batch size of 64). When using additional image-labelled data (IN-L), we train jointly on $\mathcal{D}^{\text{DET}} \cup \mathcal{D}^{\text{IMG}}$ using a $4\times$ schedule (90k iterations) with a sampling ratio of 1 : 4 and batch sizes of 64 and 256, respectively. This results in ~ 15 IN-L epochs and an additional ~ 11 LVIS-base epochs. For mini-batches containing images from $\mathcal{D}^{\text{DET}}$ and $\mathcal{D}^{\text{IMG}}$ we use input resolutions of 640^2 and 320^2 , respectively. We conduct our experiments on 4 32GB V100 GPUs.

For image-labelled data, $\mathcal{D}^{\text{IMG}}$, an image with class label is given, but no groundtruth bounding box is available. Following Detic, the largest class-agnostic box proposal is used to produce an RoI feature for the given image, enabling detector training. See the Detic paper for more details.

Text-based Classifier Construction. To generate plain text class descriptions we use the GPT-3 DaVinci-002 model available from OpenAI. For each class in LVIS, we generate 10 descriptions and compute the classifier with the text encoder from a CLIP ViT-B/32 model [Radford et al. 2021], as detailed in Section 3.3.2. We follow the standard method from CLIP and use the output embedding corresponding to the final token in the input text.

Vision-based Classifier Construction. The visual aggregator, detailed in Section 3.3.3, should be general and not specific to any class vocabulary. To fulfil this goal, we use the curated ImageNet-21-P dataset [Ridnik et al. 2021] as training

data *for the aggregator*. This dataset, designed for pre-training visual backbones, filters out classes with few examples from the original ImageNet-21k [J. Deng et al. 2009] dataset, leaving ~ 11 M images across ~ 11 K classes.

To generate a visual embedding from a given image exemplar, we use a CLIP ViT-B/32 visual encoder. For the aggregator, we use $N = 4$ transformer blocks, with dimension 512 (the same as the output dimension of the CLIP visual encoder) and a multilayer perceptron dimension of 2048. Comprehensive details of aggregator training is provided in the Appendix (Section 3.C). To test the effectiveness of our visual aggregator, we generate baseline vision-based classifiers by taking the vector mean of the CLIP visual embeddings from the K image exemplars.

When constructing vision-based classifiers for OVOD, we find making use of test-time augmentation (TTA) improves performance. Note, TTA here refers to augmentation of the image exemplars used to build vision-based classifiers *not* test-time augmentation of the test image on which OVOD is performed. In our work, each image exemplar is augmented 5 times and input separately to the visual encoder. Therefore, given K image exemplars, we generate $5K$ visual embeddings to be ingested by our visual aggregator. More details on the use of TTA in constructing vision-based classifiers are found in the Appendix (Section 3.A).

For total clarity regarding when datasets are used — the visual aggregator is *trained* using ImageNet-21k-P [Ridnik et al. 2021] and the vision-based classifiers for OVOD are generated from relevant image exemplars *using the trained aggregator* (see Section 3.D of the Appendix for details on sourcing the image exemplars). Detection data (e.g. *LVIS-base*) is only used to train the open-vocabulary object detector and image-level data (e.g. *IN-L*) may be used as an extra source of weak supervision as in Detic [X. Zhou et al. 2022].

Multi-Modal Classifier Construction. When computing multi-modal classifiers, we simply compute the category-wise l^2 -normalised classifier from each modality, regardless of method used to compute them, and take the vector sum. In all cases the text-based classifiers are sourced from class descriptions as described in Section 3.3.2. We combine our text-based classifiers with our vision-based classifiers, using the trained visual aggregator. However, once again, to test the effectiveness of our visual aggregator, we also combine our text-based classifiers

Model	Backbone	Extra Data	APr	mAP
ViLD [Gu et al. 2022]	ResNet-50	X	16.1	22.5
Detic [X. Zhou et al. 2022]	ResNet-50		16.3	30.0
ViLD-ens [Gu et al. 2022]	ResNet-50		16.6	25.5
OV-DETR [Zang et al. 2022]	ResNet-50 + DETR		17.4	26.6
F-VLM [Kuo et al. 2022]	ResNet-50		18.6	24.2
Ours (Text-Based)	ResNet-50	X	19.3	30.3
Ours (Vision-Based)			18.3	29.2
Ours (Multi-Modal)			19.3	30.6
RegCLIP [Zhong et al. 2022]	ResNet-50	CC3M	17.1	28.2
OWL-ViT [Minderer et al. 2022] [†]	ViT-B/32	LiT	19.7	23.3
Detic [X. Zhou et al. 2022]	ResNet-50	IN-L	24.6	32.4
Ours (Text-Based)	ResNet-50	IN-L	25.8	32.7
Ours (Vision-Based)			23.8	31.3
Ours (Multi-Modal)			27.3	33.1
Fully-Supervised [X. Zhou et al. 2022]	ResNet-50	X	25.5	31.1

Table 3.1: Detection performance on the LVIS Open Vocabulary Detection Benchmark using our three types of classifier compared with previous works. Best and second-best performing models are coloured **blue** and **red**, respectively. We split models into those which only use LVIS-base as training data (top) and those which use additional image-level data (bottom). Furthermore, we show results for a fully-supervised model from Detic trained on LVIS-all in **grey**. [†] OWL-ViT reports bbox AP metrics and was trained on Objects365 [Shao et al. 2019] and VisualGenome [Krishna et al. 2017] *not* LVIS-base, therefore it is possible LVIS-defined “rare” classes are contained in the detection training data of OWL-ViT. Due to limited compute resources we present and compare to models which use ResNet-50 [K. He et al. 2016] backbones or similar. We report mask AP metrics except for [†].

with the baseline vision-based classifiers described in the previous paragraph.

3.4.3 Open-Vocabulary Detection Results

LVIS OVOD Benchmark. Table 3.1 shows results on *LVIS-val* for our work, which uses text-based, vision-based and multi-modal classifiers, compared to a range of prior work. We report overall mask AP performance and mask AP for “rare” classes only. The latter metric is the key measure of OVOD performance. We separate the comparisons into those models which do not use additional image-level data (top half of Table 3.1) and those which do (bottom half of Table 3.1). For a fair evaluation, we compare to models from prior works which use a ResNet-50 [K. He et al. 2016] backbone. There are two exceptions: (1) OWL-ViT [Minderer et al.

2022] which only investigates using Vision Transformers for OVOD [Dosovitskiy et al. 2021] – we compare ResNet-50 models to the ViT-B/32 OWL-ViT model as it requires similar compute during inference in terms of GLOPs (141.5 and 139.6, respectively); (2) OV-DETR [Zang et al. 2022] which uses a DETR-style architecture [Carion et al. 2020] consisting of a ResNet-50 CNN backbone and modified transformer encoder and decoder.

In the experiments without using extra data ($\mathcal{D}^{\text{IMG}} = \emptyset$), our models with text-based or multi-modal classifiers obtain the best performance on both APr and overall mAP, while F-VLM and Detic only performs strongly on APr and mAP, respectively. Our model with text-based classifiers is most directly comparable to Detic and our model outperforms Detic by 3.0 APr. When using extra data, we make use of a ImageNet-21k subset as in Detic (IN-L). Models with text-based and multi-modal classifiers outperform Detic (previous state-of-the-art) by 1.2 and 2.7 APr respectively. **Note that**, our models trained with IN-L even outperform the fully-supervised baseline using CenterNet2, i.e. trained on “rare” class box annotations. To the best of our knowledge, this is the first work on open-vocabulary detection that outperforms a comparable fully-supervised model on the challenging LVIS benchmark. Note, some other works use larger vision backbones, e.g. Swin-B [Ze Liu et al. 2021], but due to limited computation resources we only present and compare to models with ResNet-50 backbones or similar. Our models with text-based and multi-modal classifiers surpass state-of-the-art performance when using a ResNet-50 backbone or similar.

Cross-dataset Transfer. Table 3.2 shows results for cross-dataset transfer from LVIS to Objects365 [Shao et al. 2019] when using our text-based classifiers. We compare our work to equivalent models from Detic, reporting box AP metrics as standard in Objects365. In all cases, these models are trained on *LVIS-all* and the models in the bottom two rows use *IN-L* as extra weak supervision. The trained open-vocabulary detectors are evaluated on the Objects365 validation set. Following Detic, we define “rare” classes in Objects365 as the $\frac{1}{3}$ of classes with the lowest frequency in the Objects365 training set. Note the Detic models we compare to in Table 3.2 are not the same as those listed in Table 4 of the Detic paper, which use a large Swin-B backbone and all of ImageNet-21k as extra data. Instead, we compare to other Detic models publicly available which are trained

Model	Extra Data	Objects365 mAP	Objects365 AP50	Objects365 APr
Detic [X. Zhou et al. 2022]	x	13.9	19.7	9.5
Ours (Text-Based)		14.8	21.0	10.1
Detic [X. Zhou et al. 2022]	IN-L	15.6	22.2	12.4
Ours (Text-Based)		16.6	23.1	13.1

Table 3.2: Detection performance when training on *LVIS-all* (i.e. all LVIS training data) and evaluating on Objects365 [Shao et al. 2019], where the least frequent $\frac{1}{3}$ of classes are defined as “rare”. Best performing models are coloured **blue**. Our text-based classifiers outperform Detic when transferring to Objects365 across all classes and “rare” classes only. Note the Detic [X. Zhou et al. 2022] models we compare to on Objects365 are not the same as the models listed in the Detic paper, which use a large Swin-B backbone and all of ImageNet-21k as extra data for weak supervision. We use a fair comparison Detic model which uses the same training data (see text for more details). We report box AP metrics for Objects365.

on *LVIS-all* (and *IN-L*). For ease, we provide links to the Detic configuration and checkpoint files used in this cross-dataset transfer evaluation ^{1 2}. Evaluation on Objects365 after training on LVIS is easily done with Detic or with our text-based classifiers. In the case of Detic, the simple classifiers based on LVIS class names are replaced with equivalent simple classifiers based on Objects365 class names. For our text-based classifiers, plain text descriptions are generated for each of the Objects365 classes and are encoded as described in Section 3.3.2 and replace the text-based classifiers for LVIS. In both models, Detic and ours, all other parameters of the open-vocabulary detector remain the same. In all cases, using our text-based classifiers gives performance improvements over the equivalent Detic model. Considering all classes, our method with extra data outperforms Detic by 1.0 mAP and 0.9 AP50. For “rare” classes as defined above, our method with extra data outperforms the equivalent Detic model by 0.7 APr. These results demonstrate that our method, which uses text-based classifiers generated from rich class descriptions, provides additional information compared to using a simple classifier based on the class name only, even when the training and testing class vocabularies are disjoint.



Figure 3.4: Some qualitative detection examples using our model with text-based classifiers, detecting “rare” category instances in *LVIS-val*. Our text-based classifiers are sourced from rich natural language descriptions of a given class by prompting an GPT-3 LLM.

3.4.4 Ablation Study

Results with Vision-Based Classifiers. Using vision-based classifiers for OVOD is an under-explored area and so we compare our method, detailed in Section 3.3.3, to baseline classifiers in which the same visual encoder is used, but the action of the aggregator is replaced by performing simple vector mean. The orange rows of Table 3.3 compares the use of our visual aggregator to performing simple vector mean across visual embeddings instead. When no additional image-level data is used (top two orange rows) our aggregator (Model B) boosts performance by 3.5 APr compared to the vector mean baseline (Model A). For models which train on additional image-level data (IN-L) our aggregator (Model G) boosts performance by 2.2 APr compared to the baseline (Model F). This comparison demonstrates the utility of our visual aggregator in constructing better vision-based classifiers rather than naïvely averaging the K visual embeddings. Note our vision-based classifiers and the baseline classifiers both utilise TTA as mentioned in Section 3.4.2 (see Section 3.A of the Appendix for details on TTA). The results in Table 3.3 use $K = 5$. Further results for $K = 1, 2, 10$ are found in the Appendix (Section 3.E).

Results with Multi-Modal Classifiers. To evaluate the effectiveness of multi-modal classifiers we perform similar experiments as those using vision-based classifiers, except for each model we combine the vision-based classifiers with the text-based classifiers, as described in Section 3.3.4. Results for multi-modal classifiers are shown in grey rows of Table 3.3. With no additional image-level data, the vector mean baseline (Model D) outperforms the use of our aggregator (Model E) by 1.4 APr. However, for models with image-level data (IN-L) our aggregator (Model J) boosts performance by 0.8 APr compared to the baseline (Model

¹Detic Configuration Checkpoint Extra Data: ✗

²Detic Configuration Checkpoint Extra Data: IN-L

Model	Visual Mean?	Visual Agg.?	Text Cls.?	Extra Data?	APr	mAP
A	✓			✗	14.8	28.8
B		✓			18.3	29.2
C			✓	✗	19.3	30.3
D	✓		✓	✗	20.7	30.5
E		✓	✓		19.3	30.6
F	✓			IN-L	21.6	31.3
G		✓			23.8	31.3
H			✓	IN-L	25.8	32.7
I	✓		✓	IN-L	26.5	32.8
J		✓	✓		27.3	33.1

Table 3.3: Detection performance on the LVIS OVOD benchmark comparing all three of our methods: (1) orange — vision-based classifiers; (2) blue — text-based classifiers; (3) grey — multi-modal classifiers. Results for models trained only on LVIS-base and LVIS-base+IN-L are shown in the top and bottom halves, respectively. Visual Mean?: *simple vector mean* is used to combine visual embeddings of image exemplars, Visual Agg.?: *our visual aggregator* is used to combine visual embeddings, Text Cls.?: text-based classifiers are used. Models which use text-based and vision-based classifiers represent our models with multi-modal classifiers. We report mask AP metrics.

I). Furthermore, comparing the multi-modal classifiers (grey rows in Table 3.3) with text-based classifiers (blue rows in Table 3.3) demonstrates that in all cases adding information from image exemplars yields improved OVOD performance — our best multi-modal model improves performance over our best text-based model by 1.5 APr confirming that combining the vision and text modalities utilises complementary information between the two.

Relationship between IN-L and LVIS “rare” classes. Section 3.B of the Appendix splits the APr metric into two based on the “rare” LVIS categories contained in IN-L. One may expect improvements in APr performance when training on IN-L to only come from “rare” categories found in IN-L. Our evaluation finds this not to be the case. Detailed results can be found in the Appendix (Section 3.B) which breaks the APr metric into “rare” categories found in IN-L and those not.

Additional Ablation Experiments. Section 3.A of the Appendix presents ablation experiments which demonstrate: (1) applying a learnable bias before cal-

culating the final detection score for a region improves OVOD performance; (2) improvements in OVOD performance using our text-based classifiers is orthogonal to applying this learnable bias; (3) applying TTA on image exemplars yield better vision-based classifiers for OVOD; (4) comparisons between our text-based classifiers and those generated from manual prompts. Please refer to Section 3.A of the Appendix for details and evaluation results for these experiments.

3.5 Conclusion

In this paper, we tackle open-vocabulary object detection by investigating the importance of the method used to generate classifiers for open-vocabulary detection. This work goes beyond the very simple methods used in prior work to generate such classifiers — with the class name only. We present a novel method which combines a large language model (LLM) and a visual-language model (VLM) to produce improved classifiers. Moreover, we investigate using image exemplars to provide classifiers for OVOD and present a method for generating such classifiers using a large classification dataset and a simple transformer based architecture. Finally, we combine our classifiers from the two modalities to produce multi-modal classifiers for OVOD. Our experiments show that our method using natural language only outperforms current state-of-the-art OVOD works, especially in cases where no extra image-level data is used. Furthermore, our multi-modal classifiers set new state-of-the-art performance with a large improvement over prior work.

Acknowledgements

We thank Lauren Bates-Brownsword for help with proofreading. This research was supported by the EPSRC CDT in AIMS EP/L015897/1 and EP/S515541/1, the EPSRC Programme Grant VisualAI EP/T028572/1, and a Royal Society Research Professorship. Weidi Xie would like to acknowledge the National Key R&D Program of China (No. 2022ZD0161400).

3.A Ablation Studies

We now ablate some of the key components using the open-vocabulary LVIS benchmark without any extra image classification data i.e. $\mathcal{D}^{\text{IMG}} = \emptyset$, unless stated otherwise. All metrics have the standard LVIS definition and we report mask AP metrics in all cases. For reference APr, APc and APf represent mean average precision across “rare”, “common” and “frequent” classes, respectively, as defined in LVIS. Moreover, mAP, AP50 and AP75 represent mean average precision across all classes but for all intersection-over-union (IoU) criteria, IoU= 0.5 and IoU= 0.75, respectively.

Bias?	Init. Value	mAP	AP50	AP75	APr	APc	APf
$\times$	N/A	29.7	43.7	31.6	15.6	30.5	35.0
$\checkmark$	-2.0	29.9	43.5	32.0	17.7	30.1	35.0

Table 3.4: The effect of including a learnable bias on detections scores for OVOD. The top row does not use a learnable bias, as in Detic. The bottom row applies a learnable bias prior to computing final detection scores with logistic sigmoid. Applying a learnable bias improves performance on novel/rare categories (APr). We report mask AP metrics.

Effect of Detection Score Bias. Table 3.4 shows the effect of adding a learnable bias to the detection scores before applying a logistic sigmoid to get a final detection score in the range $[0, 1]$ To evaluate the effect of the learnable bias only, our proposed text-based classifiers sourced from rich class descriptions, as described in Section 3.3.2, *are not* used and instead the same simple text-based classifiers used in Detic, of form “a(n) {class name}”, are used in this comparison. We observe adding a learnable bias improves open-vocabulary detection by 2.1 AP on rare categories compared to not using a bias, as done in Detic. Without the use of a bias, class-agnostic proposals are not biased towards being labelled as background. With respect to a given class, a proposal is most likely to be negative, therefore use of a bias makes intuitive sense to reflect this and stabilises early training of the detector. Similar findings were found in RetinaNet [T.-Y. Lin et al. 2017b].

Natural Language Descriptions. Table 3.5 shows the effect of using rich class descriptions, sourced from a LLM, rather than forming text-based classifiers from simple text prompts of the format “a(n) {class name}” as in Detic. To compare fairly to Detic with detection data only (top row), we report a set of results which

Model	mAP	AP50	AP75	APr	APc	APf
Detic	30.2	44.2	32.1	16.4	31.0	35.4
Ours (w/o bias)	30.4	44.4	32.3	18.6	30.8	35.2
Ours (w/ bias)	30.3	44.2	32.2	19.3	30.5	35.0

Table 3.5: The effect of using our text-based classifiers sourced from rich descriptions. In contrast, Detic uses simple classifiers based on class names only (top row). Results for a detector trained using our text-based classifiers but no learnable bias is shown in the middle row. Our proposed model makes use of text-based classifiers sourced from rich descriptions and a learnable bias (bottom row). Our method for text-based classifiers improves performance on novel/rare categories (APr) by a large amount. We report mask AP metrics.

do not make use of the learnable bias on the detection scores as detailed above (middle row). Using our text-based classifiers without a learnable bias improves performance on rare categories by 2.2 APr compared to the public Detic model. Using rich class descriptions, a learnable bias and our method (bottom row) further improves open-vocabulary detection on novel/rare categories by 2.9 and 0.7 APr compared to the public Detic model and our method without a learnable bias, respectively.

Visual Encoder	TTA?	mAP	APr	APc	APf
	✗	29.0	16.3	29.1	34.4
CLIP ViT-B/32	✓, harsh	29.0	17.2	28.7	34.6
	✓, gentle	29.2	18.3	28.7	34.4

Table 3.6: The effect of using test-time augmentation (TTA) when generating classifier embeddings from image exemplars using a CLIP image encoder. Both TTA recipes use two common augmentations — **ColorJitter** and **RandomHorizontalFlip**. For harsh/gentle TTA — min scale of **RandomResizedCrop** = 0.5/0.8. Both harsh and gentle TTA perform better than no TTA in terms of performance on novel/rare categories (APr). We report mask AP metrics.

Test-Time Augmentation on Image Exemplars for Vision-Based Classifiers. Table 3.6 shows the effect of using test-time augmentation (TTA) on image exemplars to produce vision-based classifiers with our trained aggregator. For each image exemplar we generate 5 augmentations. As in the main paper, we use the case of $K = 5$ — for each class in the LVIS vocabulary we have 5 RGB image exemplars. As mentioned in Section 3.4.2, we augment each exemplar 5 times when using TTA. We consider two augmentation variations, with

each containing `ColorJitter` and `RandomHorizontalFlip`. The ‘harsh’ variation uses `RandomResizedCrop(scale=(0.5,1.0))` and the ‘gentle’ variation uses `RandomResizedCrop(scale=(0.8,1.0))`. We find adding ‘gentle’ TTA performs best, improving open-vocabulary detection by 2.0 AP on rare categories compared to no use of TTA. In the main paper, when using vision-based classifiers we utilise ‘gentle’ TTA on the image exemplars.

Prompt	mAP	APr	APc	APf
a/an class name	29.9	17.7	30.1	35.0
a photo of a/an class name	29.5	15.9	30.0	34.9
a photo of a/an class name in the scene	29.4	16.2	29.7	34.8
Our LLM descriptions	30.3	19.3	30.5	35.0

Table 3.7: The effect of using manually crafted prompts against our rich class descriptions sourced from LLMs. All models use a learnable bias on the detection scores and text-based classifiers. Using our class descriptions improves performance on “rare” classes compared to manually crafted prompts.

Comparing our LLM Descriptions to Manually Designed Text Prompts.

Table 3.7 compares the detector performance between using simple manually crafted prompts (first three rows) and our rich class descriptions sourced from an LLM (final row), for constructing text-based classifiers. We report results for the case where the detector is trained on *LVIS-base* only (i.e. no additional image-level data is used). In all cases we apply the learnable bias on the detection score. We note that of the manual prompts, the simplest one, of the form “a(n) {**class name**}”, performs best across all metrics. Using our text-based classifiers generated from LLM descriptions improves performance on rare classes by 1.6 APr and by 0.4 mAP across all classes. Performance on “common” and “frequent” classes is largely similar as the availability of labelled detection data for these classes renders the quality of the classifier less important.

3.B A Closer Look at “rare” Class Performance

Model	$\mathcal{D}^{\text{IMG}}$	mAP	APr	APr-w	APr-z
Detic [X. Zhou et al. 2022]	$\times$	30.2	16.3	15.7	19.7
Ours (Text-Based)		30.3	19.3	19.2	19.4
Ours (Multi-Modal)		30.6	19.2	18.5	22.2
Detic [X. Zhou et al. 2022]	$\checkmark$	32.4	24.9	25.4	23.0
Ours (Text-Based)		32.6	25.8	26.7	21.7
Ours (Multi-Modal)		33.1	27.3	27.8	24.9

Table 3.8: Comparison between Detic and our models using text-based classifiers and multi-modal classifiers on LVIS. As IN-L does not cover all classes in the LVIS vocabulary, we split the rare class metric APr into APr-w and APr-z which represent rare classes with and without weak annotations from IN-L, respectively. Best performing models are shown in **bold**. We report mask AP metrics.

Section 3.4.1 detailed the data used to train our detector. IN-L contains images from the 997 classes from LVIS present in ImageNet-21k [J. Deng et al. 2009].

IN-L gives weak supervision during detector training. Out of the 337 “rare” classes in LVIS, 277 are covered by IN-L and are therefore weakly supervised, leaving 60 classes for which no weak supervision is available.

To investigate the improvement in performance when using IN-L, we split the rare class metric (APr), which reports average precision across rare classes, into APr-w which averages across rare classes *present* in IN-L and APr-z which averages across rare classes *not present* in IN-L which are therefore truly zero-shot classes. Note that when no extra image-level data is used, i.e. $\mathcal{D}^{\text{IMG}} = \emptyset$, all rare classes are truly zero-shot classes.

Table 3.8 shows the result of using this breakdown. These results show training on IN-L improves performance on rare classes not contained in IN-L, which may not be expected. The weak supervision from IN-L leads to a reduction in false positives for all rare classes leading to improved performance across all metrics. Moreover, our multi-modal classifiers perform best across all metrics.

3.C Vision-based Classifier Pipeline Implementation Details

For the transformer architecture of the visual aggregator detailed in Section 3.3.3, we use 4 transformer encoder blocks, each with a hidden dimension of 512 and MLP dimension 2048. As input to the first transformer block, we encode each image exemplar with a CLIP image encoder which remains frozen throughout training, yielding one embedding per exemplar. The set of embeddings are input with a learnable [CLS] token. The output [CLS] token is used as the final vision-based classifier.

To train the model we use the ImageNet-21k-P dataset [Ridnik et al. 2021] for 10 epochs. To speed up and improve training we store a dynamic queue of size $4096 \times K$ CLIP encoded embeddings, with 512 positions in the queue updated each iteration, using a last-in first-out policy. Each set of K represents encodings from K randomly sampled images for a single class. We use $K = 5$. For contrastive training, we use a temperature of 0.02 in the InfoNCE [van den Oord et al. 2018] loss function, the AdamW [Loshchilov and Hutter 2018] optimiser with standard hyperparameters and a learning rate of 0.0002. Furthermore, during training we uniformly sample $k \in [1 : K]$ to simulate varying numbers of image exemplars being available for downstream OVOD when $\mathcal{C}^{\text{TEST}}$ is defined. Therefore, for a given iteration, there may 1 – 5, visual embeddings input per class. Prior to input to the CLIP image ViT encoder, we apply random augmentations to each sampled image from ImageNet-21k-P. We use an augmentation policy similar to SimCLR [Ting Chen et al. 2020], which includes `RandomResizedCrop`, `ColorJitter` and `RandomGrayscale`. We discover that test-time augmentation of the image exemplars available for the vocabulary in $\mathcal{C}^{\text{TEST}}$ improves downstream OVOD performance. For each image exemplar, we generate 5 test-time augmentations. Therefore if we have L image exemplars for a given class in $\mathcal{C}^{\text{TEST}}$, $5L$ augmented images are encoded using the CLIP image encoder and fused using the learnt transformer architecture — the visual aggregator — as described in Section 3.3.3. Use of test-time augmentation is ablated in Section 3.A.

3.D Sourcing Image Exemplars

In this section, we detail how image exemplars are sourced when performing experiments using vision-based classifiers and multi-modal classifiers, as described in Section 3.3.3 and 3.3.4, respectively. We start with an empty image exemplar dictionary (IED) for the 1203 LVIS classes.

From constructing the image-level dataset IN-L, we know that ImageNet-21k [J. Deng et al. 2009] contains 997 out of the 1203 classes in the LVIS vocabulary, using exact WordNet [Miller 1995] synset matching. We add IN-L to our IED. The result is 988 classes have more than 40 images in ImageNet-21k. This leaves 215 classes for which there are too few or no image exemplars (< 40).

Next, to try and fill this gap, we turn to LVIS itself. We add the LVIS training annotations with area greater than 32^2 to our IED. There are now 1095 LVIS classes with more than 40 examples, leaving 48 classes with at least 10 exemplars and 60 classes with less than 10 exemplars.

The final dataset we turn to is VisualGenome [Krishna et al. 2017], which provides bounding box annotations for 7605 WordNet synsets. We include the annotations from VisualGenome with an exact WordNet synset match with the LVIS vocabulary to our IED. We now have 1110 LVIS classes with at least 40 exemplars and 1160 with at least 10 exemplars. Reducing our minimum required number of exemplars per class from 40 to 10 leaves 43 classes with too few exemplars.

At this point, we inspect each of the remaining 43 classes by hand and find that all have other synsets present in ImageNet-21k which are visually identical or very similar. For example, “anklet” is a “common” class in LVIS, for which LVIS gives a definition of “an ornament worn around the ankle” and a WordNet synset of `anklet.n.03`. This synset is not found in ImageNet-21k but `anklet.n.02`, defined as “a sock that reaches just above the ankle” by WordNet, is present and visual inspection shows these images to actually exactly match `anklet.n.03`. Therefore, we add ImageNet-21k images relating to `anklet.n.02` to our IED. As another example, `penny.n.02` (as in the penny coin) is a “rare” class in LVIS for which exemplars could not be found automatically. However, ImageNet-21k contains images of `coin.n.01` which is a hypernym `penny.n.02`. The images for `coin.n.01` are visually extremely similar and often identical to those one would

expect for `penny.n.02` and so we add ImageNet-21k images relating to `coin.n.02` to our IED.

After applying some human effort as described above, our IED contains at least 40 image exemplars for 1110 (92% of LVIS classes) and at least 10 image exemplars for all LVIS classes.

3.E Varying Number of Image Exemplars for Vision-based Classifiers

In this section we show results using vision-based classifiers varying the number of K image exemplars used per class. Figure 3.5 shows performance on the LVIS OVOD benchmark for rare classes using $K = 1, 2, 5, 10$, where K is the number of image exemplars per class used. We compare our method which makes use of our aggregator (red dashed), which has a transformer architecture, with the simple vector mean of the embeddings (blue solid) for the K image exemplars. In both cases we apply the ‘gentle’ TTA detailed and ablated in Section 3.A.

These results use LVIS-base as detection training data, no additional image-level labelled data i.e. $\mathcal{D}^{\text{IMG}} = \emptyset$ and CLIP ViT-B/32 as the pre-trained visual encoder to produce initial embeddings from each exemplar.

Figure 3.5 shows that for each value of K , the use of our aggregator boosts performance on rare classes demonstrating the utility of our aggregator at combining the most useful information from the K given exemplars. Our method for $K = 5$ surpasses the performance of $K = 10$ with simple vector averaging. For $K = 1$, our method improves performance by 2.3 APr which further demonstrates the improved feature representation — $K = 1$ involves no aggregation as only 1 exemplar is available per class.

Furthermore, we compare to the performance of our text-based classifiers which make use of rich class descriptions sourced from a GPT-3 model. Our vision-based classifiers cannot surpass the performance of our text-based classifiers demonstrating the need for more research into using image exemplars for OVOD.

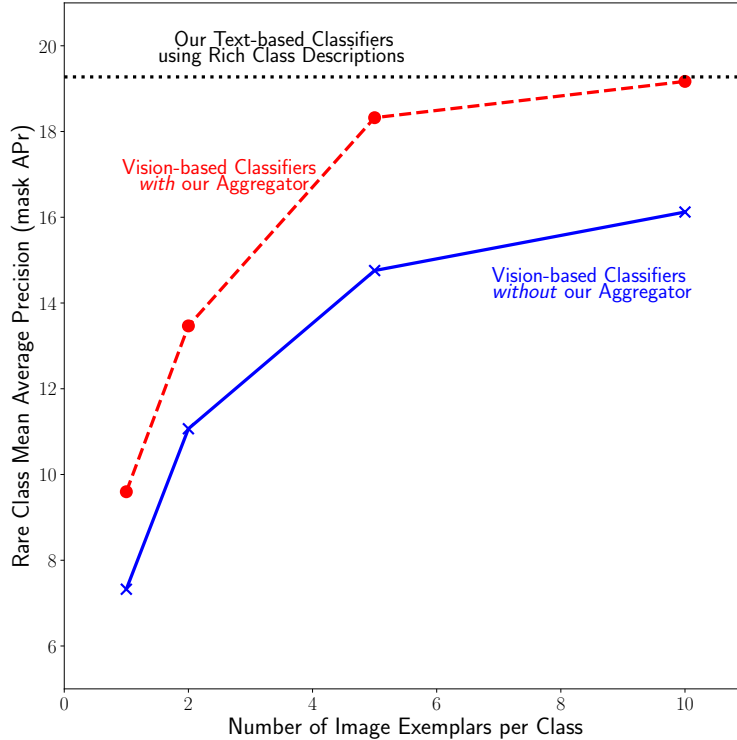


Figure 3.5: Detection performance of our vision-based classifiers on the LVIS OVOD benchmark. We vary the number of image exemplars available per class, K , to investigate the effect of the number of image exemplars on OVOD performance.

3.F Additional Example Class Descriptions

In this section we show a selection of rich class descriptions sourced from the `text-davinci-002` text completion model from OpenAI. For each class in the LVIS vocabulary we generate 10 rich descriptions. We also give the LVIS frequency category — rare, common, frequent.

3.F.1 Generated descriptions for “bagpipe” (rare)

1. A bagpipe is a wind instrument with a bag that is filled with air.
2. A bagpipe typically consists of a blowstick, a chanter, and one or more drones.
3. A bagpipe is a musical instrument that has a bag, a blowpipe, and usually two drones.
4. A bagpipe is a wind instrument with a bag that collects air, a reed pipe for each note, and a blowpipe.
5. A bagpipe is a musical instrument that is played by blowing into a bag of air.
6. Bagpipes vary in appearance, but most have a bag made from a animal skin, a blowpipe, a chanter, and one or more drones.
7. A bagpipe is a musical instrument that has a bag, a blowpipe, and usually drone pipes.

8. A bagpipe is a musical instrument that is usually made out of wood.
9. A typical Highland bagpipe has a chanter with a double reed, a blowstick, three drones with single reeds, and a bag.
10. A bagpipe consists of a blowing bag, a chanter, a drone, and usually one or more drones.

3.F.2 Generated descriptions for “trench coat” (rare)

1. A trench coat is a coat made of heavy cloth, sometimes waterproof, that hangs to about knee length.
2. A trench coat looks like a long, military-style coat.
3. A trench coat typically looks like a long, belted raincoat.
4. A trench coat is a long, water-resistant coat that is typically worn over top of a suit.
5. A trench coat typically has a removable liner, a double-breasted front, and belted cuffs.
6. A trench coat generally refers to a type of coat that is longer than waist length.
7. A trench coat is a coat that is usually a little bit longer than waist length, has a tie or a belt around the waist, and has a collar.
8. A trench coat is a coat made of waterproof material, typically hip-length or longer, with a belt and a collar.
9. A trench coat is a long, light coat with a belt.
10. A trench coat is a raincoat made of heavy-duty fabric, typically poplin, gabardine, or drill.

3.F.3 Generated descriptions for “walrus” (rare)

1. A walrus is a large, flippered marine mammal with a bulky body, short limbs, and a large head with two long tusks protruding from the mouth.
2. A walrus is a blubbery mammal with long tusks, whiskers, and a seal-like face.
3. A walrus is a large, flippered marine mammal with a long, tusked head.
4. A walrus is a stocky, rounded pinniped with small flippers, short fur, and long tusks.
5. A walrus is a large, flippered marine mammal with a bulky body, short tail, and wide, flat head.
6. A walrus is a large ocean mammal with two long tusks, a thick fur coat, and large flippers.
7. A walrus is a large flippered marine mammal with a discontinuous distribution about the North Pole in the Arctic Ocean and sub-Arctic seas of the Northern Hemisphere.
8. A walrus is a large flippered marine mammal with a thick fur coat.

9. A walrus is a large marine mammal with a body shaped somewhat like a seal.
10. A walrus is a seal with a long face and large tusks.

3.F.4 Generated descriptions for “briefcase” (common)

1. A briefcase is a rectangular, portable case used to hold papers, documents, or other materials.
2. A briefcase is a small case used to carry documents and other small items.
3. A briefcase is a small, rectangular-shaped case that is used to carry important documents or other items.
4. A briefcase is typically a rectangle shaped bag made of leather or synthetic materials.
5. A briefcase generally looks like a small, rectangular case made out of a variety of materials, such as leather, canvas, or nylon.
6. A briefcase is a narrow rectangular case used to carry documents and other valuables.
7. A briefcase is a box-shaped bag typically used by businesspeople to transport important documents.
8. A briefcase is a rectangular leather case with a handle.
9. A briefcase is a small case used to carry documents and other small items.
10. A typical briefcase is rectangular and has a handle on the top.

3.F.5 Generated descriptions for “coin” (common)

1. A coin is a small, flat, round piece of metal or plastic that is used as money.
2. A coin has a head side and a tail side.
3. A coin is usually a small, flat, round piece of metal or plastic that is used as money.
4. A coin has a round shape and is flat.
5. A coin generally has a circular shape with a raised edge, and two faces -- one on each side.
6. A coin is a small, flat, round piece of metal or plastic that is used as money.
7. A coin is a round piece of metal with an image on one side and the words "United States of America" on the other.
8. A coin is a small, round, flat piece of metal or plastic that is used as money.
9. Sure, a coin is a small, round, flat piece of metal or plastic that is used as money.
10. A coin is a small, round, metal disk with an image on one side and raised lettering on the other.

3.F.6 Generated descriptions for “waffle” (common)

1. A waffle is a pancake-like pastry that is cooked in a waffle iron and has a distinctively grid-like pattern on the top and bottom.
2. A waffle is a dish made from leavened batter or dough that is cooked between

two plates that are patterned to give a characteristic size, shape, and surface impression.

3. A waffle is a thin, round, batter-based cake that is cooked in a waffle iron and is usually served with syrup.
4. A waffle looks like a round, honeycomb-patterned cake that is cooked in a waffle iron.
5. A waffle is typically a leavened batter or dough that is cooked between two plates that are patterned to give it a characteristic size, shape, and surface impression.
6. A waffle looks like an egg-shaped pancake with deep indentations.
7. A waffle looks like a grid of squares.
8. A waffle is a pancake-like pastry that is made from a leavened batter or dough and is cooked between two heated plates.
9. A waffle is a pancake-like food that is cooked in a waffle iron.
10. A waffle is a hotcake with different patterns on it.

3.F.7 Generated descriptions for “avocado” (frequent)

1. An avocado looks like a pear-shaped fruit with green or blackish skin.
2. It is a green fruit that has a dark brown or black seed in the center.
3. An avocado is a pear-shaped green fruit with smooth, green skin and a large seed in the center.
4. An avocado is a fruit that is brown and bumpy on the outside and green and creamy on the inside.
5. An avocado is a fruit with a dark green or blackish skin and a soft, fleshy inside.
6. An avocado is a green, pear-shaped fruit with a smooth, fleshy texture.
7. An avocado is a pear-shaped fruit with smooth, green skin.
8. An avocado is shaped like an egg and has a greenish-brownish skin.
9. An avocado is typically a dark green or black color on the outside with a soft, light green or yellow color on the inside.
10. An avocado is a pear-shaped fruit with smooth, green skin and a large, pit in the center.

3.F.8 Generated descriptions for “sausage” (frequent)

1. A sausage is a cylindrical shape made of ground meat, typically pork, and a variety of spices and other ingredients.
2. A sausage is typically a ground-up mix of pork, beef, or other meats with spices and salt, encased in a thin skin.
3. A sausage is typically a cylindrical shaped food made from ground meat, spices, and other ingredients.
4. A sausage is a long, thin, cylindrical piece of meat.
5. Sausages are typically long, cylindrical shaped foods made from ground meat and spices.
6. A sausage is a tubular meat product typically made from ground pork, beef, or poultry.
7. A sausage is a cylindrical casing filled with meat, typically pork, and various herbs and spices.
8. A sausage looks like a tubular shape made of ground up meat that is usually encased in a thin layer of intestine.
9. A sausage is a cylindrical casing of meat that is typically filled with ground pork, although many other variations exist.
10. When cooked, a sausage is typically cylindrical and can vary in length.

3.F.9 Generated descriptions for “spectacles” (frequent)

1. A spectacle is a pair of eyeglasses.
2. A spectacles is a type of eyewear that consists of a frame that holds two lenses in front of the eyes.
3. Spectacles are a type of eyewear that helps people see more clearly.
4. Spectacles are glasses that are worn in order to improve vision.
5. A spectacles usually refers to a glass or plastic lens worn in front of the eye to correct vision, or protect the eye from debris, dust, wind, etc.
6. A spectacles is a type of corrective lens used to improve vision.
7. A spectacle is a lens worn in front of the eye to correct vision, for cosmetic reasons, or to protect the eye.
8. A spectacles has a frame that goes around your head and two lenses in front of your eyes.
9. A spectacles has two glass or plastic lenses in metal or plastic frames that rest on the ears.
10. A pair of spectacles is a frame that holds two eyeglasses lenses in front of a person's eyes.

3.G Example Image Exemplars

In this section we show a selection of image exemplars, for LVIS classes, found using the process described in Section 3.D. We also give the LVIS frequency category — rare, common, frequent. For cases where the image exemplar comes from a dataset with bounding boxes (LVIS or VisualGenome) we show the bounding box in yellow.

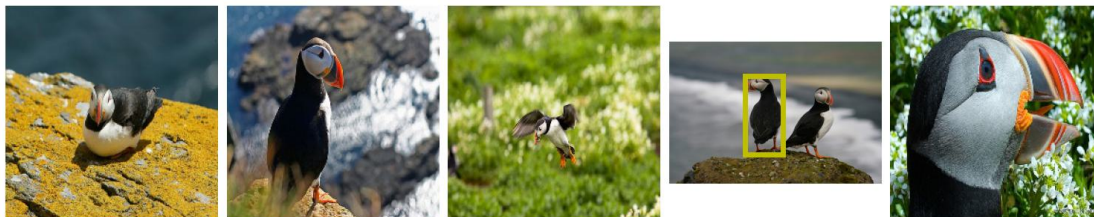


Figure 3.6: Image Exemplars for “puffin” (rare).



Figure 3.7: Image Exemplars for “apricot” (rare).



Figure 3.8: Image Exemplars for “flamingo” (common).



Figure 3.9: Image Exemplars for “lantern” (common).



Figure 3.10: Image Exemplars for “aerosol can” (common).



Figure 3.11: Image Exemplars for “wineglass” (frequent).



Figure 3.12: Image Exemplars for “beanie” (frequent).



Figure 3.13: Image Exemplars for “fire engine” (frequent).

3.H More Qualitative Results

In this section we show more rare category detections on the LVIS OVOD benchmark using our multi-modal classifier trained with IN-L.

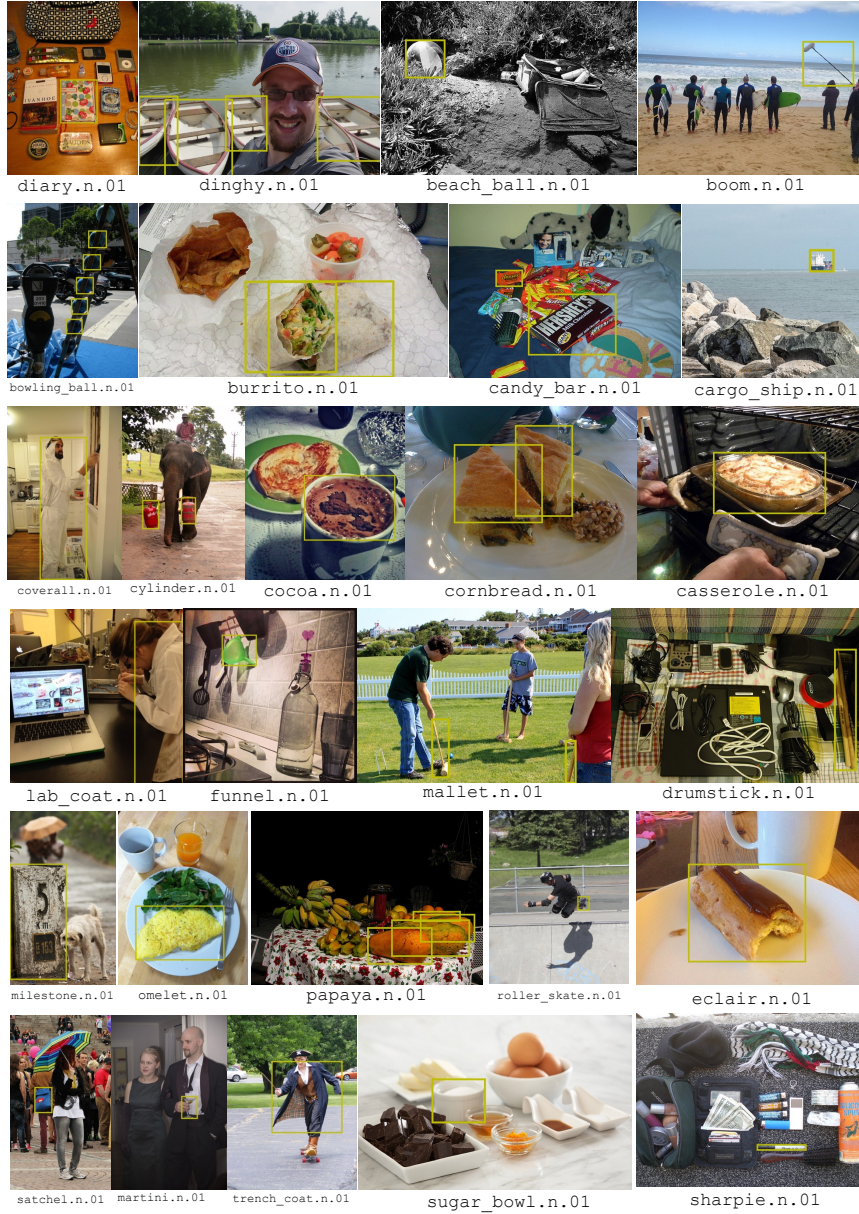


Figure 3.14: Additional qualitative results on LVIS OVOD benchmark.

3.I Dugong



Figure 3.15: An example of a dugong — a visually distinctive marine species with a less well known name.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor (**only required where there isn't already a statement of contribution within the paper itself**).

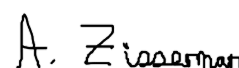
Title of Paper	Multi-Modal Classifiers for Open-Vocabulary Object Detection
Publication Status	☒ Published ☐ Accepted for Publication ☐ Submitted for Publication ☐ Unpublished and unsubmitted work written in a manuscript style
Publication Details	Kaul, P., Xie, W., & Zisserman, A. (2023). Multi-modal classifiers for open-vocabulary object detection. In <i>Proceedings of the 40th International Conference on Machine Learning</i> (pp. 15946–15969)

Student Confirmation

Student Name:	Prannay Kaul		
Contribution to the Paper	First Author Contribution: • conception of research ideas • design and implementation of models • running of large-scale experiments • writing and presentation of the paper		
Signature 	Date	12th January 2024	

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Andrew Zisserman		
Supervisor comments The descriptions of the contributions is accurate.		
Signature 	Date	12th January 2024

This completed form should be included in the thesis, at the end of the relevant chapter.

Chapter 4

THRONE: A Hallucination Benchmark for the Free-form Generations of Large Vision-Language Models

The previous paper utilised multi-modal information to improve the performance of open-vocabulary object detection. In contrast, this paper considers the hallucination of objects in multi-modal models—specifically large vision-language models. Objects hallucinations in text generations from large vision-language models are similar to false positives in object detection.

The paper has been *accepted* for publication at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2024. At the time of thesis submission, the paper *was* under review at the same conference (reflected in the statement of authorship).

THRONE: A Hallucination Benchmark for the Free-form Generations of Large Vision-Language Models

Prannay Kaul^{2,*} Zhizhong Li^{1,†} Hao Yang¹
Yonatan Dukler¹ Ashwin Swaminathan¹ C. J. Taylor¹
Stefano Soatto¹

¹AWS AI Labs ²VGG, University of Oxford

lzhizhon@amazon.com prannay@robots.ox.ac.uk

Abstract

Mitigating hallucinations in large vision-language models (LVLMs) remains an open problem. Recent benchmarks do not address hallucinations in open-ended free-form responses, which we term “Type I hallucinations”. They focus on, if at all, hallucinations responding to very specific questions—yes-no or multiple-choice questions regarding a particular object or attribute—which we term “Type II hallucinations”, and they often require closed-source models which are subject to arbitrary change. Additionally, we observe a reduction in Type II hallucinations does not lead to a congruent reduction in Type I hallucinations; rather, it often increases. We propose THRONE, a novel automatic framework for quantitatively evaluating Type I hallucinations in LVLM free-form outputs. We use public language models (LMs) to identify hallucinations in LVLM responses and compute informative metrics. We evaluate a large selection of recent LVLMs using public datasets. Our results show advances on existing metrics are disconnected from the reduction of Type I hallucinations, and established benchmarks for measuring Type I hallucination prevalence are incomplete. Finally, we provide a simple and effective data augmentation method to reduce Type I and Type II hallucinations as a strong baseline.

4.1 Introduction

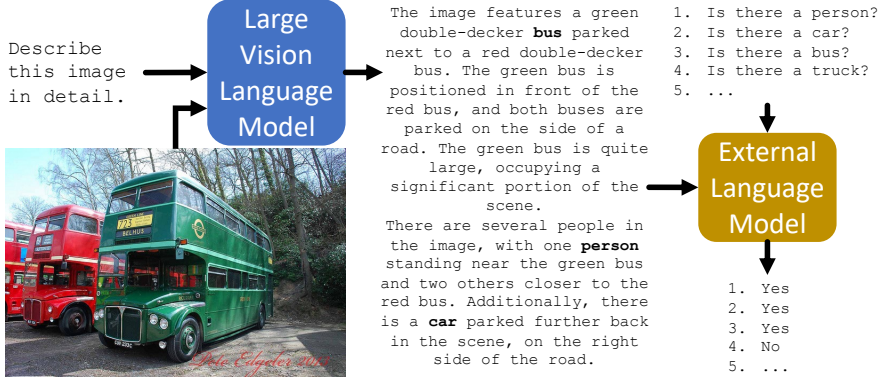


Figure 4.1: THRON (Ours): LVLMs are prompted with a concept neutral instruction. An external LM performs abstractive QA on the response to establish the existence of Type I hallucinations.

This paper proposes a benchmark to evaluate hallucinations by large vision-language models (LVLMs) when generating free-form responses, specifically detailed descriptions, based on a given image.

The rapid advancement in large language models (LLMs) [W. X. Zhao et al. 2023] has pushed the growth in large vision-language models (LVLMs) [Tsimpoukelli et al. 2021; Alayrac et al. 2022; Junnan Li et al. 2022; Junnan Li et al. 2023; W. Dai et al. 2023; S. Huang et al. 2023; Z. Peng et al. 2023; H. Liu et al. 2023b; Q. Ye et al. 2023; D. Zhu et al. 2023; J. Chen et al. 2023] which can produce text responses like LLMs, but condition on input text *and images*, enabling multi-modal perception and comprehension.

LVLMs are largely built on LLMs and therefore inherit their advantages, but also their disadvantages. LLMs have been shown to produce hallucinations [H. Ye et al. 2023; Yue Zhang et al. 2023]—generations which appear plausible but are factually wrong—and current LVLMs also demonstrate this critical issue—models generate text contradicting evidence in the visual input [Y. Zhou et al. 2023]. Hallucinations prevent the use of LVLMs in safety-critical situations such as medical image analysis and therefore evaluating and mitigating hallucinations in LVLMs is crucial for their deployment in such settings. Determining the presence

* Work done during internship at Amazon. †Corresponding author.

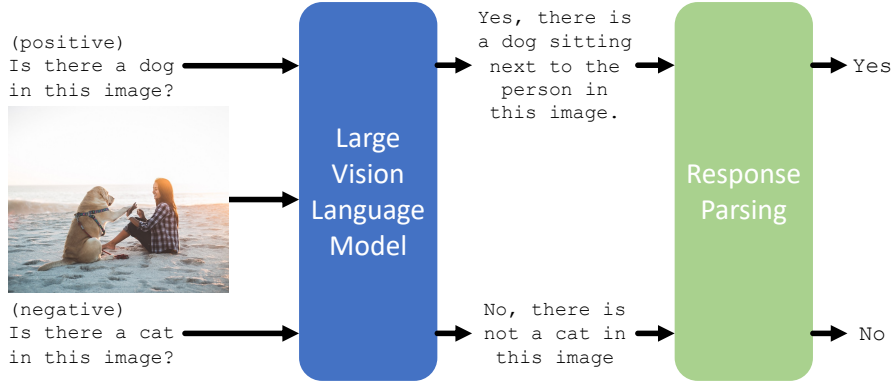


Figure 4.2: POPE: Questions with specific concepts prompt an LVLM directly to evaluate Type II hallucinations [Yifan Li et al. 2023]. Hand-crafted rules parse LVLM responses to give yes/no labels.

and cause of hallucinations in LVLMs remains an open question [Junyang Wang et al. 2023; Y. Zhou et al. 2023].

We divide LVLM hallucinations into two types, namely: Type I, occurring when an LVLM answers an open question and *is not* guided by a user’s text instructions to consider a particular concept—e.g. **What is happening in this image?**; and Type II, occurring when an LVLM *is* guided towards a specific concept by text instructions and invites a response from a fixed set of options like yes-no—e.g. **Is there a traffic light in this image?** Fig. 4.3 illustrates the difference between these two types of hallucination. Reducing hallucinations in both cases is required for useful, multi-purpose LVLMs. However, later in Fig. 4.4 we observe that the same LVLM can give contradicting answers when being evaluated for Type I vs. Type II hallucinations. This implies that measuring and reducing one is not a substitute for the other.

Existing works to evaluate LVLMs largely ignore direct quantification of hallucinations and instead develop comprehensive benchmarks to judge various other desirable abilities such as: optical character recognition, fine-grained recognition and attribute detection [Bohao Li et al. 2023; C. Fu et al. 2023; Y. Liu et al. 2023]. The extent of hallucinations in these benchmarks is not delineated and requires human effort to inspect individual predictions. There are two major established works which specifically evaluate hallucinations in vision-language models: POPE [Yifan Li et al. 2023] and CHAIR [Rohrbach et al. 2018], which we discuss in detail in Section 4.2. However, we observe they both have shortcomings in effectively benchmarking hallucinations:

POPE [Yifan Li et al. 2023] is a recent work addressing Type II hallucinations with respect to object classes. However, we find Type I and Type II hallucinations are disconnected—POPE gives an incomplete picture on LVLM hallucinations. Moreover, POPE systematically undersamples negative object categories leading to a large underestimation of Type II hallucinations (see Section 4.5.4).

CHAIR [Rohrbach et al. 2018] does address Type I hallucinations—establishing object category hallucination in short image captions using simple text matching. However, CHAIR is not suited to current LVLMs because the simple text matching it employs cannot comprehend abstract or hypothetical concepts present in today’s LVLMs (see Fig. 4.4). Further, manual rules for each set of classes are required for usable text matching, and trivial model generated outputs can attain a perfect CHAIR score.

To address the issues of current LVLM hallucination benchmarks, we propose THRONES (*Text-from-image Hallucination Recognition for Open-ended Evaluation*), which leverages language models (LMs) to evaluate Type I hallucinations in free-form, open-ended image descriptions with respect to an object vocabulary of interest. By utilizing LMs, THRONES is able to accurately judge whether an object mentioned in an LVLM response is implied to exist in the image or is abstractly alluded to (see Fig. 4.4).

Moreover, in THRONES, we provide access for all, by leveraging open-source LMs that can run on common GPUs, instead of relying on paid closed-source commercial models [OpenAI 2023a] that are subject to arbitrary change, as done in other works [Bohao Li et al. 2023; Y. Liu et al. 2023; Bitton et al. 2023]. Through a combination of multiple open-source LMs, we ensure THRONES is robust to any biases of a particular LM in judging Type I hallucinations.

We make four contributions: *first*, we establish an accurate and accessible benchmark to quantitatively evaluate object hallucinations in free-form responses, leveraging LMs to judge the existence of Type I hallucinations—quantitatively showing half the judgement errors of CHAIR; *second*, we evaluate a number of current LVLMs on THRONES and demonstrate observed progress in reducing Type II hallucinations do not correspond to a congruent reduction in Type I hallucinations; *third*, we show a recent method, POPE, is incomplete in Type II hallucination

discovery due to its sampling strategy, and mitigate this issue in THRONE while presenting results for a complete version of POPE; and *fourth*, we provide a simple augmentation of visual instruction tuning data which significantly improves performance on Type I hallucinations, while maintaining or improving Type II hallucination performance.

4.2 Related Work

Hallucination Benchmarks for LVLMS. In response to the development of LVLMS, few evaluation benchmarks focusing on hallucinations have been introduced. CHAIR [Rohrbach et al. 2018] is one of the first works to assess hallucinations, designed for short *image captions*. CHAIR uses a fixed set of object classes, extended with their synonyms, as text strings to find predicted object classes in image captions via exact text matching. Subsequently, each class matched in the caption is compared to the COCO [T.-Y. Lin et al. 2014] captions and bounding box annotations to establish object hallucinations. CHAIR was developed prior to current instruction-tuned LVLMS which produce long free-form responses ($10\times$ longer than image captions) with a diverse vocabulary, limiting its applicability to modern LVLMS. As shown in Fig. 4.3, exact text matching in CHAIR cannot understand abstract concepts alluded to in a free-form response, and synonyms have to be manually cross-checked to prevent confusion during evaluation e.g. CHAIR checks all responses for the phrase “toilet seat”, because the exact text matching classifies “seat” as the COCO class “chair”. Finally, CHAIR metrics compare the overall number of predicted objects to the overall number of predicted objects judged to be hallucinations—ignoring the recall of ground-truth objects and the distribution of object classes. This means a single correct prediction across the entire evaluation dataset along with a generic response otherwise e.g. **A natural scene**, achieves a perfect score ($\frac{0 \text{ hallucinated objects}}{1 \text{ predicted objects}} = 0.0$). See the Supplementary Material for a full overview of CHAIR. Our method, THRONE, uses open-source LMs, capable of comprehending abstract concepts, to automatically judge hallucinations in free-form responses. In addition, our method considers both recall and precision to yield a holistic benchmark and does not require any manual curation. THRONE and CHAIR both evaluate Type I hallucinations—hallucinations in response to concept-neutral prompts e.g. **Describe this image in detail**.

POPE [Yifan Li et al. 2023] is a recently proposed benchmark to evaluate object hallucinations in LVLMs—specifically Type II hallucinations, in which an LVLm is directly queried with a yes-no question regarding the existence of a particular object of the form: `Is there {a/an} {object_class_name} in the image?`. The LVLm response is parsed using simple rules to determine whether a Type II hallucination has occurred. Precision and recall metrics are compiled using the parsed LVLm responses and the ground-truth annotation data. Despite focusing on measuring hallucinations, POPE only queries an LVLm with 3 positive and 3 sampled negative questions per COCO image i.e. the evaluation is artificially balanced. This means many potential hallucinations with respect to the COCO categories are not captured by their method. In Section 4.5.4, we show POPE dramatically underestimates Type II hallucinations and present the results of a complete version. Our method, THRONE, evaluates the prevalence of Type I hallucinations, which we observe are disjoint to Type II hallucinations.

Comprehensive Benchmarks for LVLms have recently grown in number. MMBench [Y. Liu et al. 2023] and MM-Vet [Yu et al. 2023] assess various aspects of LVLm performance such as: color perception, celebrity recognition, and numerical calculation. However, these works use paid closed-sourced API models for evaluation. We believe this reduces access to the wider community and lacks reproducibility as closed-sourced API calls are stochastic and models are often updated or discontinued. Exceptions are MME [C. Fu et al. 2023] and SEED-Bench [Bohao Li et al. 2023], but the impact of Type II hallucinations on final metrics is conflated with any number of other aspects of model performance. Our method, THRONE, directly addresses Type I hallucinations, only making use of open-source language models and datasets.

Large Vision Language Models (LVLms) have rapidly developed on the back of advancements in large language models (LLM) [Touvron et al. 2023a; Raffel et al. 2020; Brown et al. 2020]. In contrast to earlier vision-language models such as CLIP [Radford et al. 2021; Jia et al. 2021], LVLms are generally comprised of a pretrained LLM and image encoder, aligned with a connector module of varying complexity. Some works are highlighted here. Frozen [Tsimpoukelli et al. 2021] is an early work fine-tuning the vision encoder to dynamically prefix the prompt to a frozen LLM. Flamingo [Alayrac et al. 2022] combines visual and language features

Image	Prompt	Response
	What is happening in this image?	In the image, a man is hanging clothes on a clothesline that is attached to the back of a moving car.
	Is there a traffic light in this image?	Yes, there is a traffic light in the image, and the man is standing on the back of a car near it.

Figure 4.3: **Type I vs. Type II Hallucinations:** (Top) LVLMS prompted with concept-neutral instructions produce Type I hallucinations. (Bottom) Instructions specifying a concept produce Type II hallucinations. Examples from LLaVA-v1.5 [H. Liu et al. 2023a].

using cross-attention layers in an otherwise frozen LLM. BLIP-2 [Junnan Li et al. 2023] uses a frozen image encoder to learn a Querying-Transformer (Q-Former) on image-text pairs that is used as the connector. This architecture is adapted in [W. Dai et al. 2023] for dialogue via training with visual instruction tuning. LLaVA [H. Liu et al. 2023b] uses COCO [T.-Y. Lin et al. 2014] annotations and GPT-4 [OpenAI 2023a] to generate visual instruction tuning data in a plain-text pipeline. Combining this generated data with standard VQA datasets (VQAv2 etc.) further boosts performance [H. Liu et al. 2023a]. Different works modify the training approach by using efficient adaptation [Q. Ye et al. 2023; E. J. Hu et al. 2022] multiple training stages [Bai et al. 2023b] or introduce the use of discrete tokens for localization [Z. Peng et al. 2023]. We note however, that most of these works evaluate performance on traditional vision-language datasets like VQAv2 [Antol et al. 2015], which do not consider the extent of hallucinations, a known problem with LLMs [Yue Zhang et al. 2023; H. Ye et al. 2023].

4.3 THRONE

POPE [Yifan Li et al. 2023] and other benchmarks (MME [C. Fu et al. 2023], MM-Bench [Y. Liu et al. 2023]) directly query LVLMS with a restricted desired answer space, e.g. yes-no (MME, POPE) and A-B-C-D multiple choice (MMBench), as



Type II Hallucination Evaluation (POPE)

Instruction [GT]	Response [Prediction]
Is there...	
a person [No]	No, there is no person in the image. [No]
a banana [Yes]	Yes, there is a banana in the image. [Yes]
an apple [No]	No, there is not an apple in the image. [No]
an orange [No]	No, there is no orange in the image. [No]
...in the image?	No Type II Hallucinations Found

Type I Hallucination Evaluation

Instruction	Response
Describe this image in detail.	The image showcases a fruit stand at a grocery store, featuring a variety of fruits on display. There are several bunches of bananas , with some placed in the foreground and others in the background. The bananas are arranged in different sections, creating an appealing presentation for customers . In addition to the bananas , there are also apples and oranges on display. The apples are located towards the left side of the image, while the oranges are placed in the middle and right side of the stand. The fruits are well-organized and presented in an attractive manner, making it an inviting sight for shoppers .
■ - GT Class ■ - Type II Hallucination ■ - Hypothetical Content (not a Hallucination)	
Type I Hallucinations Present and Found	
MSCOCO Object Prediction from Description	
Human	CHAIR
banana apple orange	banana person apple orange
	THRONE (Ours)
	banana apple orange

Figure 4.4: **A Comparison of POPE, CHAIR and THRONE:** Directly querying LVLMs for object existence (person, banana etc.) using concept-specific instructions, as in POPE (bottom left), does not produce the same hallucinations, if any, as using concept-neutral instructions (right). We highlight the Type I hallucinations in orange. CHAIR relies on exact text matching to a fixed set of objects and synonyms, thus incorrectly labels “customers” and “shoppers” as hallucinations, highlighted in red. THRONE is designed for the rich vocabulary and the free-form generations of modern LVLMs by harnessing LMs to establish object existence. By using an LM to pass judgement, our evaluation correctly captures “customers” and “shoppers” as hypothetical content in the free-form generation.

shown in Fig. 4.2. These benchmarks only consider such short answers, whereas the impressive performance in LVLMs is in generating free-form coherent text. Moreover, POPE, which addresses Type II hallucinations, undersamples negative classes meaning hallucinations are dramatically underestimated (see Section 4.5.4 and Fig. 4.7). On the other hand, we use *all classes for every image*, ensuring a full evaluation of Type I hallucinations.

CHAIR [Rohrbach et al. 2018] also evaluates Type I hallucinations, but was developed when vision-language models could only generate short and simple captions similar to those in COCO Captions [H. Fang et al. 2015]. Moreover, it lacks accurate comprehension of responses (see the right side of Fig. 4.4) and ignores the recall of groundtruth objects. In Section 4.5.5, we describe quantitative evaluations, using a human oracle, which demonstrate THRONE *halves* the rate of hallucination misjudgement in CHAIR. See Section 4.2 and the Supplementary Material for more details on CHAIR and its shortcomings. Fig. 4.4 shows an overview of the three aforementioned methods: POPE, CHAIR and our method, THRONE.

4.3.1 Evaluating Hallucinations with THRONE

To address these limitations, we propose a framework, THRONE, shown in Fig. 4.1, to evaluate the prevalence of Type I hallucinations in LVLM responses conditioned on an image and a neutral text prompt.

For each image in a labeled dataset, $\mathcal{I}$, addressing a set of classes, $\mathcal{C}$, the LVLM is queried with the same instruction: `Describe this image in detail.`, regardless of image content. The LVLM response, which is expected to be long free-form text containing an image description, is generated and stored. Next, a publicly available, open-source, *external* language model (LM) performs *abstractive question answering* (AQA) using the LVLM response as context and a question of the form: `Please answer yes or no. Is there {a/an} {object class name} in this image?` or similar, *for every class* in $\mathcal{C}$ (right side of Fig. 4.1). By selecting an appropriate LM and using a simple prompt template (see Section 4.4 for specific details), we ensure the AQA response is either `yes` or `no`—our method does not require any additional parsing. This is in contrast to other works which require added parsing or interpretation by a paid closed-source model.

After performing AQA on each response generated by the LVLM for every class in $\mathcal{C}$, we obtain an array of predicted labels:

$$\hat{\mathbf{Y}} \in \{0, 1\}^{|\mathcal{I}| \times |\mathcal{C}|} \quad (4.1)$$

where 0/1 indicates a negative/positive existence *judgement* by the LM with respect to the relevant LVLM response.

Similarly using the ground-truth data for $\mathcal{I}$, an array of ground-truth labels can be constructed:

$$\mathbf{Y} \in \{0, 1\}^{|\mathcal{I}| \times |\mathcal{C}|} \quad (4.2)$$

Using these two arrays, we calculate four metrics: (1) Overall Precision, P_{ALL} ; (2) Overall Recall, R_{ALL} ; (3) Classwise Precision, P_{CLS} ; and (4) Classwise Recall, R_{CLS} . Overall metrics are calculated in a class-agnostic manner. Classwise metrics are calculated in a class-conscious manner, by computing precision and recall for each category separately and then averaging. This follows common practice in

object detection and instance segmentation [T.-Y. Lin et al. 2014; Everingham et al. 2010].

False positives in LVLMs reduce precision, and are dominated by hallucinations—precision indicates the extent of Type I hallucinations in LVLM responses. The recall metrics inform the level of class coverage by an LVLM when producing image descriptions. The classwise metrics give a general measure of performance as the overall metrics are skewed towards the most common categories. A common way to combine precision, P , and recall, R , metrics is through the generalized F score, F_β :

$$F^\beta = (1 + \beta^2) \cdot \frac{P \cdot R}{(\beta^2 \cdot P) + R} \quad (4.3)$$

Prior work such as POPE [Yifan Li et al. 2023], use the common balanced F-score (or F^1 -score) which equally weights precision and recall. However, given THRONE is concerned with measuring hallucinations and is therefore particularly interested in precision, we choose $\beta = 0.5$ or the $F^{0.5}$ -score, thereby weighing precision twice as important as recall. The $F^{0.5}$ -score is commonly used in pandemic misinformation filters [Purnat et al. 2021], recommender systems [Ng et al. 2014], active stock selection [Rossi et al. 2020] and other areas where false positives are costlier than false negatives. From this we can calculate the overall and classwise $F^{0.5}$ -scores, $F_{\text{ALL}}^{0.5}$ and $F_{\text{CLS}}^{0.5}$, respectively. To mitigate against class imbalance issues, we use $F_{\text{CLS}}^{0.5}$ as the principle metric of comparison between LVLMs in THRONE.

4.3.2 Ensuring Robustness via Ensembling

Any LM used for AQA in THRONE may misjudge Type II hallucinations—no LM is perfect. Fig. 4.5 (top) shows two different variants from the same model family (FLAN-T5 [Longpre et al. 2023]), yielding opposite responses when prompted with the same response and question. Moreover, an LM may yield different answers to semantically identical questions, despite conditioning on the same response—shown in Fig. 4.5 (bottom). To ensure THRONE is robust to spurious performance by any one LM, we ensemble various LMs and semantically equivalent question formats. The use of N distinct LMs and M distinct question formats yields NM answers for each (LVLM response, class) combination. This set of

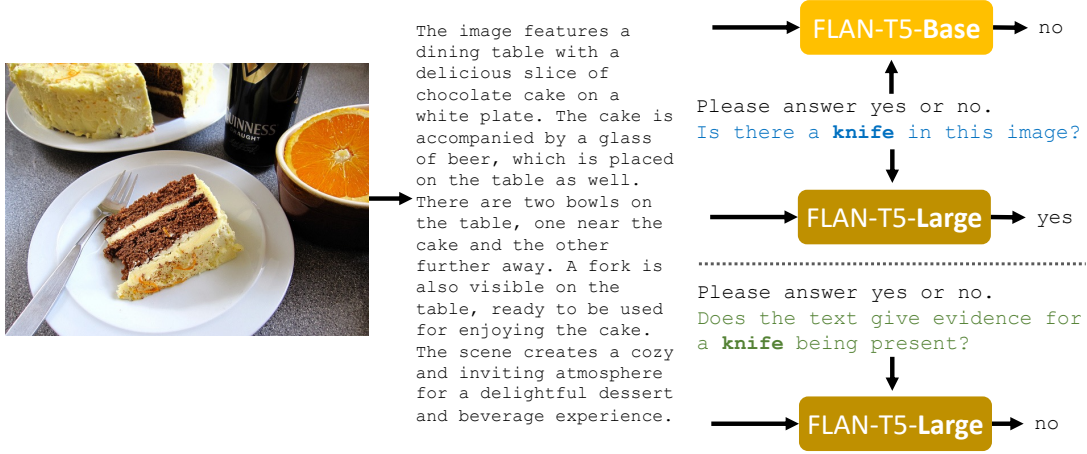


Figure 4.5: **AQA Ensembling in Evaluation:** Using different LMs or different prompts when running AQA on LVLM generated responses can produce opposing answers to *identical prompts* or *identical LMs*. To ensure THRONE is robust to this, we ensemble *multiple LMs* and *multiple prompts* in our evaluation pipeline.

NM answers is combined based on equal voting by each (LM, question) pair to “elect” the predicted answer. Continuing the notation from Eq. (4.1), stacking predictions from each (LM, question) pair yields a 3D array of predicted labels: $\bar{\mathbf{Y}} \in \{0, 1\}^{|\mathcal{I}_{\text{OURS}}| \times |\mathcal{C}_{\text{OURS}}| \times NM}$. To combine the answers from each (LM, question) pair via voting, we require agreement between at least k answers. Where sufficient agreement does not exist we introduce an “ignore” label. Mathematically, the elements in $\hat{\mathbf{Y}}$ (from Eq. (4.1)) are calculated as:

$$\hat{Y}_{i,j} = \begin{cases} 0, & \sum_{k=1}^{NM} \bar{Y}_{i,j,k} \leq (NM - k) \\ 1, & \sum_{k=1}^{NM} \bar{Y}_{i,j,k} \geq k \\ -1, & \text{otherwise} \end{cases}$$

where an “ignore” label exists in $\bar{\mathbf{Y}}$, it is removed from the calculations of P_{ALL} , R_{ALL} , P_{CLS} , and R_{CLS} , as we cannot be confident in the AQA process for that particular (LVLM response, class) combination. The choice of k reflects the desired level of confidence. We make use of a *unanimous* voting mechanism ($k = NM$)—use the prediction only if *all* (LM, question) pairs agree, otherwise ignore. Human evaluation of our benchmark and choice of voting mechanism is found in the Supplementary Material. Once we have applied this voting mechanism, we can calculate the metrics described at the end of Section 4.3.1.

4.4 Implementation

We provide details of our framework regarding the selection of public datasets, public LMs and LM prompts.

4.4.1 Dataset

In our benchmark we use COCO [T.-Y. Lin et al. 2014] for a number of reasons: (1) its annotations of 80 categories are exhaustive *at an image level* (image-level recall $\approx 99\%$ [T.-Y. Lin et al. 2014])—if there are many `book` instances in an COCO image, at least one is annotated with bounding boxes; (2) many LVLMs are partly trained on COCO data and so should be familiar with the set of categories; (3) its images generally contain complex scenes suitable for generating long free-form descriptions unlike image recognition datasets like ImageNet [J. Deng et al. 2009].

We utilize the validation set of COCO 2017, which contains $|\mathcal{I}| = 5000$ images and $|\mathcal{C}| = 80$ categories. Using the single LVLM text prompt `Describe this image in detail.`, we generate 5000 responses. As we query each LVLM response for each category in $\mathcal{C}$, a single LM performs AQA $|\mathcal{I}| \times |\mathcal{C}| = 400k$ times across the LVLM responses—one instance of AQA per (image response, class) pair. In Section 4.5.3, we also present results using Objects365 [Shao et al. 2019] which is rarely used in LVLM training.

4.4.2 Language Models

To assess Type I hallucinations in an LVLM response using THRONE, we require a language model (LM) which can answer questions on the existence of object categories based on the LVLM response. MMBench [Y. Liu et al. 2023] makes use of a ChatGPT model to identify multiple choice answer selections and still reports mistakes. For THRONE, we choose FLAN-T5 models [Longpre et al. 2023; Raffel et al. 2020]. We make this choice because they: (1) have undergone instruction tuning with thousands of tasks [Longpre et al. 2023]; (2) are open-source and therefore accessible the community; (3) can fit locally on a single GPU for the models we choose; (4) follow the request to only respond `yes` or `no`; and (5) are optimized for use in the free and public Text-Generation-Inference API [Face 2023] for acceleration. As described in Section 4.3.2, we utilize N LMs to ensure

Model	L	P_{ALL}	R_{ALL}	F^1_{ALL}	$F^{0.5}_{ALL}$	P_{CLS}	R_{CLS}	F^1_{CLS}	$F^{0.5}_{CLS}$
LLaMA-Adapter-v2 [P. Gao et al. 2023]	514	63.6	73.3	68.1	65.3	68.2	70.6	69.4	68.7
LLaMA-Adapter-v2.1 [P. Gao et al. 2023]	512	63.8	73.7	68.4	65.5	67.4	71.2	69.3	68.1
InstructBLIP [W. Dai et al. 2023]	525	70.8	74.3	72.5	71.5	77.2	71.9	74.5	76.1
Otter-Image [Bo Li et al. 2023]	257	33.0	31.2	32.1	32.7	25.2	16.9	20.2	22.9
MiniGPT4 [D. Zhu et al. 2023]	473	81.7	59.8	69.0	76.1	79.9	61.8	69.7	75.5
MiniGPT-v2 [J. Chen et al. 2023]	381	79.0	66.6	72.3	76.2	77.6	67.0	71.9	75.2
Qwen-VL-Chat [Bai et al. 2023b]	524	74.7	77.2	75.9	75.2	78.0	76.1	77.0	77.6
mPLUG-Owl [Q. Ye et al. 2023]	555	55.5	71.9	62.6	58.1	66.3	68.3	67.3	66.7
LRV-Instruction-v2 [F. Liu et al. 2023]	103	82.0	56.7	67.0	75.3	78.4	58.8	67.2	73.5
LLaVA-v1.3* [H. Liu et al. 2023b]	532	80.5	65.2	72.1	76.9	79.9	65.3	71.9	76.5
LLaVA-v1.5 [H. Liu et al. 2023a]	509	68.1	61.0	64.4	66.6	69.9	56.4	62.5	66.8

Table 4.1: **THRONE Results with COCO** for a selection of instruction-tuned LVLMS. We select $F^{0.5}_{CLS}$ as the principal metric for evaluation in our benchmark to balance across classes and to prioritize precision (which reflects the extent of hallucination) over recall. Best and second-best performance are denoted by **blue** and **red**, respectively. *Our implementation using official code to enable fair comparison. L corresponds to the median response length (measured in # of characters).

our method is robust. Specifically, we use $N = 3$ variants of FLAN-T5, namely:

- FLAN-T5-Base (250M parameters)
- FLAN-T5-Large (780M parameters)
- FLAN-T5-XL (3B parameters)

4.4.3 Prompt Ensembling

To guarantee each of the FLAN-T5 variants we described above, faithfully produce responses of either **yes** or **no** only during AQA, we use the following input template to each LM, reflecting the format used when training FLAN-T5¹:

Text: *{LVM Response}* Read the text about an image and answer the question. Question:
Please answer yes or no. *{Question}*

We use $M = 3$ formulations of question, all with the same semantic meaning:

- Is there a/an *{class_name}* in this image?
- Does the text imply a/an *{class_name}* is in the image?
- Does the text explicitly mention a/an *{class_name}* is in the image?

As outlined in Section 4.3.2, we use a *unanimous voting* mechanism to combine the answers from each (LM, question) pair and so $k = 3 \times 3 = 9$.

¹<https://tinyurl.com/5n6nexze>

4.5 Evaluation Results

In this section, we: (1) outline our LVLm selection and reasoning; (2) present THRONE on COCO for evaluating Type I hallucinations; (3) extend THRONE to Objects365 (containing a larger vocabulary); (4) analyze and extend POPE to enable improved evaluation of Type II hallucinations; and (5) highlight results from our ablation studies found in the Supplemental Material.

4.5.1 Models

For fair comparison between existing LVLms, each publicly available model we evaluate uses an LLM with ~ 7 B parameters. The LVLms generally have different sized image encoders—e.g. LLaVA [H. Liu et al. 2023b] uses a CLIP ViT-L/14 [Dosovitskiy et al. 2021; Radford et al. 2021] with an input resolution of 336×336 , while InstructBLIP [W. Dai et al. 2023] uses a ViT-g/14 [Zhai et al. 2022] trained with EVA [Y. Fang et al. 2023] and an input resolution of 224×224 —we note pre-trained image encoder size and resolution is not something we are able to control easily. Each model we consider contains instruction tuning in the final training phase—instruction tuned models provide free-form descriptions; THRONE focuses on models which *can* generate free-form descriptions. E.g. BLIP-2 [Junnan Li et al. 2023] produces “detailed descriptions” which have a median response length of 31 characters, resembling captions in the COCO Caption dataset, which are often incoherent and lack detail—whereas InstructBLIP produces responses with improved style and detail with a median response length of 525. The Supplementary Material contains comprehensive details on each of the LVLms we evaluate.

4.5.2 THRONE Results on COCO

Results are shown in Table 4.1. The principal metric we use to judge model performance is the classwise $F^{0.5}$ -score (highlighted gray). We also report all the metrics outlined in Section 4.3.1, $(P, R, F^1, F^{0.5})$ for overall (left) and class-wise averaging (right), utilizing the *unanimous* voting presented in Section 4.3.2. See the Supplementary Material for results and an analysis of different voting mechanisms. We observe that Qwen-VL performs best on our benchmark, with LLaVA-v1.3 performing second best. Interestingly, we find the successor model, LLaVA-

v1.5, performs markedly worse than its predecessor. Similarly, despite LLaMA-Adapter-v2.1 and MiniGPT-v2 gaining a lot of ground in POPE (see Fig. 4.7) compared to LLaMA-Adapter and MiniGPT4, both model perform slightly worse on THRONE. These observations demonstrate that improvements on other benchmarks (POPE, MME, MMBench etc.) may be orthogonal and potentially at odds to improved performance on THRONE. Using the results of the 11 LVLMS we evaluate, THRONE and POPE, which measure Type I and Type II hallucinations, respectively, have a Spearman’s rank correlation coefficient of just 0.2—the relationship between performance on POPE and THRONE *on the same dataset* is far from monotonic. For class-wise precision, P_{CLS} , the best performing models (MiniGPT and LLaVA-v1.3) hallucinate $\sim 20\%$ of the objects—we show in the Supplementary Material that the vast majority of false positive objects in the free-form image descriptions evaluated are obvious hallucinations rather than misclassifications of visually similar objects e.g. mistaking a squash racket for a tennis racket. These results demonstrate that much work still remains to adequately suppress Type I hallucinations in LVLMS.

Model	P_{ALL}	R_{ALL}	F^1_{ALL}	$F^{0.5}_{ALL}$	P_{CLS}	R_{CLS}	F^1_{CLS}	$F^{0.5}_{CLS}$
LLaMA-Adapter-v2 [P. Gao et al. 2023]	46.7	33.9	39.3	43.4	48.9	28.5	36.0	42.8
LLaMA-Adapter-v2.1 [P. Gao et al. 2023]	46.8	34.0	39.4	43.5	48.8	28.8	36.2	42.8
InstructBLIP [W. Dai et al. 2023]	54.5	37.2	44.2	49.8	53.7	33.6	41.3	48.0
Otter-Image [Bo Li et al. 2023]	21.4	12.7	16.0	18.8	9.5	4.4	6.0	7.7
MiniGPT4 [D. Zhu et al. 2023]	53.0	32.9	40.6	47.2	49.9	31.9	39.0	44.9
MiniGPT-v2 [J. Chen et al. 2023]	54.5	36.0	43.4	49.4	51.3	34.6	41.3	46.8
Qwen-VL-Chat [Bai et al. 2023b]	58.3	39.1	46.9	53.1	57.8	35.9	44.3	51.5
mPLUG-Owl [Q. Ye et al. 2023]	43.7	33.4	37.8	41.2	48.2	29.0	36.2	42.6
LRV-Instruction-v2 [F. Liu et al. 2023]	57.5	26.7	36.5	46.7	51.4	26.6	35.1	43.3
LLaVA-v1.3* [H. Liu et al. 2023b]	57.6	32.9	41.9	50.1	52.6	30.5	38.6	45.9
LLaVA-v1.5 [H. Liu et al. 2023a]	54.0	39.5	45.6	50.3	53.9	34.3	41.9	48.4

Table 4.2: **THRONE Evaluation with Objects365.** Evaluation results for a selection of instruction-tuned LVLMS, we use a subset of Objects365 for the THRONE evaluation. Best and second-best performance are denoted by **blue** and **red**, respectively. *Our implementation using official code to enable fair comparison.

4.5.3 THRONE Results on Objects365

To demonstrate generality beyond COCO, which many LVLMS train on directly or indirectly, we apply THRONE to the Objects365 dataset [Shao et al. 2019]. Like COCO, Objects365 is exhaustive in its image-level class labeling, but has a larger object vocabulary and is not used as training data in the LVLMS we evaluate. To

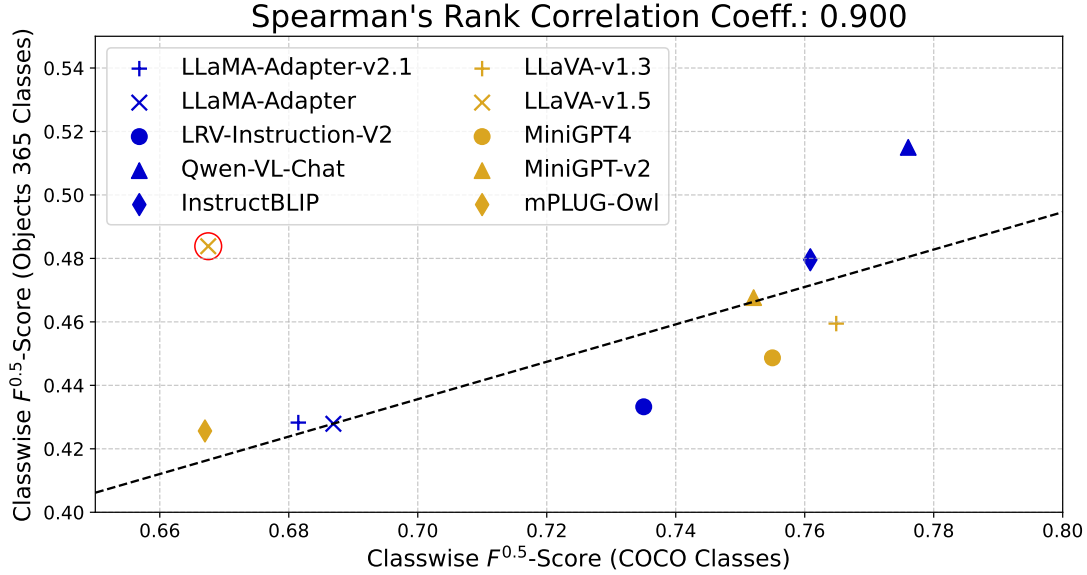


Figure 4.6: **Comparing THRONE on COCO and Objects365.** We observe that despite variations in the data distribution, THRONE metrics, which measure Type I hallucinations generalize and have a strong Spearman’s rank correlation of $r = 0.900$. LLaVA-v1.5 designated as an outlier and ignored when calculating ranking correlation.

gather a manageable subset of the Objects365 validation set (80k images), we use the natural sampling algorithm from [K. Lee et al. 2022], resulting in 5110 images (the COCO validation set has 5k images). We present the results for THRONE on Objects365 in Table 4.2. We observe again that Qwen-VL-Chat [Bai et al. 2023b] performs best on THRONE. Unlike for COCO (Table 4.1), we find that LLaVA-v1.5 [H. Liu et al. 2023a] performs better than LLaVA-v1.3 [H. Liu et al. 2023b] on Objects365, but note this is largely due to increased recall. This likely results from additional training data exposing LLaVA-v1.5 to a greater number of concepts than LLaVA-v1.3, which only trains on visual instruction tuning data derived from COCO. Similarly, LLaMA-Adapter-v2.1 and MiniGPT-v2 marginally improve over LLaMA-Adapter and MiniGPT4, respectively, by having better recall, despite showing much larger gains on POPE. Figure 4.6 shows the strong correlation in THRONE performance between evaluating on COCO and Objects365. This demonstrates measuring Type I hallucinations in an LVLM using THRONE with a relatively small dataset like COCO, is indeed indicative of the intrinsic level of Type I hallucination in a given LVLM.

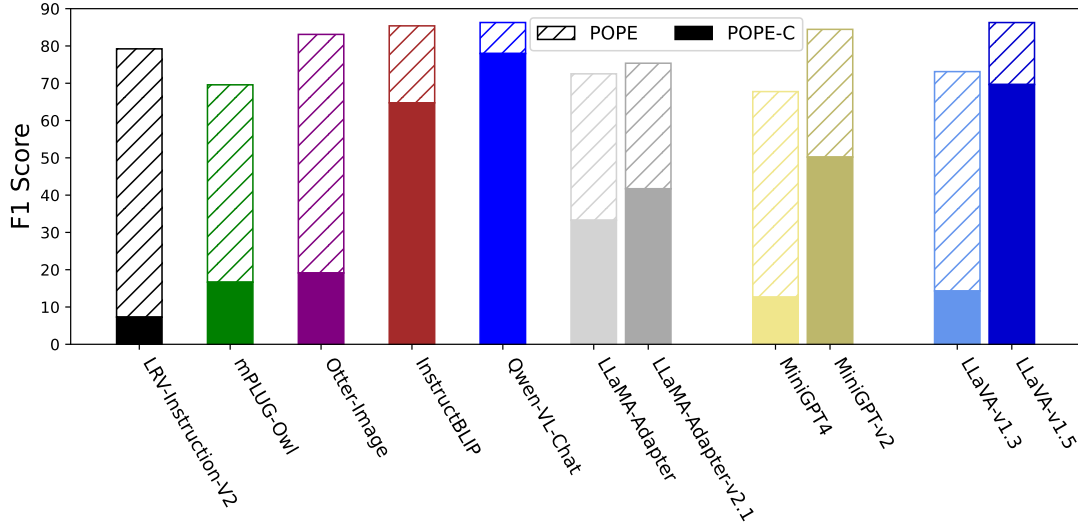


Figure 4.7: **Instability of POPE to complete evaluation of Type II Hallucinations.** Extending POPE to an exhaustive analysis on all COCO images and classes (POPE-C) leads to a dramatic reduction in performance across all models, demonstrating the fragility of POPE to the sampling mechanism. By undersampling negative classes, POPE severely underestimates Type II hallucinations in LVLMs.

4.5.4 Completing POPE for Type II Hallucinations

Our experiments have used THRONE to evaluate the prevalence of Type I hallucinations in LVLM responses on COCO. POPE evaluates Type II hallucinations on COCO, but we find POPE is largely incomplete. First, POPE only evaluates on 500 COCO validation set images. Second, for each image only a subset of classes (at most 12) are evaluated—each image is only queried with 15% of possible questions. Finally, POPE artificially balances evaluation questions between positives and negatives, despite object class existence in images being inherently imbalanced. The above reasons make the evaluation of Type II hallucinations using POPE insufficient (see Section 4.2 for more details on POPE). To correct this, we complete POPE by using all images to exhaustively query each LVLM *for every class* in the COCO vocabulary, as done in THRONE. We name this POPE-Complete (POPE-C). Fig. 4.7 shows the extreme difference in evaluation between POPE and our exhaustive version, POPE-C—reporting F^1 -score (POPE does not utilize $F^{0.5}$ -score). As POPE only evaluates at most 9 negative classes, only a small subset of potential hallucinations of COCO classes are evaluated, thereby heavily underestimating the extent of Type II hallucination. For each LVLM analyzed, we observe a large—in many cases an extreme—reduction in precision and therefore

in F1-score. E.g. Otter-Image, LRV-Instruction-v2, LLaVA-v1.3, MiniGPT4 and mPLUG-Owl all have an F1-score reduction of at least 50%.

Our evaluation contains three pairs of LVLMs in which one is the follow-up work to the other: (MiniGPT4 $\rightarrow$ MiniGPT-v2, LLaMA-Adapter-v2 $\rightarrow$ v2.1, LLaVA-v1.3 $\rightarrow$ LLaVA-v1.5), where each follow-up work generally trains on more data for more tasks with a more advanced language model. Comparing the right hand side of Fig. 4.7 to Table 4.1, we observe that these follow-up works generally show an improvement in POPE (and POPE-C), but a small reduction in performance on THRONE with COCO. This observation suggests that progress in reducing Type II hallucinations can be orthogonal to reducing Type I hallucinations.

4.5.5 Ablations

In the Supplementary Material we present three key ablation experiments and give an executive summary of results here.

First, after subsampling COCO images and LVLM responses, we replace the LMs in THRONE with human judgement as an oracle for Type I hallucination occurrence. When comparing THRONE and CHAIR with human judgements, we estimate using THRONE improves the precision of judging Type I hallucinations to 96% from 91% when using CHAIR—this is a reduction in false discovery rate of more than 50%. Note, we find most estimated errors in THRONE arise from the particular class definitions in COCO, e.g. the COCO class `tv` includes computer monitors, which the oracle judgement is aware of.

Second, we apply the same class (and image) sampling strategy as in POPE to THRONE and show this sampling overestimates $F_{CLS}^{0.5}$ by an average of 12.3 points compared to the complete use of classes in THRONE (Table 4.1).

Finally, we vary the choice of k i.e. the voting mechanism used to combine answers from multiple (LM, question) pairs. We use the *unanimous* voting mechanism ($k = 9$) in THRONE to minimize the false discovery rate and find the valid alternatives of *simple majority* ($k = 5$) or *all-but-one* ($k = 8$) voting mechanisms have strong correlations and rank correlations across all compute metrics, in THRONE, of > 0.99 and > 0.94 , respectively.

4.6 Improved Baselines

Much needs to be done to study Type I hallucinations. As a first step to their mitigation, we demonstrate a baseline method to augment the visual instruction tuning data for LLaVA models [H. Liu et al. 2023b; H. Liu et al. 2023a], yielding improvements on THRONE while maintaining similar performance regarding Type II hallucinations on POPE.

4.6.1 Visual Instruction Tuning Data Augmentation

Similar to chain-of-thought learning [Wei et al. 2022], during instruction tuning, we augment all visual instruction tuning samples constructed by LLaVA [H. Liu et al. 2023b] by prepending the task of enumerating a list of objects (present and absent) and indicating approximate locations, if applicable. Other than this, the LLaVA data and training pipeline remains unchanged. To generate the new data for this object enumeration task, we use the same COCO bounding box annotations used to generate the vision instruction tuning data in LLaVA. The simple text-only format (no special tokens) we use is:

```
Instruction: <image> Give a list of objects and locations
            in the image.
Response:   {class_name_1} [{location_1}/absent]
            ...
            {class_name_N} [{location_N}/absent]
```

where `location_i` is a plain text indicator representing the location of the center point of the relevant object in the image on a 3×3 grid (e.g. `bottom left`). To provide negatives in the training data, if `class_name_i` is not present, we use the plain text indicator `absent`. Prior work [Y. Zhou et al. 2023] shows classes which co-occur often in training data are the most common hallucinations, therefore we bias our negative sampling towards class pairs that frequently co-occur using a correlation matrix. We detail and ablate this choice in the Supplementary Material.

4.6.2 Improved Baseline Results

Table 4.8 shows the result of evaluating our improved baseline method on THRONE, POPE, and POPE-C. During inference, we approximate our training data augmen-

Model	Object Enumeration Data	THRONE			POPE			POPE-C		
		P_{CLS}	R_{CLS}	$F_{CLS}^{0.5}$	P	R	F^1	P	R	F^1
LLaVA-v1.3	X	79.9	65.3	76.5	58.0	98.4	73.0	7.7	99.2	14.3
	COCO	83.2	68.8	79.9	73.2	88.2	80.0	9.8	69.4	17.2
	COCO + VG	86.2	67.0	81.5	83.0	82.5	82.8	13.8	50.4	21.7
LLaVA-v1.5	X	69.9	56.4	66.8	81.9	90.8	86.1	58.7	85.7	69.7
	COCO	87.2	76.6	84.9	88.6	85.3	87.0	58.9	87.5	70.4
	COCO + VG	86.1	77.0	84.1	89.8	83.7	86.7	64.5	86.1	73.7

Table 4.3: **Improved Baseline via Object Enumeration:** Adding our object enumeration task to LLaVA training and inference leads to large improvements on THRONE particularly in terms of classwise precision, P_{CLS} , over standard LLaVA models, demonstrating a reduction in Type I hallucinations. Additionally, this method yields small improvements in POPE and POPE-C, showing a reduction in Type II hallucinations.

tation by first prompting the LVLM to perform the object enumeration tasks *and then* generating a response to the prompt from the relevant benchmark. We additionally show results of utilizing VisualGenome bounding box annotations [Krishna et al. 2017] on the COCO images where available. On THRONE, we observe a large increase in classwise precision and therefore $F_{CLS}^{0.5}$, particularly for LLaVA-v1.5, demonstrating the ability of our method to reduce Type I hallucinations. Moreover, on POPE and POPE-C, using our object enumeration yields small improvements in precision, indicating reduced Type II hallucinations as well. In the Supplementary Material, we ablate the sampling of negatives during object enumeration training and the effect of removing the object enumeration task during inference.

4.7 Conclusion

We establish a novel benchmark, THRONE, for evaluating hallucinations generated by LVLMs in free-form image descriptions i.e. *Type I hallucinations*. Our benchmark utilizes multiple LMs and prompt formats with a simple voting mechanism to yield an accurate evaluation of Type I hallucinations in LVLM responses. We ensure accessibility for all by utilizing open-source LMs capable of running on a single commercial GPU. Using THRONE, we benchmark 11 publicly available LVLMs on two datasets, COCO and Objects365, and demonstrate limited progress has been made in specifically addressing Type I hallucinations. Moreover, we show how the established benchmark, POPE, underestimates *Type II*

hallucinations, which occur in response to specific questions e.g. yes-no questions and present results for a completed version (POPE-C) to enable a comparison of Type I hallucinations through THRONE and Type II hallucinations using POPE-C. Finally, we propose a simple data augmentation for LVLM training that can result in a large reduction in Type I hallucinations whilst maintaining or improving Type II hallucination performance.

Limitations and Ethical Considerations are discussed in the Supplementary Material.

4.A Evaluated LVLM Details

In this section, we briefly outline details in the LVLMs we evaluate with THRONE, such as design choices and training data.

LLaMA-Adapter-v2 [P. Gao et al. 2023]

- LLM: LLaMA-v1-7b [Touvron et al. 2023a]
- Image Encoder: CLIP ViT-L/14 [Radford et al. 2021]
- Vision-Language Connector: Linear projection, learnt visual queries and additional vision transformer blocks
- Pretraining Data: LAION-400M [Schuhmann et al. 2021], COYO-700M [Byeon et al. 2022], MMC4 [W. Zhu et al. 2023], SBU [Ordonez et al. 2011], CC3M [Sharma et al. 2018], and COCO Captions [H. Fang et al. 2015]
- Finetuning Data: GPT4LLM [B. Peng et al. 2023] and LLaVA [H. Liu et al. 2023b]

LLaMA-Adapter-v2.1 [P. Gao et al. 2023]

Same as LLaMA-Adapter-v2 with one modification.

- Finetuning Data: GPT4LLM [B. Peng et al. 2023], LLaVA [H. Liu et al. 2023b] and VQAv2 [Antol et al. 2015]

InstructBLIP [W. Dai et al. 2023]

- LLM: Vicuna-v1.1-7b [Chiang et al. 2023]
- Image Encoder: EVA ViT-g/14 [Y. Fang et al. 2023]
- Visual-Language Connector: Querying Transformer (Q-Former)
- Pretraining Data: LAION400M [Schuhmann et al. 2021], CC3M [Sharma et al. 2018], CC12M [Changpinyo et al. 2021], Visual Genome [Krishna et al. 2017] and COCO Captions [H. Fang et al. 2015]

- Finetuning Data: COCO Captions [H. Fang et al. 2015], TextCaps [Sidorov et al. 2020], VQAv2 [Antol et al. 2015], OKVQA [Marino et al. 2019], A-OKVQA [Schwenk et al. 2022], OCR-VQA [Mishra et al. 2019] and LLaVA [H. Liu et al. 2023b]

Otter-Image [Bo Li et al. 2023]

- LLM: MPT-7b [Team 2023] (initialized from OpenFlamingo weights [Awadalla et al. 2023])
- Image Encoder: CLIP ViT-L/14 [Radford et al. 2021]
- Visual-Language Connector: Cross-attention modules between visual and language features
- Pretraining Data: MMC4 [W. Zhu et al. 2023], LAION-2B-en [Schuhmann et al. 2022] and synthetic data from ChatGPT [OpenAI 2023b]
- Finetuning Data: MIMIC-IT [Bo Li et al. 2023]

MiniGPT-4 [D. Zhu et al. 2023]

- LLM: Vicuna-v0-7b [Chiang et al. 2023]
- Image Encoder: EVA ViT-g/14 [Y. Fang et al. 2023]
- Vision-Language Connector: Linear projection and Q-Former (initialized and frozen from BLIP-2 [Junnan Li et al. 2023])
- Pretraining Data: CC3M [Sharma et al. 2018], CC12M [Changpinyo et al. 2021], SBU [Ordonez et al. 2011] and LAION-400M [Schuhmann et al. 2021]
- Finetuning Data: Synthetic data using ChatGPT [OpenAI 2023b] and a sample of CC3M/CC12M images [Changpinyo et al. 2021; Sharma et al. 2018]

MiniGPT-v2 [J. Chen et al. 2023]

- LLM: Llama2-7b-chat [Touvron et al. 2023b]

- Image Encoder: EVA ViT-g/14 [Y. Fang et al. 2023]
- Vision-Language Connector: Linear projection
- Pretraining Data: GRiT-20M [Z. Peng et al. 2023], COCO Captions [H. Fang et al. 2015], TextCaps [Sidorov et al. 2020], RefCOCO/+g [Kazemzadeh et al. 2014; J. Mao et al. 2016], Visual Genome [Krishna et al. 2017], GQA [Hudson and Manning 2019], VQAv2 [Antol et al. 2015], OKVQA [Marino et al. 2019], A-OKVQA [Schwenk et al. 2022] and OCR-VQA [Mishra et al. 2019]
- Finetuning Data: COCO Captions [H. Fang et al. 2015], TextCaps [Sidorov et al. 2020], RefCOCO/+g [Kazemzadeh et al. 2014; J. Mao et al. 2016], Visual Genome [Krishna et al. 2017], GQA [Hudson and Manning 2019], VQAv2 [Antol et al. 2015], OKVQA [Marino et al. 2019], A-OKVQA [Schwenk et al. 2022], LLaVA [H. Liu et al. 2023b], Flickr30k [Plummer et al. 2015] and LLaVA [H. Liu et al. 2023b]

Qwen-VL-Chat [Bai et al. 2023b]

- LLM: Qwen-LM [Bai et al. 2023a]
- Image Encoder: OpenCLIP ViT-g [Ilharco et al. 2021]
- Vision-Language Connector: Single cross-attention block between image features and learnt visual queries
- Pretraining Data: LAION-2B-en [Schuhmann et al. 2022], COYO-700M [Byeon et al. 2022], CC3M [Sharma et al. 2018], CC12M [Changpinyo et al. 2021], SBU [Ordonez et al. 2011], COCO Caption [H. Fang et al. 2015] and DataComp [Gadre et al. 2023]
- Finetuning Data: LAION-2B-en/zh [Schuhmann et al. 2022], COYO-700M [Byeon et al. 2022], CC3M [Sharma et al. 2018], CC12M [Changpinyo et al. 2021], SBU [Ordonez et al. 2011], DataComp [Gadre et al. 2023], COCO Caption [H. Fang et al. 2015], GQA [Hudson and Manning 2019], Visual Genome [Krishna et al. 2017], VQAv2 [Antol et al. 2015], DVQA [Kafle et al. 2018], OCR-VQA [Mishra et al. 2019], DocVQA [Mathew et al. 2021], TextVQA [Singh et al. 2019], ChartQA [Masry et al. 2022], AI2D [Kembhavi et al. 2016],

GRiT [Z. Peng et al. 2023], VisualGenome [Krishna et al. 2017], RefCO-CO/+g [Kazemzadeh et al. 2014; J. Mao et al. 2016], Synthetic OCR data, Common Crawl PDF [Lab 2021]

mPLUG-Owl [Q. Ye et al. 2023]

- LLM: LLaMA-7b [Touvron et al. 2023a]
- Image Encoder: CLIP ViT-L/14 [Radford et al. 2021]
- Vision-Language Connector: Cross-attention and learnt visual queries
- Pretraining Data: LAION-400M [Schuhmann et al. 2021], COYO-700M [Byeon et al. 2022], CC3M [Sharma et al. 2018] and COCO Captions [H. Fang et al. 2015]
- Finetuning Data: LLaVA [H. Liu et al. 2023b], Alpaca [Taori et al. 2023], Vicuna [Chiang et al. 2023] and Baize [C. Xu et al. 2023]

LRV-Instruction-v2 [F. Liu et al. 2023]

Same as mPLUG-Owl with one modification.

- Finetuning Data: LRV-Instruction-v2 [F. Liu et al. 2023], Alpaca [Taori et al. 2023], Vicuna [Chiang et al. 2023] and Baize [C. Xu et al. 2023]

LLaVA-v1.3 [H. Liu et al. 2023b]

- LLM: Vicuna-v1.3-7b [Chiang et al. 2023]
- Image Encoder: CLIP ViT-L/14 [Radford et al. 2021]
- Pretraining Data: LAION-400M [Schuhmann et al. 2021], CC3M [Sharma et al. 2018] and SBU [Ordonez et al. 2011]
- Finetuning Data: LLaVA [H. Liu et al. 2023b]

LLaVA-v1.5 [H. Liu et al. 2023a]

- LLM: Vicuna-v1.5-7b [Chiang et al. 2023]
- Image Encoder: CLIP ViT-L/14 [Radford et al. 2021]

- Pretraining Data: LAION-400M [Schuhmann et al. 2021], CC3M [Sharma et al. 2018] and SBU [Ordonez et al. 2011]
- Finetuning Data: LLaVA [H. Liu et al. 2023b], VQAv2 [Antol et al. 2015], GQA [Hudson and Manning 2019], TextVQA [Singh et al. 2019], OCR-VQA [Mishra et al. 2019] and VisualGenome [Krishna et al. 2017]

4.B Voting Mechanism Ablation

Model	k	P_{ALL}	R_{ALL}	F_{ALL}^1	$F_{\text{ALL}}^{0.5}$	P_{CLS}	R_{CLS}	F_{CLS}^1	$F_{\text{CLS}}^{0.5}$	Ignore %
LLaMA-Adapter-v2	9	63.6	73.3	68.1	65.3	68.2	70.6	69.4	68.7	2.4
	5	61.8	75.0	67.7	64.0	65.7	72.0	68.7	66.9	0.0
	8	63.4	73.8	68.2	65.2	68.0	70.8	69.4	68.5	1.4
LLaMA-Adapter-v2.1	9	63.8	73.7	68.4	65.5	67.4	71.2	69.3	68.1	2.4
	5	61.7	75.3	67.8	64.0	64.7	72.5	68.4	66.1	0.0
	8	63.6	74.1	68.5	65.5	67.2	71.5	69.3	68.0	1.5
InstructBLIP	9	70.8	74.3	72.5	71.5	77.2	71.9	74.5	76.1	2.5
	5	68.2	77.2	72.4	69.8	73.2	74.3	73.7	73.4	0.0
	8	70.6	75.1	72.8	71.4	76.8	72.3	74.5	75.9	1.5
Otter-Image	9	33.0	31.2	32.1	32.7	25.2	16.9	20.2	22.9	8.5
	5	25.6	34.7	29.5	27.1	16.4	20.1	18.0	17.0	0.0
	8	32.4	31.8	32.1	32.3	23.9	17.2	20.0	22.2	4.8
MiniGPT-4	9	81.7	59.8	69.0	76.1	79.9	61.8	69.7	75.5	2.9
	5	74.8	64.9	69.5	72.6	73.0	65.4	69.0	71.3	0.0
	8	80.8	61.1	69.6	75.9	79.1	62.4	69.8	75.1	1.8
MiniGPT-v2	9	79.0	66.6	72.3	76.2	77.6	67.0	71.9	75.2	2.8
	5	73.6	71.4	72.5	73.1	72.1	70.5	71.3	71.7	0.0
	8	78.4	67.8	72.7	76.0	76.9	67.7	72.0	74.8	1.8
Qwen-VL-Chat	9	74.7	77.2	75.9	75.2	78.0	76.1	77.0	77.6	2.7
	5	71.4	79.6	75.2	72.9	73.9	78.0	75.9	74.7	0.0
	8	74.4	77.9	76.1	75.1	77.6	76.5	77.1	77.4	1.6
mPLUG-Owl	9	55.5	71.9	62.6	58.1	66.3	68.3	67.3	66.7	2.4
	5	54.3	73.9	62.6	57.3	63.7	69.9	66.6	64.8	0.0
	8	55.5	72.6	62.9	58.2	66.2	68.6	67.4	66.7	1.4
LRV-Instruction-v2	9	82.0	56.7	67.0	75.3	78.4	58.8	67.2	73.5	3.6
	5	77.5	60.8	68.1	73.5	74.6	61.9	67.7	71.7	0.0
	8	81.7	57.9	67.8	75.5	78.0	59.4	67.4	73.4	2.0
LLaVA-v1.3	9	80.5	65.2	72.1	76.9	79.9	65.3	71.9	76.5	2.4
	5	76.4	68.7	72.4	74.7	75.6	68.0	71.6	73.9	0.0
	8	80.0	66.3	72.5	76.9	79.4	65.8	72.0	76.3	1.4
LLaVA-v1.5	9	68.1	61.0	64.4	66.6	69.9	56.4	62.5	66.8	2.7
	5	65.5	62.6	64.0	64.9	67.1	57.8	62.1	65.0	0.0
	8	67.8	61.5	64.5	66.4	69.6	56.7	62.5	66.6	1.3

Table 4.4: **Comparison of Voting Mechanisms in THRONE.** We compare three different voting mechanisms: *unanimous*, $k = 9$; *simple majority*, $k = 5$; and *jury majority*, $k = 8$. Moreover in each case we report the number of ignore labels as a result of each voting mechanism. The number of labels is 400,000 for a given LVLM. We find that the number of ignore labels is low in almost all cases and metrics are strongly correlated (> 0.99). In THRONE, we use a *unanimous* voting mechanism ($k = 9$) to minimize the likelihood of hallucination judgement errors.

4.C THRONE vs. POPE Sampling

Model	Sampling	P_{ALL}	R_{ALL}	F^1_{ALL}	$F^{0.5}_{ALL}$	P_{CLS}	R_{CLS}	F^1_{CLS}	$F^{0.5}_{CLS}$
LLaMA-Adapter-v2	THRONE	63.6	73.3	68.1	65.3	68.2	70.6	68.3	68.0
	POPE	77.3	73.2	75.2	76.5	84.6	70.0	76.9	81.2
	Δ	(-13.7)	0.2	(-7.1)	(-11.1)	(-16.3)	0.5	(-8.6)	(-13.2)
LLaMA-Adapter-v2.1	THRONE	63.8	73.7	68.4	65.5	67.4	71.2	68.1	67.5
	POPE	79.2	73.2	76.1	77.9	85.4	71.8	76.3	80.6
	Δ	(-15.4)	0.5	(-7.7)	(-12.4)	(-18.0)	(-0.6)	(-8.2)	(-13.2)
InstructBLIP	THRONE	70.8	74.3	72.5	71.5	77.2	71.9	73.1	75.2
	POPE	82.2	74.4	78.1	80.5	85.0	70.0	77.9	82.6
	Δ	(-11.4)	0.0	(-5.5)	(-9.0)	(-7.8)	1.9	(-4.8)	(-7.4)
Otter-Image	THRONE	33.0	31.2	32.1	32.7	25.2	16.9	18.7	21.5
	POPE	66.5	34.9	45.7	56.3	70.7	17.3	35.6	49.2
	Δ	(-33.5)	(-3.7)	(-13.7)	(-23.6)	(-45.5)	(-0.4)	(-17.0)	(-27.7)
MiniGPT-4	THRONE	81.7	59.8	69.0	76.1	79.9	61.8	67.6	73.6
	POPE	89.1	58.7	70.8	80.7	88.2	60.0	70.5	78.9
	Δ	(-7.5)	1.1	(-1.7)	(-4.7)	(-8.2)	1.8	(-2.9)	(-5.4)
MiniGPT-v2	THRONE	79.0	66.6	72.3	76.2	77.6	67.0	70.4	74.0
	POPE	88.3	65.8	75.4	82.7	89.8	68.5	76.3	82.6
	Δ	(-9.3)	0.8	(-3.1)	(-6.5)	(-12.2)	(-1.5)	(-5.9)	(-8.6)
Qwen-VL-Chat	THRONE	74.7	77.2	75.9	75.2	78.0	76.1	76.3	77.1
	POPE	83.8	76.6	80.0	82.2	87.8	74.6	80.0	84.4
	Δ	(-9.1)	0.7	(-4.1)	(-7.1)	(-9.8)	1.5	(-3.7)	(-7.3)
mPLUG-Owl	THRONE	55.5	71.9	62.6	58.1	66.3	68.3	65.2	65.3
	POPE	72.9	71.2	72.0	72.6	82.0	64.5	74.1	78.6
	Δ	(-17.4)	0.7	(-9.4)	(-14.4)	(-15.6)	3.7	(-8.9)	(-13.3)
LRV-Instruction-v2	THRONE	82.0	56.7	67.0	75.3	78.4	58.8	65.0	71.5
	POPE	88.6	54.8	67.7	78.9	85.0	56.2	68.8	77.4
	Δ	(-6.6)	1.9	(-0.7)	(-3.6)	(-6.6)	2.6	(-3.7)	(-5.9)
LLaVA-v1.3	THRONE	80.5	65.2	72.1	76.9	79.9	65.3	70.4	75.2
	POPE	85.5	61.6	71.6	79.3	87.9	61.7	72.6	80.8
	Δ	(-4.9)	3.6	0.5	(-2.4)	(-8.0)	3.6	(-2.2)	(-5.5)
LLaVA-v1.5	THRONE	68.1	61.0	64.4	66.6	69.9	56.4	62.2	66.8
	POPE	81.5	64.4	72.0	77.4	86.2	59.2	70.4	78.8
	Δ	(-13.4)	(-3.4)	(-7.6)	(-10.9)	(-16.2)	(-2.7)	(-8.2)	(-12.0)

Table 4.5: **Balanced Sampling (POPE) vs. Exhaustive Sampling (THRONE)**: Applying POPE sampling to THRONE leads to an underestimation of the prevalence of Type I hallucinations regardless of LVLm.

In the main paper, we demonstrated how the sampling method used in POPE [Yifan Li et al. 2023] leads to an underestimation of Type II hallucinations and outline a complete version of POPE (POPE-C), which shows the true extent of Type II hallucinations in LVLms. In this section, we perform the opposite—we apply POPE type sampling to THRONE and compare the results to the complete sampling method used in THRONE as outlined in the main paper. Table 4.5 shows the results of applying POPE style sampling to THRONE, once again, applying POPE style sampling leads to a large underestimation of Type I hallucinations. POPE style sampling applied to THRONE leads to a mean underestimation of $F^{0.5}_{CLS}$ by

10.9 points compared to complete sampling, which is the default in `THRONE`.

4.D CHAIR Overview

We present a detailed description of the CHAIR evaluation presented in [Rohrbach et al. 2018] below. The method of CHAIR, like THRONE, attempts to capture the extent of hallucinations in free-form generated text pertaining an image, however, focuses on more traditional image captioners. Similarly to THRONE, CHAIR does not use concept-focused prompts (i.e. nstead is focused on “Type-I”) and is intended to be used only in captioning tasks. CHAIR defines a manual pipeline on-top of the annotated MSCOCO image dataset to produce a list of ground truth objects present in the scene.

Given a set of ground truth objects in an image and a model-generated image caption, CHAIR extracts the objects present in the scene via a traditional hard-rule extraction method and then attempts to map each of the predicted objects into one of the 80 class categories of MSCOCO. The mapping of the extracted objects from the caption into the class set of MSCOCO uses a pre-defined synonym dictionary for each object category. Once the objects of the predicted caption are extracted and mapped to one of the MSCOCO categories, CHAIR evaluates for “false-positive” predictions (i.e. allucinations). In particular the authors introduce two variants of CHAIR, the first variant CHAIR_i quantifies the extent of hallucinated instances as,

$$\text{CHAIR}_i = \frac{|\{\text{hall. object}\}|}{|\{\text{pred. object}\}|}.$$

In this setting, hallucinations would be objects extracted from the model-generated captions that after being mapped, are still not present in the ground-truth object list for the corresponding instance. We note that CHAIR_i can be viewed as measuring the “false discovery rate” (FDR) that is $1 - P$ where P is the precision. Using Precision as the main metric, however is limiting as it does not take into account the “False Negative Rate” (FNR) which is $1 - R$ where R is the recall. This implies that by lacking recall measurements, CHAIR_i may assign high scores to short and incomplete captions which are not comprehensive in detailing the image. This is in stark contrast with the new generation of LVLM models powered by LLMs which are designed to be more exhaustive and detailed and makes the use of CHAIR_i problematic when evaluating with LVLMs. We note that CHAIR_i is

the main metric used when people report “CHAIR scores”, and typically reported numbers correspond to the mean CHAIR_i score across the MSCOCO validation set of images.

The second variation of CHAIR is the CHAIR_s which simply measure the number of sentences (predictions) that include at least 1 hallucination as compared with all sentences considered,

$$\text{CHAIR}_s = \frac{|\{\text{sentences w/ hall. object}\}|}{|\{\text{all sentences}\}|}. \quad (4.4)$$

Note that CHAIR_s does not measure the extent of hallucination within a sentence, just the existence of at least one hallucination. This is problematic as it does not capture the extent of hallucination in the sentence especially for long-form text and does not elucidate if a caption contains many or a single hallucination.

Producing ground truth in CHAIR To create a list of ground truth objects from MSCOCO annotations, the authors of CHAIR harness two annotation types to produce the most exhaustive list of ground truth objects. First the authors directly use all of the instance segmentation labels for each image, which they aggregate into a unique list of objects existing in the image. Next the authors use the 5 human-labelled captions of each image in MSCOCO and use the same extraction and mapping pipeline applied to the predictions to produce an additional set of objects that are present in the captions. Both objects lists are combined and the authors note that captioning ground truth and instance segmentation ground truth objects are often complementary as they follow different styles. Therefore combining objects from both types of object lists is beneficial for the most exhaustive final ground truth list.

4.E Qualitative Evaluation of THRONE

4.E.1 Evaluation Method

We include a self-contained file (`THRONE_qual_eval_results.html`) in the Supplementary Material, which shows the qualitative evaluation and comparison of THRONE and CHAIR. For each LVLM evaluated, we sample 10 COCO images at random in which THRONE and CHAIR *disagree* and 5 COCO images in which THRONE and CHAIR *agree*. Therefore, we qualitatively evaluate 165 responses. These results are summarized in Table 4.6

By calculating error rates for each of these cases and noting the proportion of responses in which THRONE and CHAIR *disagree* we can estimate the overall error rate of each method using a weighted sum.

$$\begin{aligned} \text{Method Error Rate} = & (\text{Agreement Proportion}) \\ & \times (\text{Error Rate in Agreement Case}) \\ & + (\text{Disagreement Proportion}) \\ & \times (\text{Error Rate in Disagreement Case}) \end{aligned}$$

$$\text{CHAIR Error Rate} = 0.697 \times 0.025 + 0.303 \times 0.233 = 0.088 = 8.8\%$$

$$\text{THRONE Error Rate} = 0.697 \times 0.025 + 0.303 \times 0.080 = 0.043 = 4.3\%$$

4.E.2 Discussion

We find that the plurality of errors made in THRONE relate to a mismatch between the LM definition of a certain class and the definition in COCO. The most clear example is in the `tv` COCO class. In COCO, this class includes computer monitors, whereas for an LM, the implication of the existence computer monitors in an LVLM response does not lead to a “yes” response when asked `Is there a tv in this image?` or similar. When doing an manual evaluation our human oracle *is* aware of the particular COCO class definitions and answers accordingly. Using a handcrafted rule for `tv` and other similar COCO classes, we would expect the error rate of THRONE to reduce significantly, but in THRONE we deliberately avoid the use of handcrafted rules.

Method	CHAIR == THRONE?	#Responses (%)	#Responses Evaluated	#Judgements	#Judgement Errors	Error Rate
CHAIR/THRONE	✓	38350 (69.7%)	55	157	4	2.5%
THRONE	✗	16650 (30.3%)	110	376	30	8.0%
CHAIR	✗	16650 (30.3%)	110	477	111	23.3%

Table 4.6: **Summary of Qualitative Evaluation:** Our qualitative results show that for responses in which THRONE and CHAIR differ, there is a large difference in the error rate. When THRONE and CHAIR agree, the error rate is small.

#Responses Analyzed	#False Positives Identified	#Hallucinations	#Misclassifications
90	71	69	2

Table 4.7: **Hallucinations Dominate False Positives:** Human evaluation establishes the vast majority ($\frac{69}{71} \approx 97\%$) of false positive object classes in LVLMM responses are true hallucinations rather than plausible misclassifications of objects.

As mentioned in the main paper, the errors in CHAIR are more fundamental and result due to simple text matching of synonyms not being able to discriminate between abstract concepts alluded to in a response and direct objects implied to exist in the image based on the response.

Table 4.7 shows results for human analysis of false positives. We analyze 90 responses, the 15 samples for mPLUG-Owl, MiniGPT-v2, MiniGPT-4, LLaVA-7b-v1.5, LLaVA-7b-v1.3 and InstructBLIP—the final 90 responses in the self-contained file: `THRONE_qual_eval_results.html` in the Supplementary Material.

4.F Improved Baseline Implementation Details

In the main paper, we introduced a simple method to augment the LLaVA visual instruction tuning data with an object enumeration task to reduce Type I and Type II hallucinations when used to train LLaVA models. The format used for object enumeration is:

Instruction: `<image> Give a list of objects and locations
in the image.`

Response: `{class_name_1} [{location_1}/absent]
...
{class_name_N} [{location_N}/absent]`

where `location_i` represents the location of the bounding box center point on a 3×3 grid.

We give additional details on the construction of the object enumeration task here.

4.F.1 Object Enumeration Implementation Details

The LLaVA visual instruction tuning data contains 157712 samples applied to 81479 images from the COCO training set (some images correspond to multiple samples). We ensure the absolute character length of our object enumerate task for a single sample is not exceedingly long—we do not want the visual instruction tuning data to be pushed outside the context length of the LLaVA model. This is done by limiting the number of instances *per class* in a sample to 3.

For each *sample* we construct an object enumeration task using bounding box data as follows: first, sort bounding box annotations for a given image by box area in descending order; second, loop over the sorted annotations adding the instance (`class_name_i`, `location_i`) to the object enumeration task if there are less than 3 instances of `class_name` in the task; third, sample 6 negative classes and append them to the object enumeration task using `absent` as the location string.

The sampling of negative classes is detailed next.

4.F.2 Negative Sampling Implementation Details

To sample negatives in the object enumeration task we first build a co-occurrence matrix from the bounding box annotations. The pseudocode for building this matrix is as follows:

```
from pycocotools.coco import COCO
import numpy as np
train_dset = COCO(instances_path)
num_cats = len(train_dset.getCatIds())
co_occur = np.array((num_cats, num_cats))
cat_id2cont_id = {x: i for i, x in sorted(enumerate(train_dset.getCatIds()))}
for iid in train_dset.getImgIds():
    anns = train_dset.loadAnns(train_dset.getAnnIds(imgIds=iid))
    pres_cats = [coco_cid2cont_cid[x['category_id']] for x in anns]
    pres_cats = np.unique(pres_cats)
    for r in pres_cats:
        for c in pres_cats:
            if r != c:
                co_occur[r, c] += 1
```

After building this co-occurrence matrix negative classes are sampled in a manner which is aware of the classes present in a given image. The pseudocode is as follows (using some variables from the above pseudocode):

```
present_cat_ids: List[int] # list of category ids present in the image
present_cont_ids = [cat_id2cont_id[x] for x in present_cat_ids

# combine co-occurrence across present categories
# ensuring present categories can not be sampled
present_co_occur = co_occur[present_cont_ids].copy()
present_co_occur[:, present_cont_ids] = 0
present_co_occur = present_co_occur / present_co_occur.sum(axis=1, keepdims=True)
present_co_occur = present_co_occur.sum(axis=0)
present_co_occur = present_co_occur / present_co_occur.sum()
# sharpen distribution
present_co_occur = present_co_occur ** 10
present_co_occur = present_co_occur / present_co_occur.sum()
```

```

rng = np.random.RandomState(iid)
neg_ids = rng.choice(
    sorted(train_dset.getCatIds()),
    size=6,
    p=present_co_occur,
    replace=False
)

```

This method of sampling yields negative classes which commonly co-occur with positive classes in a given image. Therefore, the object enumeration task trains the LVLM to distinguish individual objects and classes rather than relying on global context.

4.F.3 Object Enumeration Data Details

In Table 3 of the main paper, we present results on THRONE, POPE and POPE-C when training with our object enumeration task using COCO or COCO and VisualGenome as object enumeration data. Approximately 33000 of the 81479 COCO images in the LLaVA visual instruction tuning data are contained in the VisualGenome dataset. When using COCO *and* VisualGenome data, we construct the object enumeration task for an image from VisualGenome data when possible and COCO otherwise—we do not combine COCO and VisualGenome annotations for any image.

4.F.4 Inference Details

In Table 3 of the main paper, we present results on THRONE, POPE and POPE-C when training with our object enumeration task *and* performing the object enumeration task at inference. In the next section we show the effect of not performing object enumeration during inference on THRONE and POPE, instead directly addressing the relevant task.

4.F.5 Ablation Results

Model	Obj. Enum. Data	Obj. Enum. Negatives	Obj. Enum. Inference	THRONE			POPE		
				P_{CLS}	R_{CLS}	$F_{CLS}^{0.5}$	P	R	F^1
LLaVA-v1.3	$\times$	N/A	N/A	79.9	65.3	76.5	58.0	98.4	73.0
	COCO	$\times$	$\checkmark$	82.4	69.2	79.4	64.8	95.2	77.1
	COCO + VG	$\times$	$\checkmark$	85.8	60.4	79.1	66.0	95.3	78.0
	COCO	$\checkmark$	$\checkmark$	83.2	68.8	79.9	73.2	88.2	80.0
	COCO + VG	$\checkmark$	$\checkmark$	86.2	67.0	81.5	83.0	82.5	82.8
LLaVA-v1.5	$\times$	N/A	N/A	69.9	56.4	66.8	81.9	90.8	86.1
	COCO + VG	$\checkmark$	$\times$	79.3	76.1	78.6	83.2	86.4	84.8
	COCO + VG	$\checkmark$	$\checkmark$	86.1	77.0	84.1	89.8	83.7	86.7

Table 4.8: **Effect of Negatives and Inference:** Including negatives in our object enumeration task improves performance on THRONE and POPE in terms of precision and F-score. Performing the object enumeration task at inference time improves performance on THRONE and POPE, but hampers inference time as the object enumeration task can generate long sequences.

4.G Limitations

In this paper we present THRONE which is a step towards measuring and mitigating hallucinations in LVLMs, nonetheless, our work has few key limitations which we list below.

1. THRONE is concerned with *only* measuring hallucinations in LVLM predictions in the form of a *false existence of an object*. As observed in LLMs, hallucinations are much more multifaceted and include many additional abstract concepts such as wrong reasoning relating to a visual scene as well as wrong attributes of a particular objects or person. These additional hallucinations are not possible to be measured with THRONE without modifications.
2. The presented method of THRONE only focused on “*Type-I hallucinations*” which does not paint a complete picture of the hallucinating behavior of an LVLM. Indeed we present POPE-C in Fig. 7 to extend hallucination measurements in both Type-I and Type-II hallucinations. We present POPE-C as an extension of POPE since we observe that POPE severely undercounts hallucinations in Type-II form.
3. Due to lack of general and exhaustive ground truth object label lists for a given image, our method relies on curated datasets such as MCOCO or Object365 that have detailed annotations which are needed for our evaluation.
4. Our method focuses only on the hallucination bias of LVLMs but does not include measurements of other types of bias of LVLM generations (e.g. related to concepts of fairness in generation) which we leave for future work.

4.H Ethical Considerations

We present THRONE which is a general evaluation pipeline for measuring hallucinations (specifically “Type-I” hallucinations) in Large-Vision-Language Models (LVLMs). Overall we believe that our contribution is ethically positive as it measures and shows that existing public LVLMs are not yet ready to be deployed in mission critical applications, as we observe that they still suffer from hallucinating objects to a large extent. In addition we believe our presented evaluation framework also provides for a “north-star” in measuring evaluation and can aid the field

and practitioners alike in measuring and making progress towards reducing evaluations in LVLMs as well as electing to use one LVLM over another. We note that measuring societal bias in LVLMs is highly important pre-requisite before their deployment, however this is not investigated in the current work.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor (**only required where there isn't already a statement of contribution within the paper itself**).

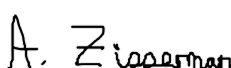
Title of Paper	THRONE: A Hallucination Benchmark for the Free-form Generations of Large Vision-Language Models
Publication Status	☐ Published ☐ Accepted for Publication ☒ Submitted for Publication ☐ Unpublished and unsubmitted work written in a manuscript style
Publication Details	Kaul, P., Li, Z., Yang, H., Dukler, Y., Swaminathan, A., Taylor, C. J., & Soatto, S. (2023). THRONE: A hallucination benchmark for the free-form generations of large vision-language models. [Manuscript submitted for publication].

Student Confirmation

Student Name:	Prannay Kaul		
Contribution to the Paper	First Author Contribution: • conception of research ideas • design and implementation of models • running of large-scale experiments • writing and presentation of the paper		
Signature 	Date	12th January 2024	

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Andrew Zisserman		
Supervisor comments The descriptions of the contributions is accurate.		
Signature 	Date	12th January 2024

This completed form should be included in the thesis, at the end of the relevant chapter.

Chapter 5

Conclusion

This chapter first details the contributions of the work in this thesis and outlines its impact (Section 5.1). Subsequently, future work for each research paper and on the overall topic of object perception as a step towards scene understanding is discussed (Section 5.2).

5.1 Achievements and Impact

Few-Shot Object Detection

In Chapter 2, we investigated prior few-shot object detection methods and identified a critical issue which limited their performance. Few-shot object detection data contains a large number of novel instances which are unlabelled—therefore they are treated as background during object detector training. The consequence of this (in terms of two-stage object detectors) is two-fold: (1) RPNs are unable to generate region proposals for novel instances in the first stage—as no suitable proposals are passed to the second stage successful detection of novel instances is impossible; and (2) the features of the second stage are heavily biased against classifying novel instances correctly and instead favour class predictions in the base categories. To address these issues, we proposed a novel pseudo-labelling approach which verifies the class labels of predicted novel instances, using a k -NN classifier, and corrects bounding box coordinates for novel instances. This process finds large numbers of unlabelled novel instances in the training data, which are used to further train the detector end-to-end. In terms of performance on the MSCOCO benchmark, this work remains comparable to more recent works in few-shot object

detection [S. Lin et al. 2023b; S. Lin et al. 2023a].

In terms of impact, the analysis of few-shot object detectors in this work has been used in open-vocabulary attribute recognition [Keyan Chen et al. 2023], semi-supervised object detection [Jiaming Li et al. 2023] and aerial image object detection [Lingjun Li et al. 2023]. The use of a k NN classifier in few-shot object detection has been built on in [Matcovici et al. 2023], but instead of using a k NN classifier as part of a pseudo-labelling process, the k NN classifier is built directly into a trained detector. Our pseudo-labelling approach to verify class labels and correct bounding boxes has been adapted to accurately annotate video data with human interactions [Z. Zhang et al. 2023] to provide verification instead of a k NN classifier. Finally, this work is covered in recent surveys on (few-shot) object detection [Köhler et al. 2021; Q. Huang et al. 2023; Jianzong Wu et al. 2023]. The open-sourced GitHub repository for this work has been starred 80 times, providing a useful reference for future work.

Open-Vocabulary Object Detection

In Chapter 3, we proposed multi-modal classifiers for open-vocabulary object detection. To define new object categories of interest, we developed: text-based classifiers—making use of an LLM to generate rich class descriptions automatically; vision-based classifiers—using a trained visual aggregator to combine visual information from a set of image exemplars; and multi-modal classifiers—the simple combination of text-based and vision-based classifiers. This work finds that utilising rich class descriptions improves novel category performance over simple classifiers which rely on class names alone. Furthermore, this work finds that multi-modal classifiers outperform both text-based and vision-based classifiers, demonstrating that each modality provides distinct information about novel categories.

This work is the first method to use both text-based and vision-based classifiers to define objects of interest in open-vocabulary object detection. Moreover, this work is the first to use multi-modal classifiers for open-vocabulary object detection. In terms of performance, our method which uses smaller visual backbones performed similarly with concurrent works utilising much larger backbones [Minderer et al. 2022; Kuo et al. 2022]. This work has inspired subsequent works in few-shot learn-

ing [Liao et al. 2023] and video recognition [Tongjia Chen et al. 2023]. Finally, this work is covered in recent surveys on open-vocabulary learning [Jianzong Wu et al. 2023; Chaoyang Zhu and Long Chen 2023]. The open-sourced GitHub repository for this work has been starred 65 times, providing a useful reference for future work.

Object Hallucinations in Large Vision-Language Models

In Chapter 4, we propose a new benchmark to quantitatively evaluate the prevalence of object hallucinations in the free-form responses of large vision-language model (LVLMs). We find many benchmarks which evaluate LVLMs require simple answers in a closed set e.g. yes-no [C. Fu et al. 2023] or multiple-choice [Y. Liu et al. 2023] questions, whereas the impressive ability of LVLMs is to generate long free-form text in response to user queries. Moreover, we find that hallucinations in free-form responses are disjoint from hallucinations in closed-set responses further motivating the need for our benchmark. For multiple LVLMs, we generate free-form image descriptions (longer than a caption) and use an ensemble of language models to establish the implied existence of objects in the image from the generated text, thus automatically enabling quantitative evaluation of object hallucinations, with respect to groundtruth image labels. To ensure access for all in the community, we make use of free and open-source language models to parse LVLM responses, instead of relying on closed-source APIs as is common in other benchmarks [Yu et al. 2023; Z. Sun et al. 2023; Bohao Li et al. 2023]. Additionally, we provide reasons for why a popular existing benchmark, POPE [Yifan Li et al. 2023], is incomplete in its evaluation of object hallucinations in closed-set responses and gives misleading results. We complete POPE and reveal the true prevalence of object hallucinations in closed-set responses for LVLMs.

This work is currently under review and we believe this paper will be useful to the vision-language community as a benchmark for evaluating object hallucinations in the free-form responses generated by LVLMs.

5.2 Future Work

We conclude by outlining promising future work for each research chapter in this thesis and then discuss future work on the overall topic of object perception as a

critical step towards scene understanding.

In Chapter 2, we proposed a novel pseudo-labelling approach for few-shot object detection. These pseudo-annotations are generated using a number of stages which each require separate hyperparameter tuning/training. Future work which combined each of these stages into a single model would be simpler, more efficient and potentially more effective. The “correct” stage of the pseudo-labelling process uses a modified Cascade R-CNN architecture [Z. Cai and Vasconcelos 2018] to output corrected bounding boxes. The “verify” stage uses a k NN classifier which takes an image patch as input and therefore requires a forward pass for *every candidate pseudo-annotation*—reducing efficiency. Adding a classification head to the modified Cascade R-CNN architecture would enable the “verify” and “correct” stages to be combined into a single model which would require a forward pass for *every image* rather than every candidate pseudo-annotation—this is analagous to efficiency gains in going from R-CNN [Girshick et al. 2014] to Fast R-CNN [Girshick 2015].

In Chapter 3, we proposed multi-modal classifiers for open-vocabulary object detection. The multi-modal classifiers we used were simply the sum of the text-based and vision-based classifiers. Future work would investigate the utility of more complex interactions between text-based and vision-based classifiers such as bilinear fusion or multi-modal aggregation using a Transformer.

Chapter 4 is a recent work which is currently under review, but we believe the general framework outlined in our benchmark can be extended to other hallucinations apart from object existence, such as object attributes. Moreover, emerging, small, open-source language models which demonstrate impressive performance can be integrated into our work potentially reducing the need for ensembling to improve robustness.

Unified Open-Vocabulary Object Perception

For comprehensive object perception in an image, a system must be able to detect/segment and classify any present object at various levels of granularity. This requires datasets beyond standard object detection datasets, which provide bounding box annotations for a fixed set of object categories. Datasets which provide various levels of annotation, from full instance masks to single points, such as

OpenImages [Krasin et al. 2017] may form part of the solution, but recent datasets such as SA-1B [Kirillov et al. 2023] which are partially manually labelled and partially automatically generated provide potential to scale annotation masks to the various granularities required for comprehensive object perception. ODinW [C. Li et al. 2022] combines 35 domain-specific object detection datasets to advance open-vocabulary object detection—focusing on the use of external language knowledge sources (WordNet [Miller 1995], Wikipedia and large language models) to aid computer vision models to adapt to various domains covered by the datasets. Moreover, recent models such as OpenSeeD [Hao Zhang et al. 2023] and Semantic-SAM [F. Li et al. 2023] provide steps towards unified object perception by combining multiple datasets and using specific prompting to detect and segment objects as desired by the user. Future work would investigate the use of wide ranging datasets and strong pretrained models to develop models capable of unified object perception, requiring simple prompts to curate outputs to the desired structure.

Scene Understanding from Vision-Language Models

Object perception is a critical first step towards comprehensive scene understanding. Inspired by the apparent reasoning capabilities of large vision-language models [H. Liu et al. 2023b] in tasks such as visual question answering, a promising direction for future work on comprehensive scene understanding is to convert a wide net of visual perception datasets into a structured representation compatible with large vision-language models, using special tokens to denote desired behaviour. Kosmos-2 [Z. Peng et al. 2023] is a recent work which uses a particular special token schema to train a large vision-language model to visually ground objects during text generation. Expanding this idea to enable LVLMs to produce relations between objects (with unique identifiers), identify object attributes/parts and include additional information such as scene location would be a promising step towards comprehensive scene understanding.

References

- Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. (2022). “Flamingo: a visual language model for few-shot learning”. In: *Advances in Neural Information Processing Systems*.
- B. Alexe, T. Deselaers, and V. Ferrari (2010). “What is an object?” In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh (2015). “Vqa: Visual question answering”. In: *Proceedings of the International Conference on Computer Vision*.
- Anas Awadalla, Irena Gao, Joshua Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Jenia Jitsev, Simon Kornblith, Pang Wei Koh, Gabriel Ilharco, Mitchell Wortsman, and Ludwig Schmidt (Mar. 2023). *OpenFlamingo*. Version v0.1.1. URL: <https://doi.org/10.5281/zenodo.7733589>.
- Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu (2023a). “Qwen Technical Report”. In: *arXiv preprint arXiv:2309.16609*.
- Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou (2023b). “Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond”. In: *arXiv preprint arXiv:2308.12966*.

- Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran (2018). “Zero-Shot Object Detection”. In: *Proceedings of the European Conference on Computer Vision*.
- Mina Bishay, Georgios Zoumpourlis, and Ioannis Patras (2019). “TARN: Temporal Attentive Relation Network for Few-Shot and Zero-Shot Action Recognition”. In: Yonatan Bitton, Hritik Bansal, Jack Hessel, Rulin Shao, Wanrong Zhu, Anas Awadalla, Josh Gardner, Rohan Taori, and Ludwig Schmidt (2023). *VisIT-Bench: A Benchmark for Vision-Language Instruction Following Inspired by Real-World Use*. arXiv: 2308.06595 [cs.CL].
- Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao (2020). “Yolov4: Optimal speed and accuracy of object detection”. In: *arXiv preprint arXiv:2004.10934*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. (2020). “Language models are few-shot learners”. In: *Advances in Neural Information Processing Systems*.
- Minwoo Byeon, Beomhee Park, Haecheon Kim, Sungjun Lee, Woonhyuk Baek, and Saehoon Kim (2022). *COYO-700M: Image-Text Pair Dataset*. <https://github.com/kakaobrain/coyo-dataset>.
- Zhaowei Cai and Nuno Vasconcelos (2018). “Cascade R-CNN: Delving Into High Quality Object Detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society.
- Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko (2020). “End-to-end object detection with transformers”. In: *Proceedings of the European Conference on Computer Vision*. Springer.
- Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin (2020). “Unsupervised learning of visual features by contrasting cluster assignments”. In: *arXiv preprint arXiv:2006.09882*.
- Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin (2021). “Emerging Properties in Self-Supervised Vision Transformers”. In: *arXiv preprint arXiv:2104.14294*.
- Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut (2021). “Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

- Ding-Jie Chen, He-Yen Hsieh, and Tyng-Luh Liu (2021). “Adaptive Image Transformer for One-Shot Object Detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Hao Chen, Yali Wang, Guoyou Wang, and Yu Qiao (2018). “LSTD: A Low-Shot Transfer Detector for Object Detection”. In: *Proceedings of the AAAI conference on artificial intelligence*.
- Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny (2023). “MiniGPT-v2: large language model as a unified interface for vision-language multi-task learning”. In: *arXiv preprint arXiv:2310.09478*.
- Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. (2019). “Hybrid task cascade for instance segmentation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Keyan Chen, Xiaolong Jiang, Yao Hu, Xu Tang, Yan Gao, Jianqi Chen, and Weidi Xie (2023). “OvarNet: Towards Open-Vocabulary Object Attribute Recognition”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Liangyu Chen, Tong Yang, Xiangyu Zhang, Wei Zhang, and Jian Sun (2021). “Points as queries: Weakly semi-supervised object detection by points”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton (2020). “A simple framework for contrastive learning of visual representations”. In: *Proceedings of the International Conference on Machine Learning*.
- Tongjia Chen, Hongshan Yu, Zhengeng Yang, Zechuan Li, Wei Sun, and Chen Chen (2023). “OST: Refining Text Knowledge with Optimal Spatio-Temporal Descriptor for General Video Recognition”. In: *arXiv preprint arXiv:2312.00096*.
- Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar (2022). “Masked-attention mask transformer for universal image segmentation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. (2023). *Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality*. URL: <https://lmsys.org/blog/2023-03-30-vicuna/>.

- François Chollet (2017). “Xception: Deep Learning with Depthwise Separable Convolutions”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Subhabrata Choudhury, Iro Laina, Christian Rupprecht, and Andrea Vedaldi (2021). “The Curious Layperson: Fine-Grained Image Recognition without Expert Labels”. In: *Proceedings of the British Machine Vision Conference*.
- Ondrej Chum, James Philbin, Josef Sivic, Michael Isard, and Andrew Zisserman (2007). “Total Recall: Automatic Query Expansion with a Generative Feature Model for Object Retrieval”. In: *Proceedings of the 11th International Conference on Computer Vision, Rio de Janeiro, Brazil*.
- Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei (2017). “Deformable convolutional networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi (2023). “InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning”. In: *Advances in Neural Information Processing Systems*.
- Xiyang Dai, Yinpeng Chen, Bin (Leo) Hsiao, Dongdong Chen, Mengchen Liu, Lu Yuan, and Lei Zhang (June 2021). “Dynamic Head: Unifying Object Detection Heads with Attentions”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- N. Dalal and B Triggs (2005). “Histogram of Oriented Gradients for Human Detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei (2009). “ImageNet: A Large-Scale Hierarchical Image Database”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Carl Doersch, Ankush Gupta, and Andrew Zisserman (2020). “CrossTransformers: spatially-aware few-shot transfer”. In: *Advances in Neural Information Processing Systems*.
- 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 (2021). “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *Proceedings of the International Conference on Learning Representations*.

- Isla Duporge, Olga Isupova, Steven Reece, David W Macdonald, and Tiejun Wang (2021). “Using very-high-resolution satellite imagery and deep learning to detect and count African elephants in heterogeneous landscapes”. In: *Remote Sensing in Ecology and Conservation*.
- Mohamed Elhoseiny, Yizhe Zhu, Han Zhang, and Ahmed Elgammal (2017). “Link the Head to the "Beak": Zero Shot Learning From Noisy Text Description at Part Precision”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Mark Everingham, Luc Van Gool, Chris K. I. Williams, John Winn, and Andrew Zisserman (2010). “The PASCAL Visual Object Classes (VOC) Challenge”. In: *International Journal of Computer Vision*.
- Hugging Face (2023). *Text Generation Inference*.
<https://github.com/huggingface/text-generation-inference>. Accessed: November 10, 2023.
- Qi Fan, Wei Zhuo, Chi-Keung Tang, and Yu-Wing Tai (2020). “Few-Shot Object Detection With Attention-RPN and Multi-Relation Detector”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Zhibo Fan, Yuchen Ma, Zeming Li, and Jian Sun (2021). “Generalized Few-Shot Object Detection without Forgetting”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, et al. (2015). “From captions to visual concepts and back”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao (2023). “EVA: Exploring the Limits of Masked Visual Representation Learning at Scale”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- P. Felzenszwalb, D. Mcallester, and D. Ramanan (2008). “A Discriminatively Trained, Multiscale, Deformable Part Model”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- P. F. Felzenszwalb, R. B. Grishick, D. McAllester, and D. Ramanan (2010). “Object Detection with Discriminatively Trained Part Based Models”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- P. F. Felzenszwalb and D. P. Huttenlocher (2004). “Efficient Graph-Based Image Segmentation”. In: *International Journal of Computer Vision*.

- Chengjian Feng, Yujie Zhong, Zequn Jie, Xiangxiang Chu, Haibing Ren, Xiaolin Wei, Weidi Xie, and Lin Ma (2022). “PromptDet: Towards Open-vocabulary Detection using Uncurated Images”. In: *Proceedings of the European Conference on Computer Vision*.
- Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Zhenyu Qiu, Wei Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, and Rongrong Ji (2023). “MME: A Comprehensive Evaluation Benchmark for Multimodal Large Language Models”. In: *arXiv preprint arXiv:2306.13394*.
- Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruva Ghosh, Jieyu Zhang, Eyal Orgad, Rahim Entezari, Giannis Daras, Sarah Pratt, Vivek Ramanujan, Yonatan Bitton, Kalyani Marathe, Stephen Mussmann, Richard Vencu, Mehdi Cherti, Ranjay Krishna, Pang Wei Koh, Olga Saukh, Alexander Ratner, Shuran Song, Hannaneh Hajishirzi, Ali Farhadi, Romain Beaumont, Sewoong Oh, Alex Dimakis, Jenia Jitsev, Yair Carmon, Vaishaal Shankar, and Ludwig Schmidt (2023). “DataComp: In search of the next generation of multimodal datasets”. In: *arXiv preprint arXiv:2304.14108*.
- Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, Hongsheng Li, and Yu Qiao (2023). “LLaMA-Adapter V2: Parameter-Efficient Visual Instruction Model”. In: *arXiv preprint arXiv:2304.15010*.
- R. B. Girshick (2015). “Fast R-CNN”. In: *Proceedings of the International Conference on Computer Vision*.
- R. B. Girshick, J. Donahue, T. Darrell, and J. Malik (2014). “Rich feature hierarchies for accurate object detection and semantic segmentation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui (2022). “Open-Vocabulary Detection via Vision and Language Knowledge Distillation”. In: *Proceedings of the International Conference on Learning Representations*.
- Agrim Gupta, Piotr Dollar, and Ross Girshick (2019). “LVIS: A dataset for large vocabulary instance segmentation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Guangxing Han, Yicheng He, Shiyuan Huang, Jiawei Ma, and Shih-Fu Chang (Oct. 2021). “Query Adaptive Few-Shot Object Detection With Heterogeneous Graph Convolutional Networks”. In: *Proceedings of the International Conference on Computer Vision*.

- K. He, X. Zhang, S. Ren, and J. Sun (2016). “Deep residual learning for image recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick (2020). “Momentum contrast for unsupervised visual representation learning”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick (2017). “Mask R-CNN”. In: *Proceedings of the International Conference on Computer Vision*.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). “Deep Residual Learning for Image Recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean (2015). “Distilling the Knowledge in a Neural Network”. In: *NeurIPS Deep Learning and Representation Learning Workshop*. URL: <http://arxiv.org/abs/1503.02531>.
- Jan Hosang, Rodrigo Benenson, Piotr Dollár, and Bernt Schiele (2015). “What makes for effective detection proposals?” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Ting-I Hsieh, Yi-Chen Lo, Hwann-Tzong Chen, and Tyng-Luh Liu (2019). “One-Shot Object Detection with Co-Attention and Co-Excitation”. In: *Advances in Neural Information Processing Systems*.
- Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen (2022). “LoRA: Low-Rank Adaptation of Large Language Models”. In: *Proceedings of the International Conference on Learning Representations*.
- Hanzhe Hu, Shuai Bai, Aoxue Li, Jinshi Cui, and Liwei Wang (2021). “Dense Relation Distillation with Context-aware Aggregation for Few-Shot Object Detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Qihan Huang, Haofei Zhang, Mengqi Xue, Jie Song, and Mingli Song (2023). “A survey of deep learning for low-shot object detection”. In: *ACM Computing Surveys*.
- Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Barun Patra, Qiang Liu, Kriti Aggarwal, Zewen Chi, Johan Bjorck, Vishrav Chaudhary, Subhojit Som, Xia Song, and Furu Wei (2023). “Language Is Not All You Need: Aligning Perception with Language Models”. In: *arXiv preprint arXiv:2302.14045*.

- Drew A Hudson and Christopher D Manning (2019). “Gqa: A new dataset for real-world visual reasoning and compositional question answering”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt (2021). *OpenCLIP*. URL: <https://doi.org/10.5281/zenodo.5143773>.
- Joel Janai, Fatma Güney, Aseem Behl, Andreas Geiger, et al. (2020). “Computer vision for autonomous vehicles: Problems, datasets and state of the art”. In: *Foundations and Trends® in Computer Graphics and Vision*.
- Jisoo Jeong, Seungeui Lee, Jeessoo Kim, and Nojun Kwak (2019). “Consistency-based Semi-supervised Learning for Object Detection”. In: *Advances in Neural Information Processing Systems*.
- Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig (2021). “Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision”. In: *Proceedings of the International Conference on Machine Learning*.
- Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie (2022). “Prompting visual-language models for efficient video understanding”. In: *Proceedings of the European Conference on Computer Vision*.
- Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan (2018). “Dvqa: Understanding data visualizations via question answering”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion (2021). “MDETR-modulated detection for end-to-end multi-modal understanding”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell (2019). “Few-shot object detection via feature reweighting”. In: *Proceedings of the International Conference on Computer Vision*.
- Leonid Karlinsky, Joseph Shtok, Sivan Harary, Eli Schwartz, Amit Aides, Rogerio Feris, Raja Giryes, and Alex M. Bronstein (2019). “RepMet: Representative-Based Metric Learning for Classification and Few-Shot Object Detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

- Prannay Kaul, Weidi Xie, and Andrew Zisserman (2022). “Label, Verify, Correct: A Simple Few-Shot Object Detection Method”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg (2014). “Referitgame: Referring to objects in photographs of natural scenes”. In: *Proceedings of the Conference on Empirical Methods in Natural Language*.
- Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi (2016). “A diagram is worth a dozen images”. In: *Proceedings of the European Conference on Computer Vision*.
- Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan (2020). “Supervised Contrastive Learning”. In: *Advances in Neural Information Processing Systems*.
- Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár (2019). “Panoptic segmentation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. (2023). “Segment anything”. In: *arXiv preprint arXiv:2304.02643*.
- G. Koch, R. Zemel, and R. Salakhutdinov (2015). “Siamese Neural Networks for One-shot Image Recognition”. In: *ICML 2015 Deep Learning Workshop (2015)*.
- Mona Köhler, Markus Eisenbach, and Horst-Michael Gross (2021). “Few-shot object detection: A comprehensive survey”. In: *arXiv preprint arXiv:2112.11699*.
- Philipp Krahenbuhl and Vladlen Koltun (2015). “Learning to propose objects”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Philipp Krähenbühl and Vladlen Koltun (2014). “Geodesic object proposals”. In: *Proceedings of the European Conference on Computer Vision*.
- Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Andreas Veit, Serge Belongie, Victor Gomes, Abhinav Gupta, Chen Sun, Gal Chechik, David Cai, Zheyun Feng, Dhyanesh Narayanan, and Kevin Murphy (2017). “OpenImages: A public dataset for large-scale multi-label and multi-class image classification.” In: *Dataset available from <https://github.com/openimages>*.
- Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. (2017). “Visual genome: Connecting language and vision using crowdsourced dense image annotations”. In: *International Journal of Computer Vision*.

- A. Krizhevsky, I. Sutskever, and G. E. Hinton (2012). “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems*.
- Weicheng Kuo, Yin Cui, Xiuye Gu, AJ Piergiovanni, and Anelia Angelova (2022). “F-VLM: Open-Vocabulary Object Detection upon Frozen Vision and Language Models”. In: *arXiv preprint arXiv:2209.15639*.
- Jet Propulsion Lab (2021). *CC-MAIN-2021-31-PDF-UNTRUNCATED*. URL: <https://downloads.digitalcorpora.org/corpora/files/CC-MAIN-2021-31-PDF-UNTRUNCATED/>.
- Hei Law and Jia Deng (2018). “CornerNet: Detecting objects as paired keypoints”. In: *Proceedings of the European Conference on Computer Vision*.
- Kibok Lee, Hao Yang, Satyaki Chakraborty, Zhaowei Cai, Gurumurthy Swaminathan, Avinash Ravichandran, and Onkar Dabeer (2022). “Rethinking few-shot object detection on a multi-domain benchmark”. In: *Proceedings of the European Conference on Computer Vision*.
- Aoxue Li and Zhenguo Li (June 2021). “Transformation Invariant Few-Shot Object Detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Jingkang Yang, and Ziwei Liu (2023). “Otter: A Multi-Modal Model with In-Context Instruction Tuning”. In: *arXiv preprint arXiv:2305.03726*.
- Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan (2023). “SEED-Bench: Benchmarking Multimodal LLMs with Generative Comprehension”. In: *arXiv preprint arXiv:2307.16125*.
- Bohao Li, Boyu Yang, Chang Liu, Feng Liu, Rongrong Ji, and Qixiang Ye (June 2021). “Beyond Max-Margin: Class Margin Equilibrium for Few-shot Object Detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Chunyuan Li, Haotian Liu, Liunian Li, Pengchuan Zhang, Jyoti Aneja, Jianwei Yang, Ping Jin, Houdong Hu, Zicheng Liu, Yong Jae Lee, et al. (2022). “Elevater: A benchmark and toolkit for evaluating language-augmented visual models”. In: *Advances in Neural Information Processing Systems*.
- Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang (2022). “Dn-detr: Accelerate detr training by introducing query denoising”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

- Feng Li, Hao Zhang, Peize Sun, Xueyan Zou, Shilong Liu, Jianwei Yang, Chunyuan Li, Lei Zhang, and Jianfeng Gao (2023). “Semantic-SAM: Segment and Recognize Anything at Any Granularity”. In: *arXiv preprint arXiv:2307.04767*.
- Jiaming Li, Xiangru Lin, Wei Zhang, Xiao Tan, Yingying Li, Junyu Han, Errui Ding, Jingdong Wang, and Guanbin Li (2023). “Gradient-based Sampling for Class Imbalanced Semi-supervised Object Detection”. In: *Proceedings of the International Conference on Computer Vision*.
- Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi (2023). “BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models”. In: *Proceedings of the International Conference on Machine Learning*.
- Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi (2022). “Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation”. In: *Proceedings of the International Conference on Machine Learning*. PMLR.
- Lingjun Li, Xiwen Yao, Xue Wang, Dongpao Hong, Gong Cheng, and Junwei Han (2023). “Robust few-shot aerial image object detection via unbiased proposals filtration”. In: *IEEE Transactions on Geoscience and Remote Sensing*.
- 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 (2022). “Grounded Language-Image Pre-training”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang (2019). “Scale-aware trident networks for object detection”. In: *Proceedings of the International Conference on Computer Vision*.
- Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen (2023). “Evaluating Object Hallucination in Large Vision-Language Models”. In: *Proceedings of the Conference on Empirical Methods in Natural Language*.
- Yiting Li, Haiyue Zhu, Yu Cheng, Wenxin Wang, Chek Sing Teo, Cheng Xiang, Prahlad Vadakkepat, and Tong Heng Lee (June 2021). “Few-Shot Object Detection via Classification Refinement and Distractor Retreatment”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Christopher Liao, Theodoros Tsiligkaridis, and Brian Kulis (2023). “Descriptor and Word Soups: Overcoming the Parameter Efficiency Accuracy Tradeoff for Out-of-Distribution Few-shot Learning”. In: *arXiv preprint arXiv:2311.13612*.

- Joseph J. Lim, C. Lawrence Zitnick, and Piotr Dollár (2013). “Sketch Tokens: A Learned Mid-level Representation for Contour and Object Detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Shaobo Lin, Kun Wang, Xingyu Zeng, and Rui Zhao (2023a). “An effective crop-paste pipeline for few-shot object detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Shaobo Lin, Kun Wang, Xingyu Zeng, and Rui Zhao (2023b). “Explore the Power of Synthetic Data on Few-Shot Object Detection”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.
- Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie (2017a). “Feature Pyramid Networks for Object Detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár (2017b). “Focal loss for dense object detection”. In: *Proceedings of the International Conference on Computer Vision*.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick (2014). “Microsoft coco: Common objects in context”. In: *Proceedings of the European Conference on Computer Vision*.
- Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez (2017). “A survey on deep learning in medical image analysis”. In: *Medical image analysis*.
- Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang (2023). “Aligning Large Multi-Modal Model with Robust Instruction Tuning”. In: *arXiv preprint arXiv:2306.14565*.
- Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee (2023a). “Improved baselines with visual instruction tuning”. In: *arXiv preprint arXiv:2310.03744*.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee (2023b). “Visual instruction tuning”. In: *Advances in Neural Information Processing Systems*.
- Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang (2022). “DAB-DETR: Dynamic Anchor Boxes are Better Queries for DETR”. In: *Proceedings of the International Conference on Learning Representations*.
- Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg (2016). “SSD: Single shot multibox

- detector”. In: *Proceedings of the European Conference on Computer Vision*. Springer.
- Yen-Cheng Liu, Chih-Yao Ma, Zijian He, Chia-Wen Kuo, Kan Chen, Peizhao Zhang, Bichen Wu, Zsolt Kira, and Peter Vajda (2021). “Unbiased Teacher for Semi-supervised Object Detection”. In: *Proceedings of the International Conference on Learning Representations*.
- Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin (2023). “MMBench: Is Your Multi-modal Model an All-around Player?” In: *arXiv preprint arXiv:2307.06281*.
- Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo (Oct. 2021). “Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows”. In: *Proceedings of the International Conference on Computer Vision*.
- J. Long, E. Shelhamer, and T. Darrell (2015). “Fully Convolutional Networks for Semantic Segmentation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, and Adam Roberts (2023). “The Flan Collection: Designing Data and Methods for Effective Instruction Tuning”. In: *Proceedings of the International Conference on Machine Learning*.
- Ilya Loshchilov and Frank Hutter (2018). “Decoupled Weight Decay Regularization”. In: *Proceedings of the International Conference on Learning Representations*.
- Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy (2016). “Generation and comprehension of unambiguous object descriptions”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi (2019). “OK-VQA: A Visual Question Answering Benchmark Requiring External Knowledge”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Ahmed Masry, Do Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque (2022). “ChartQA: A Benchmark for Question Answering about Charts with Visual and Logical Reasoning”. In: *Findings of the Association for Computational Linguistics: ACL 2022*. URL: <https://aclanthology.org/2022.findings-acl.177>.

- Stefan Matcovici, Daniel Voinea, and Alin-Ionut Popa (2023). “K-NN Embedded Space Conditioning for Enhanced Few-Shot Object Detection”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*.
- Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar (2021). “Docvqa: A dataset for vqa on document images”. In: *Proceedings of the IEEE/CVF winter conference on applications of computer vision*.
- Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang (2021). “Conditional detr for fast training convergence”. In: *Proceedings of the International Conference on Computer Vision*.
- Sachit Menon and Carl Vondrick (2023). “Visual Classification via Description from Large Language Models”. In: *Proceedings of the International Conference on Learning Representations*.
- George A. Miller (Nov. 1995). “WordNet: A Lexical Database for English”. In: *Communications of the Association for Computing Machinery*.
- Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, Xiao Wang, Xiaohua Zhai, Thomas Kipf, and Neil Houlsby (2022). “Simple Open-Vocabulary Object Detection”. In: *Proceedings of the European Conference on Computer Vision*.
- Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty (2019). “OCR-VQA: Visual Question Answering by Reading Text in Images”. In: *2019 International Conference on Document Analysis and Recognition (ICDAR)*.
- Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant (2014). “The CoNLL-2014 Shared Task on Grammatical Error Correction”. In: *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*. URL: <https://aclanthology.org/W14-1701>.
- OpenAI (2023a). *GPT-4 Technical Report*. arXiv: [2303.08774](https://arxiv.org/abs/2303.08774) [cs.CL].
- OpenAI (2023b). *Introducing ChatGPT*. URL: <https://openai.com/blog/chatgpt/>.
- Vicente Ordonez, Girish Kulkarni, and Tamara Berg (2011). “Im2Text: Describing Images Using 1 Million Captioned Photographs”. In: *Advances in Neural Information Processing Systems*. Ed. by J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger. URL: <https://proceedings.neurips.cc/paper/2011/file/5dd9db5e033da9c6fb5ba83c7a7e9-Paper.pdf>.

- Anton Osokin, Denis Sumin, and Vasily Lomakin (2020). “OS2D: One-Stage One-Shot Object Detection by Matching Anchor Features”. In: *Proceedings of the European Conference on Computer Vision*.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. (2022). “Training language models to follow instructions with human feedback”. In: *Advances in Neural Information Processing Systems*.
- Jeffrey Ouyang-Zhang, Jang Hyun Cho, Xingyi Zhou, and Philipp Krähenbühl (2022). “NMS Strikes Back”. In: *arXiv preprint arXiv:2212.06137*.
- C. Papageorgiou and T. Poggio (2000). “A Trainable System for Object Detection”. In: *International Journal of Computer Vision*.
- Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran (2018). “Image transformer”. In: *Proceedings of the International Conference on Machine Learning*.
- Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao (2023). “Instruction tuning with gpt-4”. In: *arXiv preprint arXiv:2304.03277*.
- Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei (2023). “Kosmos-2: Grounding Multimodal Large Language Models to the World”. In: *arXiv preprint arXiv:2306.14824*.
- Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik (2015). “Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models”. In: *Proceedings of the International Conference on Computer Vision*.
- Sarah Pratt, Rosanne Liu, and Ali Farhadi (2022). “What does a platypus look like? Generating customized prompts for zero-shot image classification”. In: *arXiv preprint arXiv:2209.03320*.
- Tina D Purnat, Paolo Vacca, Christine Czerniak, Sarah Ball, Stefano Burzo, Tim Zecchin, Amy Wright, Supriya Bezbaruah, Faizza Tanggol, Ève Dubé, Fabienne Labbé, Maude Dionne, Jaya Lamichhane, Avichal Mahajan, Sylvie Briand, and Tim Nguyen (July 2021). “Infodemic Signal Detection During the COVID-19 Pandemic: Development of a Methodology for Identifying Potential Information Voids in Online Conversations”. In: *JMIR Infodemiology*.
- Limeng Qiao, Yuxuan Zhao, Zhiyuan Li, Xi Qiu, Jianan Wu, and Chi Zhang (Oct. 2021). “DeFRCN: Decoupled Faster R-CNN for Few-Shot Object Detection”. In: *Proceedings of the International Conference on Computer Vision*.

- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. (2021). “Learning transferable visual models from natural language supervision”. In: *Proceedings of the International Conference on Machine Learning*. PMLR.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever (2019). “Language models are unsupervised multitask learners”. In: *OpenAI Blog*.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu (2020). “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer”. In: *Journal of Machine Learning Research*. URL: <http://jmlr.org/papers/v21/20-074.html>.
- Joseph Redmon, Santosh Divvala, Ross Girshick, and A. Farhadi (2016). “You Only Look Once: Unified, Real-Time Object Detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Joseph Redmon and Ali Farhadi (2017). “YOLO9000: Better, Faster, Stronger”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Joseph Redmon and Ali Farhadi (2018). “YOLOv3: An Incremental Improvement”. In: *arXiv preprint arXiv:1804.02767*.
- S. Ren, K. He, R. Girshick, and J. Sun (2016). “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. In: *Advances in Neural Information Processing Systems*.
- Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik (2021). “ImageNet-21K Pretraining for the Masses”. In: *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*.
- Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko (2018). “Object Hallucination in Image Captioning”. In: *Proceedings of the Conference on Empirical Methods in Natural Language*.
- Olaf Ronneberger, Philipp Fischer, and Thomas Brox (2015). “U-net: Convolutional networks for biomedical image segmentation”. In: *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention*. Springer.
- Chuck Rosenberg, Martial Hebert, and Henry Schneiderman (2005). “Semi-supervised Self-training of Object Detection Models”. In: *Winter Conference on Applications of Computer Vision*.
- Giuliano Rossi, Jakub Kolodziej, and Gurvinder Brar (Dec. 2020). “A recommender system for active stock selection”. In: *Computational Management Science*.

- H. Rowley, S. Baluja, and T. Kanade (1998). “Neural Network-Based Face Detection”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. (2022). “Laion-5b: An open large-scale dataset for training next generation image-text models”. In: *Advances in Neural Information Processing Systems*.
- Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki (2021). “Laion-400m: Open dataset of clip-filtered 400 million image-text pairs”. In: *arXiv preprint arXiv:2111.02114*.
- Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi (2022). “A-OKVQA: A Benchmark for Visual Question Answering using World Knowledge”. In: *arXiv preprint arXiv:2206.01718*.
- H. Scudder (1965). “Probability of error of some adaptive pattern-recognition machines”. In: *IEEE Transactions on Information Theory*.
- Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun (2019). “Objects365: A large-scale, high-quality dataset for object detection”. In: *Proceedings of the International Conference on Computer Vision*.
- Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut (2018). “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)*.
- Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick (2016). “Training region-based object detectors with online hard example mining”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh (2020). “TextCaps: a Dataset for Image Captioning with Reading Comprehension”. In: *Proceedings of the European Conference on Computer Vision*.
- Karen Simonyan and Andrew Zisserman (2015). “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *International Conference on Learning Representations*.
- Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Devi Parikh, and Marcus Rohrbach (2019). “Towards VQA Models That Can Read”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

- Linda B. Smith and Lauren K. Slone (2017). “A Developmental Approach to Machine Learning?” In: *Frontiers in Psychology*. URL:
<https://www.frontiersin.org/article/10.3389/fpsyg.2017.02124>.
- Jake Snell, Kevin Swersky, and Richard Zemel (2017). “Prototypical networks for few-shot learning”. In: *Advances in Neural Information Processing Systems*.
- Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister (2020). “A Simple Semi-supervised Learning Framework for Object Detection”. In: *arXiv preprint arXiv:2005.04757*.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov (2014). “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research*.
- Bo Sun, Banghuai Li, Shengcai Cai, Ye Yuan, and Chi Zhang (June 2021). “FSCE: Few-Shot Object Detection via Contrastive Proposal Encoding”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liang-Yan Gui, Yu-Xiong Wang, Yiming Yang, et al. (2023). “Aligning large multimodal models with factually augmented rlhf”. In: *arXiv preprint arXiv:2309.14525*.
- Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales (2018). “Learning to Compare: Relation Network for Few-Shot Learning”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Mingxing Tan, Ruoming Pang, and Quoc V Le (2020). “EfficientDet: Scalable and efficient object detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto (2023). *Stanford Alpaca: An Instruction-following LLaMA model*.
https://github.com/tatsu-lab/stanford_alpaca.
- The MosaicML NLP Team (2023). *Introducing MPT-7B: A New Standard for Open-Source, Commercially Usable LLMs*. URL:
<https://www.mosaicml.com/blog/mpt-7b>.
- Zhi Tian, Chunhua Shen, Hao Chen, and Tong He (2019). “FCOS: Fully Convolutional One-stage Object Detection”. In: *Proceedings of the International Conference on Computer Vision*.

- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. (2023a). “Llama: Open and efficient foundation language models”. In: *arXiv preprint arXiv:2302.13971*.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom (2023b). “Llama 2: Open Foundation and Fine-Tuned Chat Models”. In: *arXiv preprint arXiv:2307.09288*.
- Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, S. M. Ali Eslami, Oriol Vinyals, and Felix Hill (2021). “Multimodal Few-Shot Learning with Frozen Language Models”. In: *Advances in Neural Information Processing Systems*.
- Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders (2013). “Selective search for object recognition”. In: *Proceedings of the International Conference on Computer Vision*.
- Ultralytics (2023). *YOLOv8*. URL: <https://github.com/ultralytics/ultralytics>.
- Aäron van den Oord, Yazhe Li, and Oriol Vinyals (2018). “Representation Learning with Contrastive Predictive Coding”. In: *arXiv preprint arXiv:1807.03748*.
- K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek (2010). “Evaluating Color Descriptors for Object and Scene Recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin (2017). “Attention is All you Need”. In: *NIPS*.

- Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra (2016). “Matching Networks for One Shot Learning”. In: *Advances in Neural Information Processing Systems*.
- Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan (2015). “Show and tell: A neural image caption generator”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- P. Viola and M. Jones (2001). “Rapid Object Detection using a Boosted Cascade of Simple Features”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Junyang Wang, Yiyang Zhou, Guohai Xu, Pengcheng Shi, Chenlin Zhao, Haiyang Xu, Qinghao Ye, Ming Yan, Ji Zhang, Jihua Zhu, Jitao Sang, and Haoyu Tang (2023). “Evaluation and Analysis of Hallucination in Large Vision-Language Models”. In: *arXiv preprint arXiv:2308.15126*.
- Xin Wang, Thomas E. Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu (2020). “Frustratingly Simple Few-Shot Object Detection”. In: *Proceedings of the International Conference on Machine Learning*.
- Yu-Xiong Wang, Deva Ramanan, and Martial Hebert (2019). “Meta-learning to detect rare objects”. In: *Proceedings of the International Conference on Computer Vision*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. (2022). “Chain-of-thought prompting elicits reasoning in large language models”. In: *Advances in Neural Information Processing Systems*.
- Aming Wu, Yahong Han, Linchao Zhu, and Yi Yang (Oct. 2021). “Universal-Prototype Enhancing for Few-Shot Object Detection”. In: *Proceedings of the International Conference on Computer Vision*.
- Jianzong Wu, Xiangtai Li, Shilin Xu Haobo Yuan, Henghui Ding, Yibo Yang, Xia Li, Jiangning Zhang, Yunhai Tong, Xudong Jiang, Bernard Ghanem, et al. (2023). “Towards open vocabulary learning: A survey”. In: *arXiv preprint arXiv:2306.15880*.
- Jiaxi Wu, Songtao Liu, Di Huang, and Yunhong Wang (2020). “Multi-Scale Positive Sample Refinement for Few-Shot Object Detection”. In: *Proceedings of the European Conference on Computer Vision*.
- Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick (2019). *Detectron2*. <https://github.com/facebookresearch/detectron2>.

- Yang Xiao and Renaud Marlet (2020). “Few-Shot Object Detection and Viewpoint Estimation for Objects in the Wild”. In: *Proceedings of the European Conference on Computer Vision*.
- Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le (2020). “Self-training with noisy student improves imagenet classification”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley (2023). “Baize: An Open-Source Chat Model with Parameter-Efficient Tuning on Self-Chat Data”. In: *arXiv preprint arXiv:2304.01196*.
- Mengde Xu, Zheng Zhang, Han Hu, Jianfeng Wang, Lijuan Wang, Fangyun Wei, Xiang Bai, and Zicheng Liu (2021). “End-to-End Semi-Supervised Object Detection with Soft Teacher”. In: arXiv: [2106.09018 \[cs.CV\]](#).
- Xiaopeng Yan, Ziliang Chen, Anni Xu, Xiaoxi Wang, Xiaodan Liang, and Liang Lin (Oct. 2019). “Meta R-CNN: Towards General Solver for Instance-Level Low-Shot Learning”. In: *Proceedings of the International Conference on Computer Vision*.
- Ze Yang, Yali Wang, Xianyu Chen, Jianzhuang Liu, and Yu Qiao (2020). “Context-Transformer: Tackling Object Confusion for Few-Shot Detection”. In: *Proceedings of the AAAI conference on artificial intelligence*.
- Hongbin Ye, Tong Liu, Aijia Zhang, Wei Hua, and Weiqiang Jia (2023). “Cognitive mirage: A review of hallucinations in large language models”. In: *arXiv preprint arXiv:2309.06794*.
- Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. (2023). “mplug-owl: Modularization empowers large language models with multimodality”. In: *arXiv preprint arXiv:2304.14178*.
- Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang (2023). “MM-Vet: Evaluating Large Multimodal Models for Integrated Capabilities”. In: *arXiv preprint arXiv:2308.02490*.
- Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy (2022). “Open-Vocabulary DETR with Conditional Matching”. In:
- Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang (2021). “Open-Vocabulary Object Detection Using Captions”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer (2022). “Scaling Vision Transformers”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

- Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M. Ni, and Heung-Yeung Shum (2022). “DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection”. In: *Proceedings of the International Conference on Learning Representations*.
- Hao Zhang, Feng Li, Xueyan Zou, Shilong Liu, Chunyuan Li, Jianwei Yang, and Lei Zhang (2023). “A simple framework for open-vocabulary segmentation and detection”. In: *Proceedings of the International Conference on Computer Vision*.
- Hongguang Zhang, Li Zhang, Xiaojuan Qi, Hongdong Li, Philip HS Torr, and Piotr Koniusz (2020). “Few-shot action recognition with permutation-invariant attention”. In: *Proceedings of the European Conference on Computer Vision*.
- Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. (2023). “Siren’s Song in the AI Ocean: A Survey on Hallucination in Large Language Models”. In: *arXiv preprint arXiv:2309.01219*.
- Zheng Zhang, Zheng Ning, Chenliang Xu, Yapeng Tian, and Toby Jia-Jun Li (2023). “PEANUT: A Human-AI Collaborative Tool for Annotating Audio-Visual Data”. In: *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*.
- Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. (2023). “A survey of large language models”. In: *arXiv preprint arXiv:2303.18223*.
- Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luwei Zhou, Xiyang Dai, Lu Yuan, Yin Li, and Jianfeng Gao (2022). “RegionCLIP: Region-Based Language-Image Pretraining”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra (2022). “Detecting twenty-thousand classes using image-level supervision”. In: *Proceedings of the European Conference on Computer Vision*.
- Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl (2021). “Probabilistic two-stage detection”. In: *arXiv preprint arXiv:2103.07461*.
- Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl (2019). “Objects as points”. In: *arXiv preprint arXiv:1904.07850*.
- Yiyang Zhou, Chenhang Cui, Jaehong Yoon, Linjun Zhang, Zhun Deng, Chelsea Finn, Mohit Bansal, and Huaxiu Yao (2023). *Analyzing and Mitigating Object Hallucination in Large Vision-Language Models*. arXiv: [2310.00754 \[cs.LG\]](#).

- Chaoyang Zhu and Long Chen (2023). “A survey on open-vocabulary detection and segmentation: Past, present, and future”. In: *arXiv preprint arXiv:2307.09220*.
- Chenchen Zhu, Fangyi Chen, Uzair Ahmed, Zhiqiang Shen, and Marios Savvides (June 2021). “Semantic Relation Reasoning for Shot-Stable Few-Shot Object Detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny (2023). “Minigpt-4: Enhancing vision-language understanding with advanced large language models”. In: *arXiv preprint arXiv:2304.10592*.
- Wanrong Zhu, Jack Hessel, Anas Awadalla, Samir Yitzhak Gadre, Jesse Dodge, Alex Fang, Youngjae Yu, Ludwig Schmidt, William Yang Wang, and Yejin Choi (2023). “Multimodal C4: An Open, Billion-scale Corpus of Images Interleaved With Text”. In: *arXiv preprint arXiv:2304.06939*.
- Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai (2020). “Deformable detr: Deformable transformers for end-to-end object detection”. In: *Proceedings of the International Conference on Learning Representations*.
- C. L. Zitnick and P. Dollar (2014). “Edge Boxes: Locating Object Proposals from Edges”. In: *Proceedings of the European Conference on Computer Vision*.
- Zhuofan Zong, Guanglu Song, and Yu Liu (2023). “Detrs with collaborative hybrid assignments training”. In: *Proceedings of the International Conference on Computer Vision*.
- Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le (2020). “Rethinking Pre-training and Self-training”. In: *Advances in Neural Information Processing Systems*.