F-VLM: Open-Vocabulary Object Detection upon Frozen Vision and Language Models

Weicheng Kuo
Google Research, Brain Team
weicheng@google.com

Yin Cui
Google Research, Perception
yincui@google.com

Xiuye Gu
Google Research, Perception
xiuyegu@google.com

AJ Piergiovanni
Google Research, Brain Team
ajpiergi@google.com

Anelia Angelova
Google Research, Brain Team
anelia@google.com

Abstract

We present **F-VLM**, a simple open-vocabulary object detection method built upon **F**rozen **V**ision and **L**anguage **M**odels. F-VLM simplifies the current multi-stage training pipeline by eliminating the need for knowledge distillation or detection-tailored pretraining. Surprisingly, we observe that a *frozen* VLM: 1) retains the locality-sensitive features necessary for downstream detection, and 2) is a strong object classifier. We finetune only the detector head and combine the detector and VLM outputs for each region at inference time. Our recipe shows compelling scaling behavior and achieves the state of the art on LVIS open-vocabulary detection benchmark, outperforming the leading approach by 6.5 mask AP on the novel categories. Finally, we demonstrate very competitive results on COCO open-vocabulary detection benchmark and cross-dataset transfer, in addition to up to 200× training compute savings compared to prior works. Code will be released.

1 Introduction

Detection is a fundamental vision task that requires the algorithm to localize and recognize objects in an image. The task is intuitively appealing and useful for many product applications, but unfortunately requires manual annotation of bounding boxes or masks. This data collection process is tedious and costly, which limits the modern detection vocabulary size to less than an order of 10^3. This is orders of magnitude smaller than the vocabulary humans use to describe the visual world.

To overcome such limitation, Open-vocabulary Object Detection was proposed [9, 12, 44, 49]. The task aims to take detection beyond its limited training vocabulary by leveraging other sources of supervision such as image captions, or vision and language pretraining. Due to the need for region-level generalization, existing methods typically involve knowledge distillation [9, 12], region distillation on external

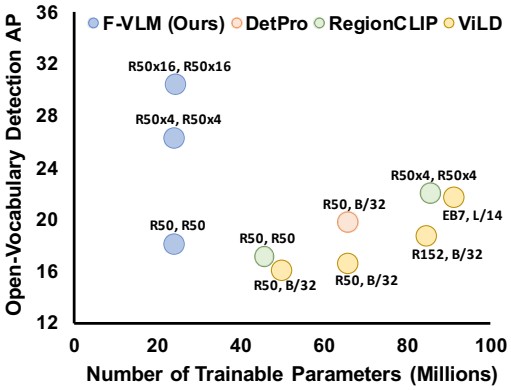


Figure 1: F-VLM is a novel open-vocabulary detection approach that outperforms existing works with much fewer trainable parameters. All methods use pretrained CLIP [30] for open-vocabulary detection, and we show (detector backbone, pretrained CLIP model) for each method. We report mask AP of novel categories on LVIS [13].

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

data [49], or pre-training with image-level captions [44], in addition to the standard detection training. Most methods depend on pre-trained vision and language models (VLMs) [9, 12, 49, 51] for generalization, and require training the whole detector from scratch [12, 51], or need a separate pretraining and finetuning process [9, 49, 42, 44].

Vision and language models (VLMs) acquire strong open-vocabulary recognition ability by learning from Internet-scale image-text pairs [19, 29, 30]. They are typically applied for zero-shot recognition (*e.g.*, on ImageNet) using frozen weights without finetuning, which stands in contrast to the existing practices to re-train or pre-train and finetune when applying VLMs for open-vocabulary detection. This causes us to ask, "*Can we build an open-vocabulary object detector directly upon frozen VLMs?*"

Object detection entails recognition and localization of objects across various scales. Although VLMs can perform open-vocabulary recognition at object level [9, 12, 49], the localization capability of VLMs in complex scenes such as LVIS [13] have been under-explored, especially when the weights are frozen. It remains unclear how to build an open-vocabulary detector with strong localization capability upon frozen VLMs, which are typically pre-trained at the image-level rather than the object-level. From the perspective of training efficiency, it is important to understand these properties about frozen VLMs, because the trend of building larger VLMs [19, 29, 30, 46] have made re-training, pre-training, and finetuning for detection more and more costly.

We propose F-VLM – a simple and scalable open-vocabulary detection approach built upon frozen VLMs (see Figure 1). For localization, we conjecture that locality sensitive information remains largely present and can be extracted by adding a lightweight detector head. For recognition, we apply the frozen VLMs at test time on the detected region features. Specifically, we train only the detector head upon a frozen VLM backbone, and combine the detection scores with the corresponding VLM predictions at test time. Our recipe reduces the training complexity of an open-vocabulary detector to below that of a standard detector, obviating the need for knowledge distillation or detection-tailored pretraining. By preserving the knowledge of pretrained VLMs completely, F-VLM maintains a similar philosophy as ViTDet [24] to decouple the detector-specific learning from the more general knowledge in the backbone.

Despite its simplicity, F-VLM shows compelling scaling behavior and achieves the state of the art performance on LVIS open-vocabulary detection benchmark, outperforming the leading approach by 6.5 mask AP_r. In addition, we show very competitive results on COCO open-vocabulary benchmark, strong cross-dataset transfer detection, and consistent improvements by increasing the backbone capacity. Last but not least, F-VLM provides up to 200$\times$ training compute savings without detection-tailored pre-training. We hope these observations will help the community explore frozen VLMs for a broader range of open-vocabulary applications. A summary of our contributions are listed below.

- We propose F-VLM – a simple open-vocabulary object detection method built upon frozen VLMs without knowledge distillation or detection-tailored pretraining.
- F-VLM has much fewer trainable parameters, which allows it to train significantly faster than prior works with a 200$\times$ reduction in computational resources.
- F-VLM surpasses the state of the art on LVIS open-vocabulary detection benchmark by 6.5 AP_r and outperforms existing approaches in cross-dataset transfer (COCO, Objects365).

2 Related Work

Zero-shot/Open-vocabulary visual recognition and representation learning. Zero-shot and open-vocabulary recognition has been a long-standing problem in computer vision. Earlier works use the visual attributes to represent categories as binary codebooks and learn to predict the attributes for novel categories [18, 34]. DeViSE [10] and ConSE [28] pioneer to learn a joint image-text embedding space using deep learning. Many works have shown the promise of representation learning from natural language associated with images, such as image tags [4, 8, 20] or text descriptions [7, 17, 35, 40, 48]. Recently, popular large VLMs scale up by training on billions of image-text pairs and acquire strong image-text representation by contrastive learning [30, 19, 29, 46]. These models achieve impressive zero-shot performance on a suite of classification benchmarks with outstanding robustness, and show great benefits in scaling model capacity. While all the above works study image-level recognition, we focus on the object-level understanding and explore how to use

frozen VLMs for open-vocabulary detection. From the perspective of frozen pretrained models, frozen classification models are beneficial for standard detection with adequate detector head capacity [39], and frozen language models are effective for multi-modal few-shot learning [38]. Similarly, we study whether frozen VLMs are effective for open-vocabulary object detection.

Zero-Shot/Open-vocabulary object detection. It is costly and labor-intensive to scale up data collection and annotation for large vocabulary detection. Zero-shot detection aims to alleviate the challenge by learning to detect novel categories not present in the training data. Many methods address this by aligning the image region features to category word embeddings [1, 31, 5, 47], or synthesizing visual features with a generative model [14, 52]. Recently, Zareian *et al*. proposes the open-vocabulary detection (OVD) benchmark with a view to bridge the performance gap between ZSD and supervised learning [44]. The model was first pretrained on image-caption data to recognize novel objects, and then finetuned for zero-shot detection [44].

Following the OVD benchmark, ViLD [12] proposes to distill the rich representation of pretrained VLM into the detector, and DetPro [9] improves upon ViLD by applying the idea of prompt optimization. RegionCLIP [49] develops a region-text pre-training strategy that leverages pretrained VLMs and image-caption data, while Detic [51] jointly trains a detector with image-level supervision. GLIP [23] formulates object detection as a phrase grounding task and pretrains on a wide variety of detection, grounding, and caption datasets for zero/few-shot object detection. Similarly, OWL-ViT [27] (arxiv) proposes to finetune pretrained vision transformers on a suite of detection/grounding datasets. All mentioned methods require training the entire detector from scratch, finetuning after detection-tailored pretraining, or training on a suite of detection/grounding datasets. In contrast, we propose to train only a detector head upon a frozen VLM without using any of the above.

3 Method

3.1 Overview

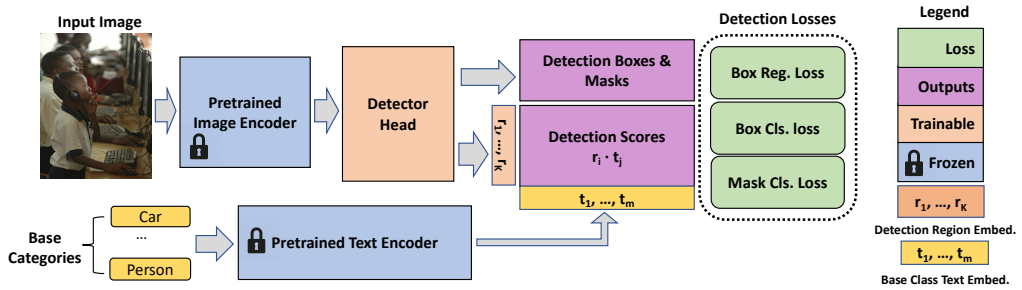
In this paper we address the problem of open-vocabulary object detection. At training time, the model has access to the detection labels of C_B base categories, but needs to detect objects from a set of C_N novel categories at test time. To make the settings more practical [44], we follow previous works and assume the availability of a pre-trained vision and language model (VLM) which has learnt from plenty of image-text pairs on the internet [12, 49].

Figure 2 shows the overall F-VLM architecture. We propose to build the open-vocabulary object detector upon frozen VLMs by training only the detector head upon frozen features, which guarantees to completely preserve the open-vocabulary classification ability of pre-trained VLMs. At test time, we combine the detector scores with the VLM scores by geometric means to obtain open-vocabulary object detection scores. By going directly to frozen pre-trained models, our approach is simple, competitive, and easily scalable.

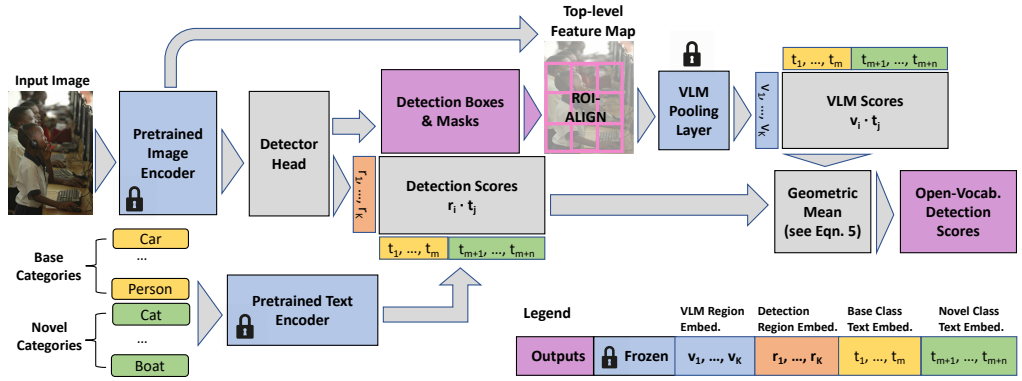
3.2 Pretraining from Vision and Language Models

Recently, Vision and Language Models (VLM) are popular because of their rich knowledge and strong representation for both visual and linguistic domains. We desire to retain their knowledge as much as possible, and avoid the effort and cost to finetune or train these VLMs from scratch. Following existing open-vocabulary detection works [9, 12, 49], we focus on contrastively pre-trained VLMs in this paper *e.g*. [19, 30]. Contrastive VLMs typically have the image and text encoders trained jointly with a contrastive objective. We use the image encoder for pre-training, and the text encoder for caching the text embeddings of detection dataset vocabulary offline (see Sec. 3.3).

We consider the VLM image encoder in two parts: 1) the feature extractor $\mathcal{F}(\cdot)$ *e.g*. ResNet-50 [30], and 2) the last feature pooling layer $\mathcal{P}(\cdot)$ *e.g*. an attention layer [30]. We adopt the same backbone architecture as the image feature extractor $\mathcal{F}(\cdot)$, which allows us to directly initialize the weights and inherit the rich semantic knowledge (see Fig. 2a). Along with the backbone initialization, we also adopt the same image normalization scheme as the VLM pretraining to maintain its open-vocabulary recognition ability. We use the last VLM pooling layer $\mathcal{P}(\cdot)$ for score combination at test time only ($\mathcal{P}(\cdot)$ unused at training time). Building upon the frozen backbone features, we adopt Faster R-CNN [33] head including the feature pyramid network [25] as the detector head following previous works [9, 12, 49]. The detector head is randomly initialized and is *the only trainable component* of



(a) **F-VLM training architecture**. At training time, F-VLM is a standard detector with the last classification layer replaced by the text embeddings from base categories. We only train the detector head and freeze the rest of the model.



(b) **F-VLM inference architecture**. At test time, F-VLM uses the detection boxes to crop out the top-level features of frozen VLM backbone and compute the VLM scores for each region. The trained detector head provides the localization, while the classification is a combination of detection and VLM scores.

Figure 2: **F-VLM architecture.** We present both training and inference time architectures of F-VLM, where the VLM pooling layer and detection/VLM score combination are the main differences.

F-VLM. Despite the image-level pretraining, we found empirically that the frozen VLM backbone contains adequate locality sensitive features to enable accurate downstream detection.

3.3 Text-Embedding Region Classifier

Notations: Let's define I as the input image, $\mathcal{F}(I)$ the backbone features from the pretrained image encoder, and $\mathcal{Q}(\cdot)$ as the function that produces region embedding $\mathbf{r}_b$ from the image features and a given box region proposal b. Mathematically,

$$\mathbf{r}_b = \mathcal{Q}(\mathcal{F}(I), b) \tag{1}$$

Standard detectors use K-way classifier because the training and test time categories are the same. This design does not support the open-vocabulary settings which require new categories to be added at test time. To accommodate this, we replace the last fully connected layer with the text embeddings of base categories (see Fig. 2a). At inference time, we can simply expand the text embeddings to include novel categories for open-vocabulary detection (see Fig. 2b). An advantage of such design is that the system can generalize to the novel categories near C_B in the embedding space.

To generate the text embeddings, it is critical to use the matching text encoder of the image encoder used for initialization in Sec. 3.2. Apart from C_B, the background category is represented by a generic phrase "background" for compatibility with other categories. The proposals not matched to any groundtruth boxes in C_B are treated as background. For each region, we compute the cosine similarity of $\mathbf{r}_b$ with the text embeddings of C_B and "background", and apply a learnable temperature τ on the logits. The detection scores $\mathbf{z}(\mathbf{r}_b)$ are given by:

$$\mathbf{z}(\mathbf{r}_b) = Softmax(\frac{1}{\tau}\left[cos(\mathbf{r}_b, \mathbf{t}_{bg}),\ cos(\mathbf{r}_b, \mathbf{t}_1),\ \cdots,\ cos(\mathbf{r}_b, \mathbf{t}_{|C_B|})\right]) \tag{2}$$

where $cos(\mathbf{a}, \mathbf{b}) = \mathbf{a}^\top \mathbf{b} / (\|\mathbf{a}\| \|\mathbf{b}\|)$, and $\mathbf{t}_i$ denotes the text embeddings of class i. We apply the standard softmax cross entropy loss on the logits (see Fig. 2a). At test time, we expand the text embeddings from C_B to $C_B \cup C_N$ for open-vocabulary detection. Similar designs have been used by previous works [1, 12, 44].

3.4 Open-Vocabulary Recognition

The open-vocabulary recognition ability of F-VLM mainly depends on the pretrained VLM at inference time. We apply the frozen VLM pooling layer $\mathcal{P}(\cdot)$ on the last layer outputs of frozen feature extractor $\mathcal{F}(\cdot)$ (see Sec. 3.2 for notations). The pooling layer is only used at test time. Because the pooling layer requires fixed-size inputs *e.g.* 7x7 for R50 [30], we crop and resize the region features with ROI-Align [15] from the last layer outputs of $\mathcal{F}(\cdot)$ (see Fig. 2b). Unlike existing works [9, 12], we do not crop the image regions and cache their embeddings in an offline process. Using the notations from Eqn. (1), we obtain the VLM region embedding $\mathbf{v}_b$ for a box b by:

$$\mathbf{v}_b = \mathcal{P}(\mathcal{F}(I), b) \tag{3}$$

Similar to Eqn. (2), we compute the VLM scores by cosine similarity as follows:

$$\mathbf{w}(\mathbf{v}_b) = Softmax(\frac{1}{T} \left[cos(\mathbf{v}_b, \mathbf{t}_{bg}),\ cos(\mathbf{v}_b, \mathbf{t}_1),\ \cdots,\ cos(\mathbf{v}_b, \mathbf{t}_{|C_B \cup N|})\right]) \tag{4}$$

where T is a fixed temperature and the text embeddings include both the C_B and C_N at inference time (see Fig. 2b). We use a fixed temperature to adjust the scale of VLM scores relative to the detection scores in Eqn. (2). Even though the VLM was never trained on regions, the cropped region features of $\mathcal{F}(\cdot)$ maintain good open-vocabulary recognition ability. We apply the geometric mean to combine the VLM scores in Eqn. (4) with the detection scores in Eqn. (2). For a region b and category i, we define $w(\mathbf{v}_b)_i$ as the VLM score and $z(\mathbf{r}_b)_i$ as the detection score. The final scores $s(\mathbf{r}_b)_i$ are given by:

$$s(\mathbf{r}_b)_i = \begin{cases} z(\mathbf{r}_b)_i^{(1-\alpha)} \cdot w(\mathbf{v}_b)_i^{\alpha} & \text{if } i \in C_B \\ z(\mathbf{r}_b)_i^{(1-\beta)} \cdot w(\mathbf{v}_b)_i^{\beta} & \text{if } i \in C_N \end{cases} \tag{5}$$

where $\alpha, \beta \in [0, 1]$ control the VLM score weights for base/novel categories, and the background score comes directly from the detector *i.e.*, $s(\mathbf{r}_b)_0 = z(\mathbf{r}_b)_0$. The right side of Fig. 2b illustrates the process of obtaining the open-vocabulary detection scores from the detection and VLM scores. Compared to the ensemble system in [12], our design is simpler without a need for knowledge distillation or double-branch Faster R-CNN heads. As a by-product of frozen weights, F-VLM can perform open-vocabulary classification by setting the region b to the whole image and $s(\mathbf{r}_b)_i = w(\mathbf{v}_b)_i$, which is equivalent to using the pretrained VLM on the whole image directly (*e.g.* CLIP [30]).

3.5 Open-Vocabulary Localization

How to localize and separate the novel objects from the background is an important problem in open-vocabulary detection. Standard detectors are not designed for localizing novel objects because most of them apply class-specific localization, including the box regression and mask prediction heads *e.g.*, Mask R-CNN [15]. Inspired by the learnt objectness [21, 22, 41], we use *class-agnostic* box regression and mask prediction heads instead. That is, for each region proposal, we predict one box and one mask for all categories, rather than one per category. This simple change allows us to localize novel objects in the open-vocabulary settings. We note that F-VLM framework is not specific to the choice of Mask R-CNN detector head and other models can potentially be applied as well *e.g.* [3, 32]. We choose Mask R-CNN following the existing works [9, 12, 44, 49].

4 Experiments

Implementation Details. We choose Mask R-CNN [15] with feature pyramid network [25] as our detector head throughout the paper. The head design follows [11, 12]. We use 1024x1024 input size with large scale jittering augmentation [11] between [0.1, 2.0], batch size 256, weight decay 1e-4, momentum 0.9, and an initial learning rate of 0.36 decayed by 0.1 at [0.8x, 0.9x, 0.95x] of the full training schedule. We train the model for 46.1k iterations which corresponds to 118 LVIS epochs. We adopt linear learning rate warm-up of 1k iterations with initial learning rate of 0.0032.

We re-weight the background in the cross-entropy loss of Faster R-CNN classifier by a factor $\gamma = 0.9$ [44, 49]. For the score combination, we use $\alpha = 0.35$ and $\beta = 0.65$ in Eqn. (5). We use a maximum of 300 detections per image, and set temperature $T = 0.01$ in Eqn. (4) throughout the paper except for the R50x64 backbone on LVIS, where $T = 0.02$ is slightly better. We use the ImageNet prompt templates of CLIP [30] and take the average embeddings over the templates as the text embedding of each category.

4.1 Open-Vocabulary Detection Benchmark

LVIS Benchmark. We evaluate our approach on the LVIS dataset [13] which contains a large and diverse set of 1203 object categories suitable for open-vocabulary detection. Following the existing works [12, 49], we treat the frequent and common categories as the base categories C_B for training, and hold out the rare categories as novel categories C_N for testing. Mask AP_r is the main metric we benchmark on. To ensure reproducibility, we report the mean of 5 independent runs following the protocol of [12] and the best practice of LVIS challenge [13]. For fair comparison, we adopt the same Mask R-CNN head architecture as [12] and use the same large scale jittering training recipe [11, 12].

Table 1 shows our main results on LVIS. In the R50 comparison, F-VLM ranks second among the other alternatives based on knowledge distillation, pretraining, or joint training with additional image-text supervision. The leading DetPro [9] utilizes prompt optimization [50] and SoCo pretraining [42], which are orthogonal to our approach. In the system-level comparison, we observe the performance of F-VLM scales up nicely with frozen model capacity, even though the amount of trainable parameters remains the same. Our best model achieves 32.8 AP_r - the best published results on this benchmark to our knowledge. Compared to the best existing approach (ViLD-EN-B7), we outperform by 6.5 mask AP_r on the novel categories (and +5.6 overall mask AP). Notably, F-VLM is the only approach without knowledge distillation or a trainable backbone.

Table 1: **LVIS Open-Vocabulary Object Detection Benchmark**. F-VLM outperforms the best existing approach by 5.6 mask AP on novel categories. All methods use the same instance-level supervision from LVIS [13] base categories, CLIP [30] pretraining, and fixed prompt templates unless noted otherwise. †: Pre-training with CC-3M [36]. ‡: Prompt optimization [50] and SoCo pretraining [42]. *: Joint training with IN-21k [6] (arxiv). ⋆: ALIGN model [19].

Backbone	Pretrained CLIP [30]	Method	Distill	Trainable Backbone	AP_r	AP_c	AP_f	AP
R50 Comparison:								
R50 [16]	ViT-B/32	ViLD [12]	✓	✓	16.1	20.0	28.3	22.5
R50 [16]	ViT-B/32	ViLD-Ens. [12]	✓	✓	16.6	24.6	30.3	25.5
R50 [9]	ViT-B/32	DetPro [9]‡	✓	✓	19.8	25.6	28.9	25.9
R50 [9]	ViT-B/32	Detic-ViLD[51]*	✗	✓	17.8	26.3	31.6	26.8
R50 [30]	R50	RegionCLIP [49]†	✓	✓	17.1	27.4	34.0	28.2
R50 [30]	R50	F-VLM (Ours)	✗	✗	18.1	23.7	27.0	24.0
System-level Comparison:								
R152 [16]	ViT-B/32	ViLD [12]	✓	✓	18.7	21.1	28.4	23.6
R152 [16]	ViT-B/32	ViLD-Ens. [12]	✓	✓	18.7	24.9	30.6	26.0
EN-B7 [37]	ViT-L/14	ViLD-Ens. [12]	✓	✓	21.7	29.1	33.6	29.6
EN-B7 [37]	EN-B7 [19]⋆	ViLD-Ens. [12]	✓	✓	26.3	27.2	32.9	29.3
R50 [9]	ViT-B/32	DetPro-Cascade [9]‡	✓	✓	20.0	26.7	30.4	27.0
R50 [16]	ViT-B/32	Detic-CN2 [51]*	✗	✓	24.6	32.5	35.6	32.4
R50x4 [30]	R50x4	RegionCLIP [49]†	✓	✓	22.0	32.1	36.9	32.3
R50x4 [30]	R50x4	F-VLM (Ours)	✗	✗	26.3	28.6	29.4	28.5
R50x16 [30]	R50x16	F-VLM (Ours)	✗	✗	30.4	32.6	32.2	32.1
R50x64 [30]	R50x64	F-VLM (Ours)	✗	✗	**32.8**	35.4	35.4	34.9

COCO Benchmark. Many existing works on zero-shot detection [1] and open-vocabulary detection [44, 12, 49] benchmark on COCO. This setup divides COCO vocabulary into 48 base categories for training and 17 novel categories for testing. We follow the standard practice and report results in the generalized detection settings without instance segmentation. The main metric is AP50 of novel categories. Similar to LVIS, we report the mean of 5 independent runs to ensure reproducibility.

Due to the smaller number of training categories, we observe a tendency to overfit when we re-use the same LVIS training recipe. We therefore made the following modifications for training: 1) increase weight decay to 0.01, 2) reduce learning rate to 0.02, 3) reduce batch size to 64, 4) train for 11250 iterations, 5) set background loss weight $\gamma = 0.2$. At test time, we set NMS threshold to 0.4, and $\alpha = 0.2$, $\beta = 0.45$ for score combination in Eqn. (5).

Table 2 shows that F-VLM is very competitive among the published results. Compared to RegionCLIP [49] which uses additional supervision (*e.g.*, CC3M [36]) for pretraining, F-VLM uses a frozen backbone without the pre-training stage. In fact, F-VLM surpasses the CLIP-R50 pretrained version of RegionCLIP by almost 14 points. We consider this baseline more comparable because both use the same R50 backbone and CLIP pre-trained weights, and neither use pretraining on caption data. This shows that F-VLM uses the same pretrained weights more effectively. Compared to OVR-CNN [44] and ViLD [12], F-VLM is very competitive or slightly better without the use of detection-tailored pretraining or knowledge distillation.

Table 2: **COCO Open-Vocabulary Object Detection Benchmark.** F-VLM is very competitive with the other methods trained with various sources. All methods use the ResNet50 backbone [16, 30]. RegionCLIP additionally use COCO Captions† [26] or CC3M‡ [36] for pre-training. *: CLIP initialization without region-level pre-training. *: Joint training with COCO captions (arxiv).

Method	Training source	Novel AP	Base AP	Overall AP
WSDDN [2]	image-level labels in $C_B \cup C_N$	19.7	19.6	19.6
Cap2Det [43]		20.3	20.1	20.1
ZSD [1]	instance-level labels in C_B	0.31	29.2	24.9
DELO [52]		3.41	13.8	13.0
PL [31]		4.12	35.9	27.9
OVR-CNN [45]	image captions in $C_B \cup C_N$ instance-level labels in C_B	22.8	46.0	39.9
CLIP-RPN [12]	image-text pairs from Internet [30] instance-level labels in C_B caption datasets (RegionCLIP, Detic)	26.3	28.3	27.8
ViLD [12]		27.6	59.5	51.3
Detic* [51]		27.8	47.1	45.0
RegionCLIP‡ [49]		**31.4**	57.1	50.4
RegionCLIP† [49]		26.8	54.8	47.5
RegionCLIP* [49]		14.2	52.8	42.7
F-VLM (Ours)		28.0	43.7	39.6

Transfer Detection Benchmark. We explore the potential of F-VLM as a general-purpose detector for different data sources with a view to move towards non dataset-specific detection. F-VLM trained on one dataset can be directly applied to another by swapping out the vocabulary without any finetuning, *e.g.*, replacing the 1203 LVIS categories with COCO 80 categories. The models we use are trained on LVIS base categories and tested on COCO and Objects365-v1 validation splits following the transfer setup of ViLD [12]. These two datasets have smaller vocabularies than LVIS, and category and image overlaps are hard to avoid, *e.g.*, COCO vs LVIS.

Table 3 presents the results in comparison with prior works and supervised baselines. The smallest F-VLM-R50 model is comparable with ViLD [12] and DetPro [9] on Objects365 and inferior on COCO. However, the performance improves consistently on both datasets as we scale up the frozen model without increasing trainable parameters. On Objects365/COCO, the largest F-VLM outperforms existing works ViLD by +3.2/+5.9 and DetPro by +4.9/+5.6, meaningfully closing the gap with the supervised model on COCO (-33%) and Objects365 (-40%).

There is no longer distinction between base and novel categories in this setting, so we assume all categories are novel and use β alone to combine detection and VLM scores in Eqn. (5) (see Sec. 3.4). From the optimal β scores on both datasets, we first observe that only the detection scores are needed ($\beta = 0$) for COCO because of the smaller and highly overlapped vocabulary with LVIS. The story is different on Objects365, where the optimal β starts out at 0.3 and increases to 0.4 with growing model capacity, because the VLM scores become increasingly useful for the long-tail categories.

Table 3: **Generalization ability of the detector trained with F-VLM on LVIS evaluated on COCO and Object365 datasets**. The results are reported in Box AP averaged over 5 runs. We also show the optimal β parameter for each model, where β controls the weighting between detection scores ($\beta = 0$) and VLM scores ($\beta = 1$). We find that COCO detection uses primarily detection scores, whereas Objects365 detection relies more on VLM scores, and β grows with model capacity.

Method	COCO				Objects365			
	β	AP	AP_{50}	AP_{75}	β	AP	AP_{50}	AP_{75}
Supervised [12]	-	46.5	67.6	50.9	-	25.6	38.6	28.0
ViLD-R50 [12]	-	36.6	55.6	39.8	-	11.8	18.2	12.6
DetPro-R50 [9]	-	34.9	53.8	37.4	-	12.1	18.8	12.9
F-VLM-R50 (Ours)	0.0	32.7	53.1	34.8	0.30	11.8	19.1	12.5
F-VLM-R50x4 (Ours)	0.0	36.0	57.5	38.7	0.30	14.2	22.6	15.2
F-VLM-R50x16 (Ours)	0.0	37.9	59.6	41.2	0.40	16.2	25.3	17.5
F-VLM-R50x64 (Ours)	0.0	**39.8**	**61.6**	**43.8**	0.40	**17.7**	**27.4**	**19.1**

4.2 Analysis and Ablations

Training Resource Benchmark. We explore the benefits of frozen VLMs in terms of training resource savings. We benchmark with ViLD [12] as it is most comparable. F-VLM adopts the same Mask R-CNN head configuration and large scale jittering training recipe [11] as ViLD. Neither approaches require a separate detection-tailored pretraining. We follow ViLD and compare the total cost for training the open-vocabulary detector on TPUv3 cores. All methods use the same batch size of 256. We use 32 cores while ViLD uses 128 cores for training. The data about ViLD training time is obtained directly from the authors [12]. To keep the benchmark simple, we assume the pretrained VLMs are given and exclude their training costs from the comparison. For F-VLM, we use the R50x64 backbone in this benchmark and our default number of iterations is 46.1k (118 epochs). We report mean results over 5 independent runs.

Table 4 shows that F-VLM is significantly faster and cheaper to train. F-VLM can train with very few epochs (*e.g.* 14.7), and still achieve a state of the art AP_r of 31.0. Compared to ViLD-ENB7 (AP_r 26.3) which trains for 460 epochs, the shortest run of F-VLM (AP_r 27.7) can train for only 7.4 epochs and is 226× more compute-efficient (57× faster in wall clock time). We believe the efficiency gain arises from the frozen backbone, which substantially simplifies the learning process. This is orthogonal to the detection-tailored pretraining used by exiting works to speed up training [9, 42, 44, 49]. The F-VLM system runs almost as fast as a standard detector [15] at inference time, because the only addition is a single attention pooling layer [30] on the detected region features (see Fig. 2b).

Table 4: **Training Time and Resource Benchmark.** We report LVIS mask AP_r to show the performance vs training cost trade-off. Because of the frozen backbone, F-VLM can still outperform the state of the art with 226× less per-core-hours on TPU-v3. In terms of wall clock time, our shortest run completes in 1.1 hour while the baseline takes 62.5 hours (57× faster).

Method	Mask AP_r	#Iters	Epochs	Training Cost (Per-Core-Hour)	Training Cost Savings
ViLD-EN-B7 [12]	26.3	180k	460	8000	1×
ViLD-R50 [12]	16.1	180k	460	4252	1.9×
F-VLM (Ours)	32.8	46.1k	118	565	14×
F-VLM (Ours)	32.5	11.5k	29.5	141	57×
F-VLM (Ours)	31.0	5.76k	14.7	71	113×
F-VLM (Ours)	27.7	**2.88k**	**7.4**	**35**	**226**×

Ablations. In Table 5 we study the score combination parameters in Sec. 3.4 using R50x64 on LVIS. In Table 5a, we vary the VLM score weight for novel categories and observe that $\beta = 0.65$ is most beneficial (see Eqn. (5)). This suggests that open-vocabulary detection primarily relies on VLM predictions. However, VLM scores are not sufficient by themselves as increasing β further yields sub-optimal performance. We note that $\beta = 1$ is a special case where we use only the VLM scores without detection scores to evaluate the emergent object recognition capability of the VLM (17.6 AP_r). From Table 5b we observe that $\alpha = 0.35$ achieves a good trade-off between open-vocabulary and standard detection (see Eqn. (5)). A higher α can slightly boost the AP_r by suppressing the base category detections, but clearly compromises the overall AP. In Table 5c, we study the temperature

Figure 3: **Visualization of F-VLM detections**. 1st col.: Full LVIS detection. 2-3rd col.: Novel object detection on LVIS (see the white arrows). We only show the novel categories for clarity. 4-5th col.: Transfer detection on Objects365. F-VLM can detect many long-tail categories (see white arrows) without finetuning. 6th col.: Transfer to COCO. (Zoom in to see the predicted labels.)

which controls the scale of VLM logits relative to the detection scores (see Eqn. (4)), and found $T = 0.02$ is optimal for both open-vocabulary and standard detection. In comparison, we found the learnt τ converges to approximately 1.0 at the end of training (see Eqn.(2)), which highlights the need to use a separate temperature T for VLM scores. For the smaller backbones (*i.e.* R50-R50x16), we found $T = 0.01$ works best on LVIS.

Table 5: **F-VLM Ablations**. We report Mask AP_r/AP on LVIS. Default settings are marked in gray.

(a) **VLM-score weights for novel classes.** $\beta = 0.65$ benefits both open-vocabulary and standard detection. We fix $\alpha = 0.35$.

β	AP_r	AP
0.35	26.7	34.0
0.50	30.1	34.6
0.65	**32.6**	**35.0**
0.80	31.8	34.5
1.00	17.6	29.8

(b) **VLM-score weights for base classes.** $\alpha = 0.35$ is a good trade-off between open-vocabulary and standard detection. We fix $\beta = 0.65$.

α	AP_r	AP
0.15	30.5	34.3
0.25	31.4	34.7
0.35	**32.6**	**35.0**
0.55	32.8	34.0
0.75	31.9	30.0

(c) **Temperature.** We found $T = 0.02$ is optimal for both open-vocabulary and standard detection.

T	AP_r	AP
0.005	29.3	31.5
0.01	31.7	33.9
0.02	**32.6**	**35.0**
0.04	28.7	34.7
0.08	23.6	33.9

Visualization. Figure 3 visualizes the detections of a F-VLM-R50x4 trained on LVIS base categories. On LVIS, F-VLM is able to detect a wide range of objects (*e.g.*, ski gears, car parts) and accurately capture many novel objects (*e.g.*, fedora, martini, and pennant). We also visualize the transfer detection on Objects365 and COCO datasets by replacing the vocabulary without finetuning. F-VLM correctly detects common (*e.g.*, boat), and long-tail objects (*e.g.*, camel, slide, goldfish).

Limitations. Open-vocabulary object detection is a challenging task and our models are still far from perfect. For example, the region proposal network of F-VLM sometimes struggles to separate novel objects from the background. Given a poorly localized region, F-VLM sometimes classifies it confidently to foreground classes, because the VLMs are less sensitive to localization errors.

Societal Impact. F-VLM directly uses the knowledge in pretrained VLMs, which means the biases of VLMs can propagate into it. We use F-VLM to demonstrate its capabilities and compare with existing works, and recommend careful analysis of ethical risks before using it for other purposes.

4.3 Conclusion

We present F-VLM – a simple open-vocabulary detection method built upon *frozen* VLMs without a need for knowledge distillation or detection-tailored pretraining. F-VLM has much fewer trainable parameters and trains significantly faster than prior works with up to 200× compute savings. Finally, F-VLM outperforms the state of the art on LVIS open-vocabulary benchmark by 6.5 AP_r, and shows very competitive transfer detection on COCO and Objects365.

References

[1] A. Bansal, K. Sikka, G. Sharma, R. Chellappa, and A. Divakaran. Zero-shot object detection. In *ECCV*, 2018.

[2] H. Bilen and A. Vedaldi. Weakly supervised deep detection networks. In *CVPR*, 2016.

[3] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[4] X. Chen and A. Gupta. Webly supervised learning of convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1431–1439, 2015.

[5] B. Demirel, R. G. Cinbis, and N. Ikizler-Cinbis. Zero-shot object detection by hybrid region embedding. In *BMVC*, 2018.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[7] K. Desai and J. Johnson. Virtex: Learning visual representations from textual annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11162–11173, 2021.

[8] S. K. Divvala, A. Farhadi, and C. Guestrin. Learning everything about anything: Webly-supervised visual concept learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3270–3277, 2014.

[9] Y. Du, F. Wei, Z. Zhang, M. Shi, Y. Gao, and G. Li. Learning to prompt for open-vocabulary object detection with vision-language model. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[10] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov. Devise: A deep visual-semantic embedding model. In *NeurIPS*, 2013.

[11] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2918–2928, 2021.

[12] X. Gu, T.-Y. Lin, W. Kuo, and Y. Cui. Open-vocabulary object detection via vision and language knowledge distillation. In *International Conference on Learning Representations*, 2022.

[13] A. Gupta, P. Dollar, and R. Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019.

[14] N. Hayat, M. Hayat, S. Rahman, S. Khan, S. W. Zamir, and F. S. Khan. Synthesizing the unseen for zero-shot object detection. In *ACCV*, 2020.

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, 2017.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[17] X. He and Y. Peng. Fine-grained image classification via combining vision and language. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5994–6002, 2017.

[18] D. Jayaraman and K. Grauman. Zero shot recognition with unreliable attributes. *NeurIPS 2014*, 2014.

[19] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. V. Le, Y. Sung, Z. Li, and T. Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. *ICML*, 2021.

[20] A. Joulin, L. v. d. Maaten, A. Jabri, and N. Vasilache. Learning visual features from large weakly supervised data. In *European Conference on Computer Vision*, pages 67–84. Springer, 2016.

[21] D. Kim, T.-Y. Lin, A. Angelova, I. S. Kweon, and W. Kuo. Learning open-world object proposals without learning to classify. *IEEE Robotics and Automation Letters*, 7(2):5453–5460, 2022.

[22] W. Kuo, B. Hariharan, and J. Malik. Deepbox: Learning objectness with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[23] L. H. Li*, P. Zhang*, H. Zhang*, J. Yang, C. Li, Y. Zhong, L. Wang, L. Yuan, L. Zhang, J.-N. Hwang, K.-W. Chang, and J. Gao. Grounded language-image pre-training. In *CVPR*, 2022.

[24] Y. Li, H. Mao, R. Girshick, and K. He. Exploring plain vision transformer backbones for object detection. *arXiv preprint arXiv:2203.16527*, 2022.

[25] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[26] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[27] M. Minderer, A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen, X. Wang, X. Zhai, T. Kipf, and N. Houlsby. Simple open-vocabulary object detection with vision transformers, 2022.

[28] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, and J. Dean. Zero-shot learning by convex combination of semantic embeddings. *ICLR*, 2014.

[29] H. Pham, Z. Dai, G. Ghiasi, H. Liu, A. W. Yu, M. Luong, M. Tan, and Q. V. Le. Combined scaling for zero-shot transfer learning. *CoRR*, abs/2111.10050, 2021.

[30] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. *ICML*, 2021.

[31] S. Rahman, S. Khan, and N. Barnes. Improved visual-semantic alignment for zero-shot object detection. In *AAAI*, 2020.

[32] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[33] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

[34] M. Rohrbach, M. Stark, and B. Schiele. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *CVPR*, 2011.

[35] M. B. Sariyildiz, J. Perez, and D. Larlus. Learning visual representations with caption annotations. In *European Conference on Computer Vision*, pages 153–170. Springer, 2020.

[36] P. Sharma, N. Ding, S. Goodman, and R. Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018.

[37] M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[38] M. Tsimpoukelli, J. Menick, S. Cabi, S. Eslami, O. Vinyals, and F. Hill. Multimodal few-shot learning with frozen language models. In *Conference on Neural Information Processing Systems*, 2021.

[39] C. Vasconcelos, V. Birodkar, and V. Dumoulin. Proper reuse of image classification features improves object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[40] J. Wang, K. Markert, M. Everingham, et al. Learning models for object recognition from natural language descriptions. In *BMVC*, 2009.

[41] R. Wang, D. Mahajan, and V. Ramanathan. What leads to generalization of object proposals? In *European Conference on Computer Vision*, pages 464–478. Springer, 2020.

[42] F. Wei, Y. Gao, Z. Wu, H. Hu, and S. Lin. Aligning pretraining for detection via object-level contrastive learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 22682–22694. Curran Associates, Inc., 2021.

[43] K. Ye, M. Zhang, A. Kovashka, W. Li, D. Qin, and J. Berent. Cap2det: Learning to amplify weak caption supervision for object detection. In *ICCV*, 2019.

[44] A. Zareian, K. D. Rosa, D. H. Hu, and S.-F. Chang. Open-vocabulary object detection using captions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14393–14402, June 2021.

[45] A. Zareian, K. D. Rosa, D. H. Hu, and S.-F. Chang. Open-vocabulary object detection using captions. In *CVPR*, 2021.

[46] X. Zhai, X. Wang, B. Mustafa, A. Steiner, D. Keysers, A. Kolesnikov, and L. Beyer. Lit: Zero-shot transfer with locked-image text tuning. *arXiv preprint arXiv:2111.07991*, 2021.

[47] Y. Zheng, R. Huang, C. Han, X. Huang, and L. Cui. Background learnable cascade for zero-shot object detection. In *ACCV*, 2020.

[48] Y. Zhong, J. Shi, J. Yang, C. Xu, and Y. Li. Learning to generate scene graph from natural language supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1823–1834, 2021.

[49] Y. Zhong, J. Yang, P. Zhang, C. Li, N. Codella, L. H. Li, L. Zhou, X. Dai, L. Yuan, Y. Li, and J. Gao. Regionclip: Region-based language-image pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[50] K. Zhou, J. Yang, C. C. Loy, and Z. Liu. Conditional prompt learning for vision-language models. In *CVPR*, 2022.

[51] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra. Detecting twenty-thousand classes using image-level supervision. *arXiv preprint arXiv:2201.02605*, 2022.

[52] P. Zhu, H. Wang, and V. Saligrama. Don't even look once: Synthesizing features for zero-shot detection. In *CVPR*, 2020.

Checklist

1. Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See the abstract and Introduction.
2. Did you describe the limitations of your work? [Yes] See the end of Section 4.2
3. Did you discuss any potential negative societal impacts of your work? [Yes] See the end of Section 4.2
4. Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
5. Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] We plan to release the code at a later time.
6. Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Implementation Details under Section 3.5
7. Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We report mean over 5 runs for comparison with other works. We did not report error bars following existing works, but we're happy to provide them separately.
8. Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See the Training Resource Benchmark in Section 4.2
9. If your work uses existing assets, did you cite the creators? [Yes] Yes we cited the creators of the datasets and models we use.
10. Did you mention the license of the assets? [N/A] We use only publicly available detection datasets (COCO, LVIS, Objects365) and CLIP models.

11. Did you include any new assets either in the supplemental material or as a URL? [No] We plan to release the code at a later time.
12. Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] We use only publicly available detection datasets (COCO, LVIS, Objects365) and CLIP models.
13. Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] We use only publicly available detection datasets (COCO, LVIS, Objects365) and CLIP models.