Simple Open-Vocabulary Object Detection with Vision Transformers

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

Google Research
{mjlm,agritsenko}@google.com

- Problem / objective
  - Open-Vocabulary Object Detection
- Contribution / Key idea
  - **OWL-VIT v1**: Vision Transformer for Open-World Localization

- **Model**

1. Architecture
   - 1.1. Image encoder: Vision Transformer
     - 1.1.1. pooling 없애고 object detection에 맞게 head 추가
   - 1.2. Text encoder: Transformer
2. Query embeddings
   - 2.1. Open-vocabulary에 맞게 learned class embeddings (classification head's weight) 대신 text embeddings 사용.
   - 2.2. Query embedding으로 text- 대신 image-conditioned embeddings 사용 가능.

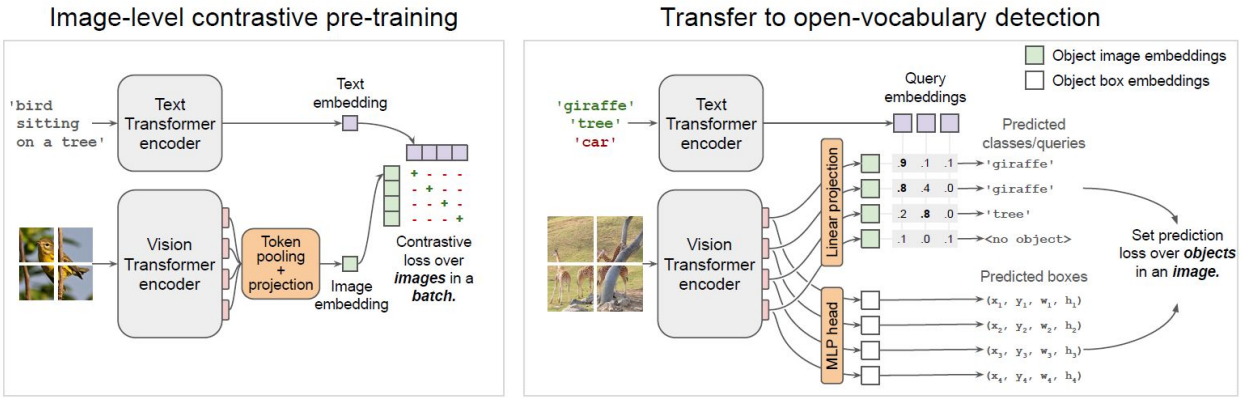
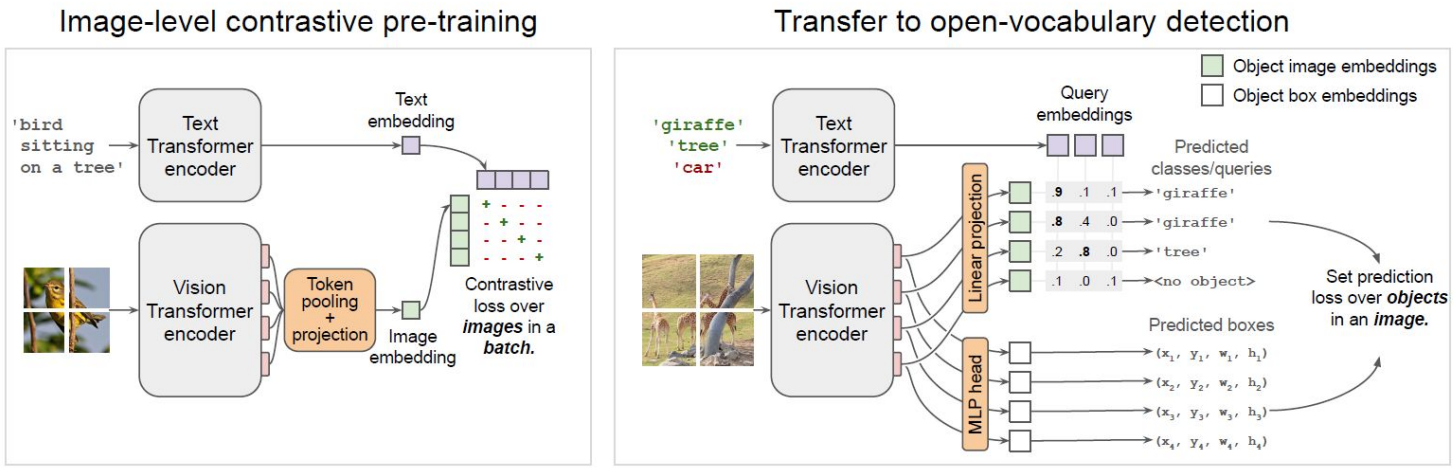


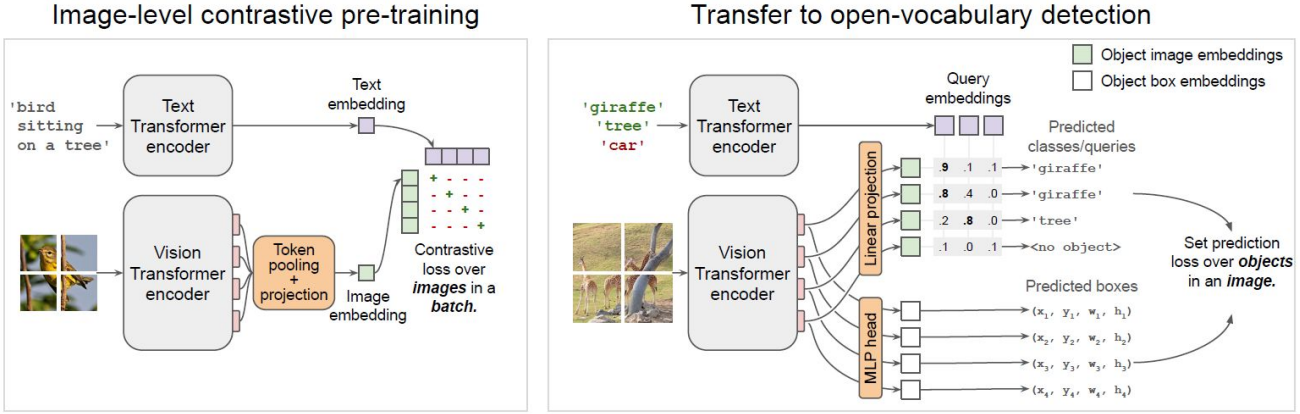
Fig. 1. Overview of our method. *Left:* We first pre-train an image and text encoder contrastively using image-text pairs, similar to CLIP [33], ALIGN [19], and LiT [44]. *Right:* We then transfer the pre-trained encoders to open-vocabulary object detection by removing token pooling and attaching light-weight object classification and localization heads directly to the image encoder output tokens. To achieve open-vocabulary detection, query strings are embedded with the text encoder and used for classification. The model is fine-tuned on standard detection datasets. At inference time, we can use text-derived embeddings for open-vocabulary detection, or image-derived embeddings for few-shot image-conditioned detection.

- **Training**

1. Stage 1: Image-Level Contrastive Pre-Training.
   1.1. Training objective: Contrastive loss b/w image and text representations
   1.2. Image representation: MAP (Multi-head Attention Pooling) embedding
   1.3. Text representation: EOS token



- **Training**
  1. Stage 2: Training the Detector.
     - 1.1. Classification head: outputs logits over per-image label space defined by the queries. (not a fixed global label space)
       - 1.1.1. ∵ we provide the set of object category names as queries for each image.
     - 1.2. Training objective: Bipartite matching loss (cf. DETR)
     - 1.3. 클래스 수 많은 데이터셋: annotated in federated manner
       - 1.3.1. Federated annotation: 이미지 내 존재하는 모든 클래스들에 대하여 라벨링하는 것이 아니라, 일부 클래스에 대해서만 라벨링
         - 1.3.1.1. 각 이미지마다 positive annotation (present) + negative annotation (known to be absent) 존재
         - 1.3.1.2. 이미지마다 최소 50개의 negative annotation이 존재하도록 "pseudo-negatives"도 포함시킴
         - 1.3.1.3. 학습 시, 이 두 annotations 모두 쿼리로 사용
       - 1.3.2. Non-disjoint label spaces: 하나의 객체가 multi-label 가질 수 있음
       - 1.3.3. Training objective: Focal sigmoid cross-entropy loss



- **Experiments**

Fig. A1. Text conditioning examples. Prompts: `"an image of a {}"`, where `{}` is replaced with one of `bookshelf`, `desk lamp`, `computer keyboard`, `binder`, `pc computer`, `computer mouse`, `computer monitor`, `chair`, `drawers`, `drinking glass`, `ipod`, `pink book`, `yellow book`, `curtains`, `red apple`, `banana`, `green apple`, `orange`, `grapefruit`, `potato`, `for sale sign`, `car wheel`, `car door`, `car mirror`, `gas tank`, `frog`, `head lights`, `license plate`, `door handle`, `tail lights`.

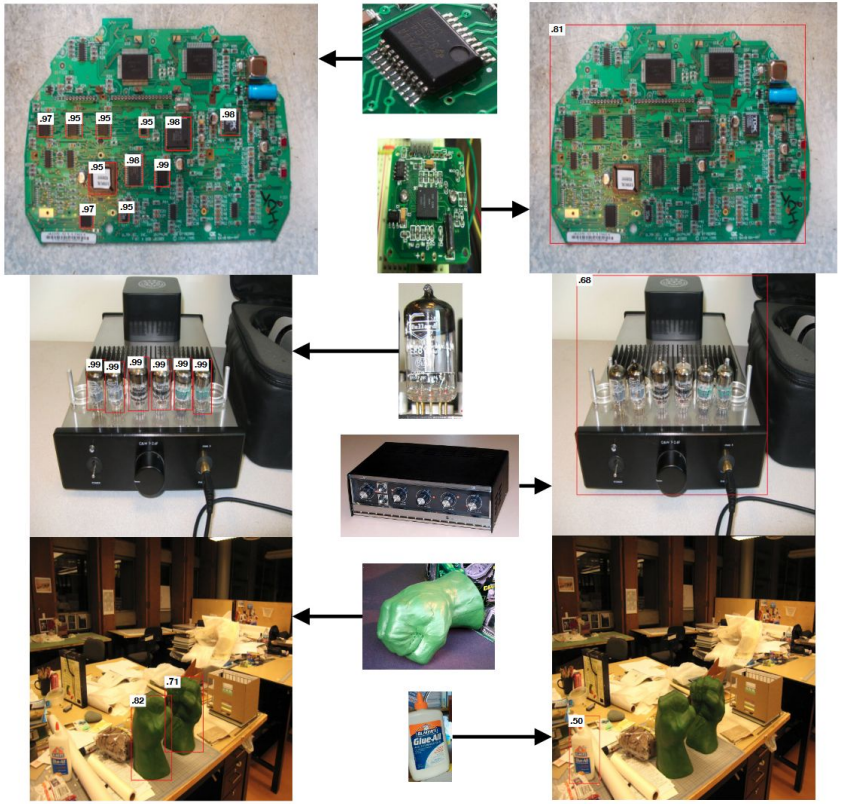


Fig. A2. Image conditioning examples. The center column shows the query patches and the outer columns show the detections along with the similarity score.

Scaling Open-Vocabulary Object Detection

Matthias Minderer **Alexey Gritsenko** **Neil Houlsby**

Google DeepMind

{mjlm, agritsenko, neilhoulsby}@google.com

- Problem / objective
  - Open-Vocabulary Object Detection
- Contribution / Key idea
  - **OWL-ST**: Self-Training recipe applied for OWL-ViT detection architecture
  - **OWL-VIT v2**: OWL-VIT v1 보완한 v2 모델

- **Motivation**

1. 문제: Detection training data의 부족
2. 해결: OWL-VIT v1 이 만든 pseudo-label 로 self-training하는 방법 제안 (OWL-ST)

- **Objective**

1. 본 approach의 3 key components 최적화
    1.1. Label space
    1.2. Annotation filtering
    1.3. Training efficiency

- **Overview**

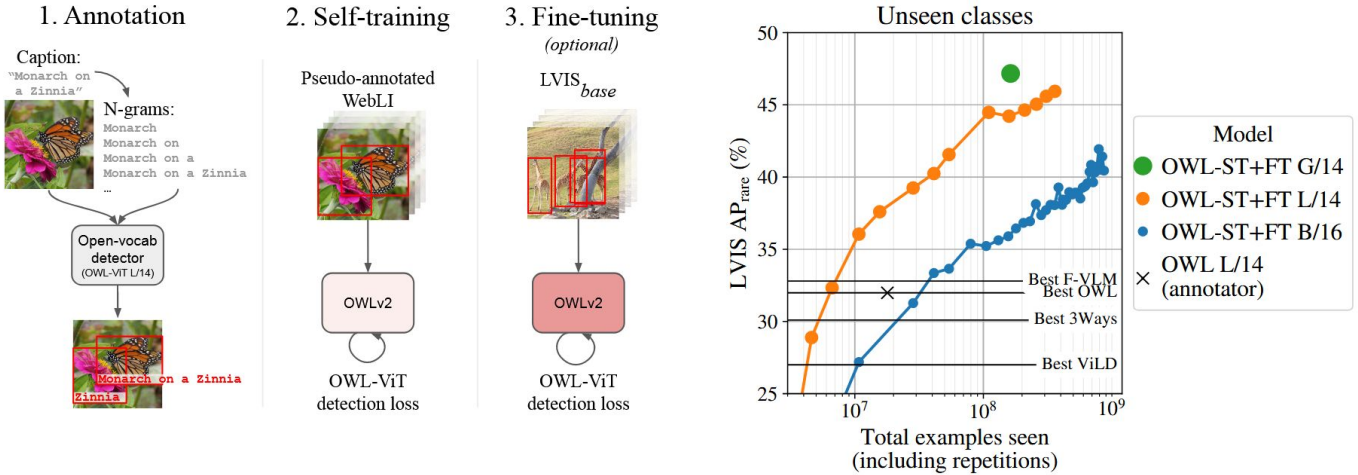
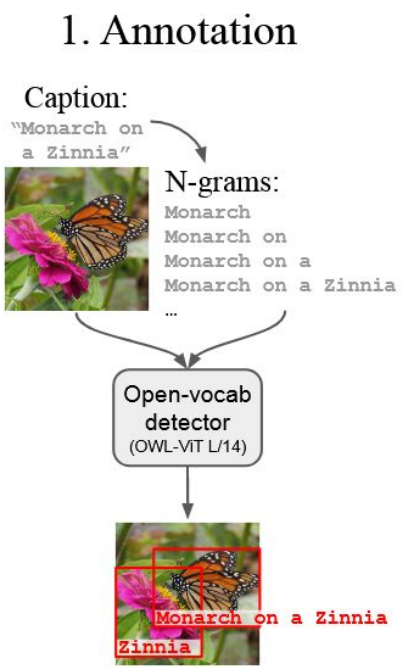


Figure 1: Overview of our method. **Left:** Our method consists of three steps: (1) Generate pseudo-box annotations on WebLI with OWL-ViT L/14, queried with caption N-grams. (2) Train new models on pseudo-annotations. (3) Optionally, fine-tune on human annotations. **Right:** Zero-shot detection performance on $\text{LVIS}_{\text{rare}}$ after fine-tuning on $\text{LVIS}_{\text{base}}$. Neither the annotator nor our models have seen any human-generated box annotations for $\text{LVIS}_{\text{rare}}$ classes. Our self-training approach improves over other methods even at moderate amounts of training (e.g. the OWL-L/14 model we use as annotator; black $\times$), and continues to improve as training is scaled up. Horizontal black lines indicate previous state-of-the-art open-vocabulary detectors which did not see $\text{LVIS}_{\text{rare}}$ classes during training.

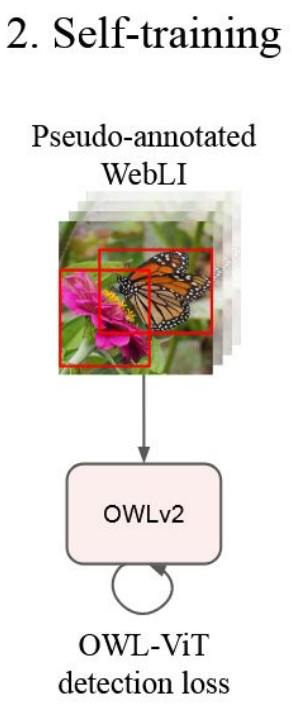
- **Stage1: Generating Web-Scale Open-Vocabulary Object Annotations**

1. WebLI 데이터셋 사용 (10B images, alt-text strings으로 구성)
2. OWL-ViT CLIP-L/14 모델 사용해서 10B WebLI images의 bounding box pseudo-annotations 획득
   - 2.1. class-agnostic하게 objects detect하고, 각 detected object의 each text query와의 similarity 통해 점수 계산
3. 아래 두 annotation label space 모두 사용
   - 3.1. Human-curated label space
     - 3.1.1. 4가지 데이터셋의 label space 합침 (중복, 복수형 제거)
   - 3.2. Machine-generated label space
     - 3.2.1. 선행 연구: grammatical parsing (image caption에서 명사구 및 개념 추출하는 방식) + strict filtering
     - 3.2.2. 선행 연구의 문제: extracted queries의 다양성 부족
     - 3.2.3. Ours: 모든 가능한 N-grams (N<=10) 을 detection prompt로 사용 후, 결과 pseudo-labels에 weak confidence filtering 적용 (generic terms 또는 stop-words로만 구성된 단어만 제거)
     - 3.2.4. 본 논문 마인드: "let the data do the work" (기존: little but clean -> Ours: noisy but many)



- **Stage2: Self-training at Scale**

1. OWL-ViT v1과 전반적으로 유사하게 학습
2. OWL-VIT v1과 OWL-VIT v2의 학습 차이점 3가지
    - 2.1. Token Dropping.
        - 2.1.1. 이미지 패치 내 픽셀들 분산값 낮은 하위 50%에 해당되는 토큰은 학습 시 미포함. (e.g., 하늘)
        - 2.1.2. 효율성 추구 목적
    - 2.2. Instance Selection.
        - 2.2.1. OWL-VIT v1: 각 인코더 토큰마다 바운딩박스 하나씩 예측
            - 2.2.1.1. 문제: 토큰수 >>> 객체수 의 경우가 대다수이기에 매우 비효율적.
        - 2.2.2. 해결: Objectness head 추가
            - 2.2.2.1. Objectness head: 각 토큰이 객체를 포함하는지 예측
                - 2.2.2.1.1. 인풋: encoder token, 아웃풋: scalar objectness score
            - 2.2.2.2. Classification head의 score로 학습
        - 2.2.3. Objectness 점수 상위 10%에 해당되는 토큰들만 학습에 사용
    - 2.3. Mosaics.
        - 2.3.1. 기존보다 더 많이 그리드화; raw image를 6x6 까지 grid 결합.
            - 2.3.1.1. 이유1: 주어진 해상도에서 더 많은 raw image로 학습할수있으니 효율성 증가.
            - 2.3.1.2. 이유2: Web images의 평균 해상도 낮음.
            - 2.3.1.3. 기대되는 효과: 평균 객체 사이즈 줄어, 작은 객체에 대한 성능 향상.
        - 2.3.2. 1×1, 2×2, 3×3, 4×4, and 6×6 grids 균등 비율로 구성.
        - 2.3.3. 각 학습 이미지에, 평균 13.2개의 raw image 들어있다고 보면됨.



1. Token Dropping.

Table A1: Performance of standard OWL-ViT (L/14), trained on Objects365 and Visual Genome as in [21], for different token drop rates. For drop rate 0.0, the standard deviation over three runs is given.

	Token drop rate				
Metric	0.00	0.25	0.33	0.50	0.70
LVIS AP^{val}_{all}	33.3 ±0.33	33.1	33.6	32.9	30.4
LVIS AP^{val}_{rare}	31.8 ±1.16	31.0	32.6	30.8	28.2

2. Mosaics.

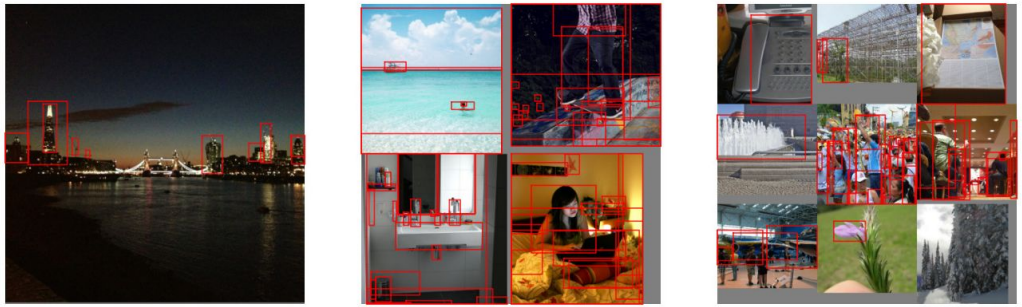


Fig. A4. Example training images. Ground-truth boxes are indicated in red. From left to right, a single image, a 2 × 2 mosaic, and a 3 × 3 mosaic are shown. Non-square images are padded at the bottom and right (gray color).

- **(Optional) Stage3: Fine-tuning**

1. Human annotations에 fine-tuning
2. "Fine-tuned classes에서 성능 향상"과 "open-vocabulary 성능 저하" 간의 trade-off 존재

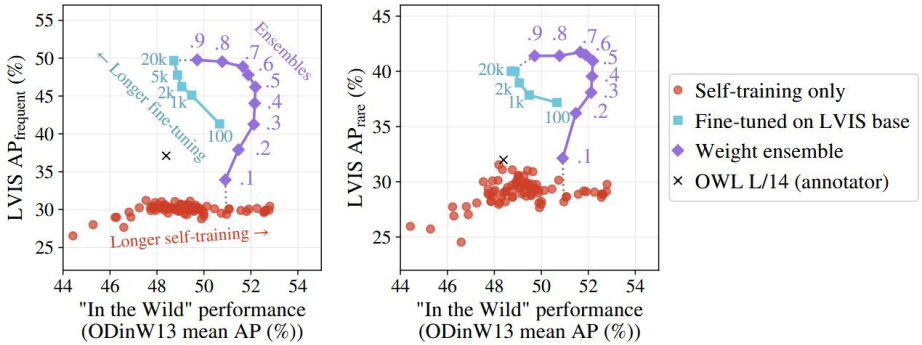
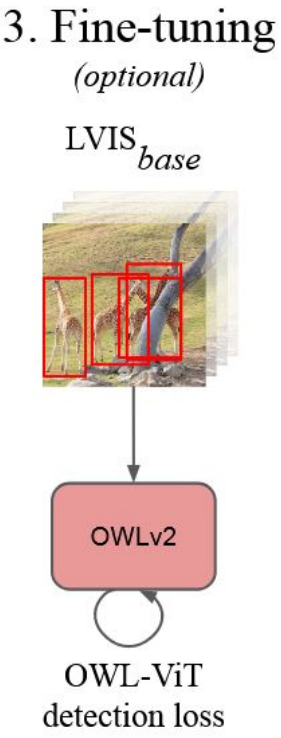


Figure 5: Trade-off between fine-tuned and open-world performance. Self-training yields continued improvements on a suite of diverse datasets (ODinW13; x-axis), but performance on any given dataset (e.g. LVIS; y-axis) may saturate (red circles). Fine-tuning on a target dataset improves performance on that dataset, but reduces the open-world generalization ability in proportion to the finetuning duration (light blue squares; numbers indicate finetuning steps). This trade-off can be improved through weight-space ensembling (averaging) of the pretrained and fine-tuned checkpoints [31] (purple diamonds; numbers indicate the mixing coefficient for the fine-tuned weights). The plot shows B/16 models self-trained on N-gram pseudo-annotations and evaluated either directly after self-training or after fine-tuning on $LVIS_{base}$. Ensembles were created between the longest-self-trained checkpoint and the weights obtained after finetuning that checkpoint for 20k steps. Note that there is significant variability in ODinW13 performance between checkpoints towards the end of self-training.



- **Experiments**

Table 1: Open-vocabulary detection performance on LVIS and ODinW. Rows for our models are shown in blue. None of our models have seen any human box annotations for $\text{LVIS}_{\text{rare}}$ classes at any stage of training, so LVIS $\text{AP}^{\text{val}}_{\text{rare}}$ (rightmost column) measures zero-shot performance. Numbers in green or red indicate the difference to the prior state of the art, i.e. F-VLM R50x64 in the open-vocabulary (top) part of the table and DetCLIPv2 Swin-L in the curated-vocabulary (bottom) part. Gray O+VG indicates that O365+VG were used indirectly (for training the annotator). Gray ODinW numbers indicate that these models were trained on OpenImages data, which overlaps with ODinW. AP^{mini} refers to the LVIS "minival" split introduced by MDETR [?].

	Method	Backbone	Self-training data	Self-training vocabulary	Human box annotations	ODinW 13	LVIS $\text{AP}^{\text{mini}}_{\text{all}}$	LVIS $\text{AP}^{\text{mini}}_{\text{rare}}$	LVIS $\text{AP}^{\text{val}}_{\text{all}}$	**LVIS $\text{AP}^{\text{val}}_{\text{rare}}$**
Open vocabulary *(evaluation vocabulary is not available at training time):*										
1	RegionCLIP [40]	R50x4	CC3M	6k concepts	$\text{LVIS}_{\text{base}}$	–	–	–	32.3	22.0
2	OWL [21]	CLIP B/16	–	–	O365+VG	–	–	–	27.2	20.6
3	OWL [21]	CLIP L/14	–	–	O365+VG	48.4	–	–	34.6	31.2
4	GLIPv2 [39]	Swin-T	Cap4M	tokens	O365+GoldG	48.5	29.0	–	–	–
5	GLIPv2 [39]	Swin-B	CC15M	tokens	FiveODs+GoldG	54.2	48.5	–	–	–
6	GLIPv2 [39]	Swin-H	CC15M	tokens	FiveODs+GoldG	55.5	50.1	–	–	–
7	F-VLM [14]	R50x4	–	–	$\text{LVIS}_{\text{base}}$	–	–	–	28.5	26.3
8	F-VLM [14]	R50x64	–	–	$\text{LVIS}_{\text{base}}$	–	–	–	34.9	32.8
9	3Ways [1]	NFNet-F0	CC12M	captions	$\text{LVIS}_{\text{base}}$	–	–	–	35.7	25.6
10	3Ways [1]	NFNet-F6	CC12M	captions	$\text{LVIS}_{\text{base}}$	–	–	–	44.6	30.1
11	OWL-ST	CLIP B/16	WebLI	N-grams	O+VG	48.8	31.8	35.4	27.0	29.6 -3.2
12	OWL-ST	CLIP L/14	WebLI	N-grams	O+VG	53.0	38.1	39.0	33.5	34.9 +2.1
13	OWL-ST	SigLIP G/14	WebLI	N-grams	O+VG	49.9	37.8	40.9	33.7	37.5 +4.7
14	OWL-ST+FT	CLIP B/16	WebLI	N-grams	O+VG, $\text{LVIS}_{\text{base}}$	48.6	47.2	37.8	41.8	36.2 +3.4
15	OWL-ST+FT	CLIP L/14	WebLI	N-grams	O+VG, $\text{LVIS}_{\text{base}}$	50.1	54.1	46.1	49.4	44.6 +11.8
16	OWL-ST+FT	SigLIP G/14	WebLI	N-grams	O+VG, $\text{LVIS}_{\text{base}}$	50.1	51.3	50.9	47.0	47.2 +14.4
Human-curated vocabulary *(evaluation vocabulary may be accessed at training time):*										
17	Detic [41]	R50	IN-21k	LVIS classes	$\text{LVIS}_{\text{base}}$	–	–	–	32.4	24.6
18	DetCLIPv2 [32]	Swin-T	CC15M	Nouns+curated	O365+GoldG	–	40.4	36.0	32.8	31.0
19	DetCLIPv2 [32]	Swin-L	CC15M	Nouns+curated	O365+GoldG	–	44.7	43.1	36.6	33.3
20	OWL-ST+FT	CLIP B/16	WebLI	N-grm+curated	O+VG, $\text{LVIS}_{\text{base}}$	48.9	51.1	41.9	45.6	40.5 +7.2
21	OWL-ST+FT	CLIP L/14	WebLI	N-grm+curated	O+VG, $\text{LVIS}_{\text{base}}$	48.7	55.8	50.0	50.4	45.9 +12.6

- **Experiments**

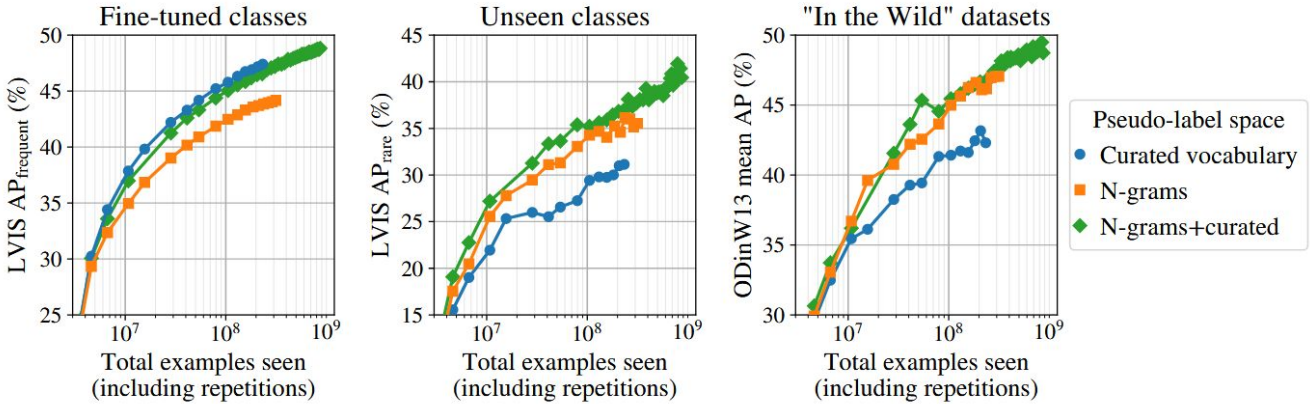


Figure 2: Comparison of pseudo-label spaces. Self-training on a human-curated list of classes yields good downstream performance on these classes, but generalizes poorly to unseen classes and datasets. Open-vocabulary generalization can be improved by obtaining weak but diverse supervision from image-associated text. WebLI image-text data was pseudo-annotated using OWL-ViT CLIP-L/14 with one of three label spaces: *Curated vocabulary* (the union of label spaces from LVIS, Objects365, OpenImagesv4, and Visual Genome), *N-grams* (lightly filtered N-grams from the text associated with each image), or a combination of both (*N-grams + curated*). OWLv2-B/16 models were then self-trained on the pseudo-annotations and fine-tuned on $LVIS_{base}$. Each point represents a separate fine-tuning run. "Examples seen" refers to the number of images after creating mosaics; the total number of raw images seen is 13.2× that number (Section 3.2).

- **Experiments**

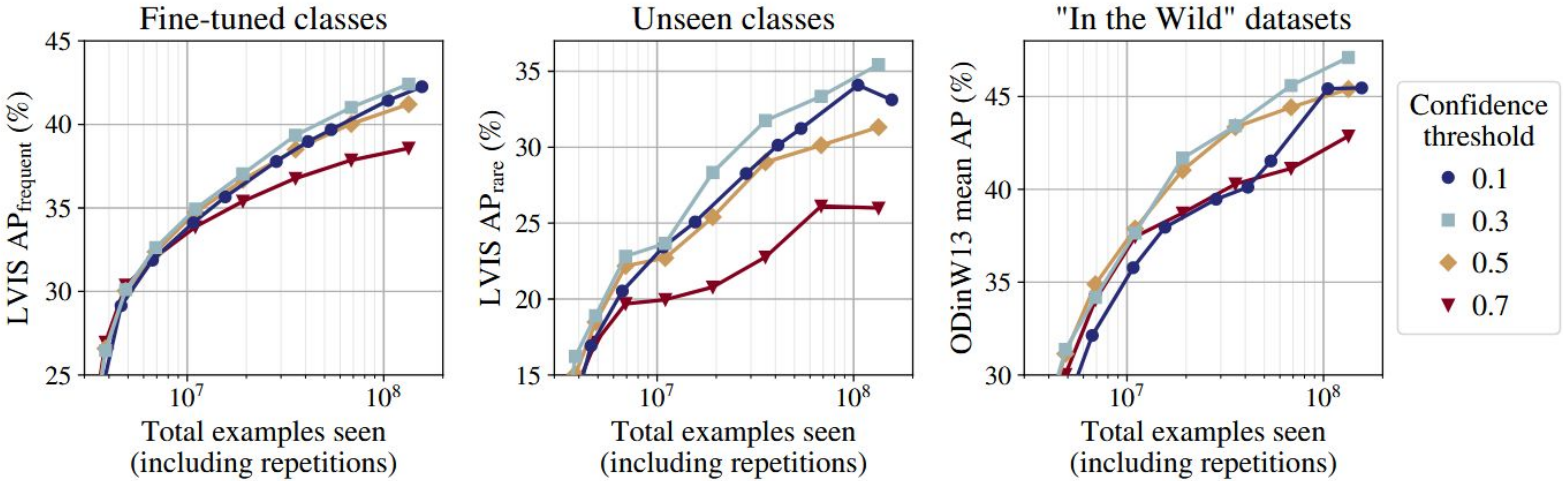


Figure 3: Impact of pseudo-annotation filtering by detection confidence on self-training effectiveness. Pseudo-labels (N-gram label space) were filtered using different confidence thresholds. Number of remaining images for each threshold: 0.1: 5B, 0.3: 2B, 0.5: 782M, 0.7: 224M. OWLv2-B/16 detectors were self-trained on the filtered pseudo-annotations and fine-tuned on $LVIS_{base}$. Each point represents a different fine-tuning run. "Examples seen" refers to the number of images after creating mosaics; the total number of raw images seen is 13.2× that number (Section 3.2).

- **Experiments**

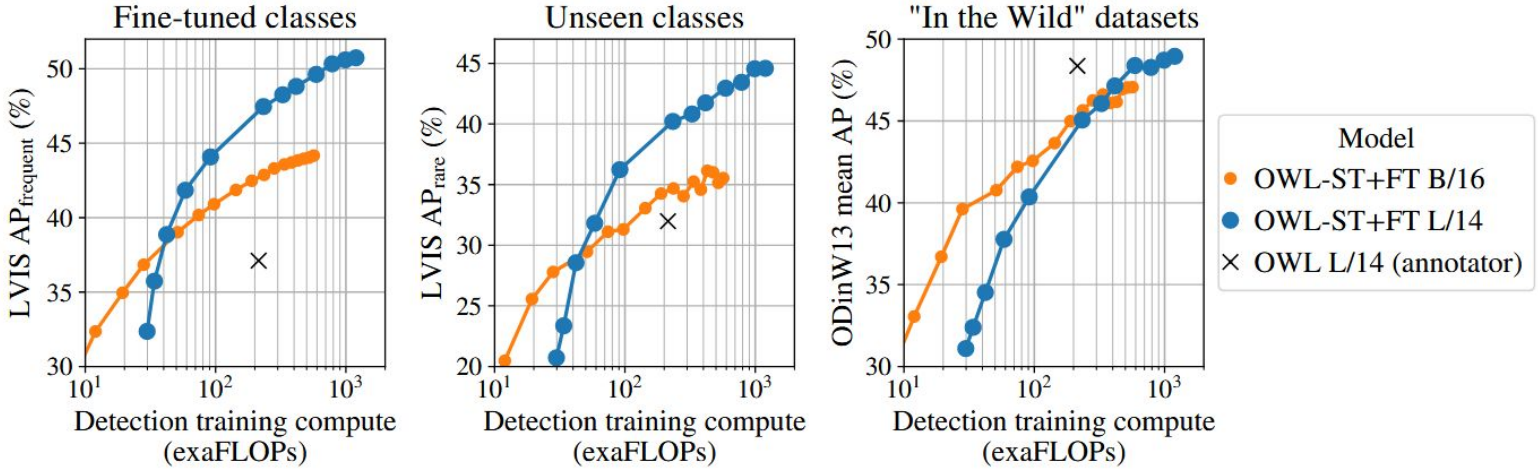


Figure 4: Scaling of detection performance with model size and training compute. Models show classic scaling behavior [36]: Performance increases monotonically with training compute, with larger models being necessary to benefit from larger amounts of compute/data. Models were self-trained on N-gram pseudo-annotations and fine-tuned on $LVIS_{base}$.

- **Experiments**

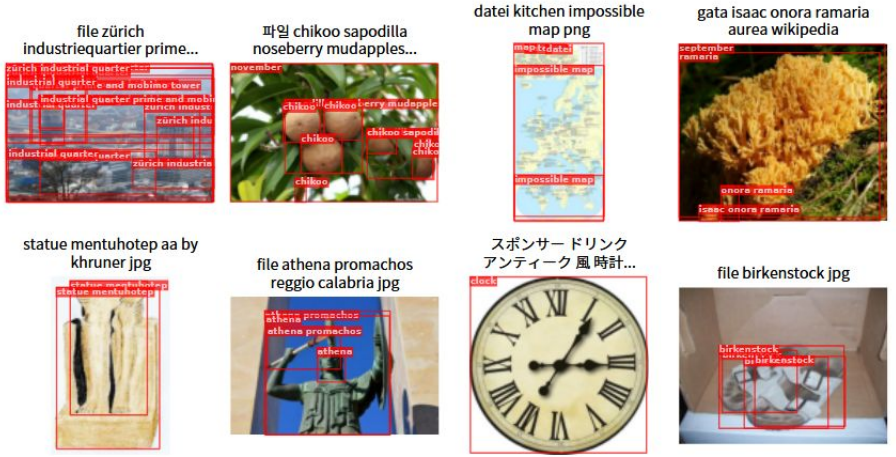


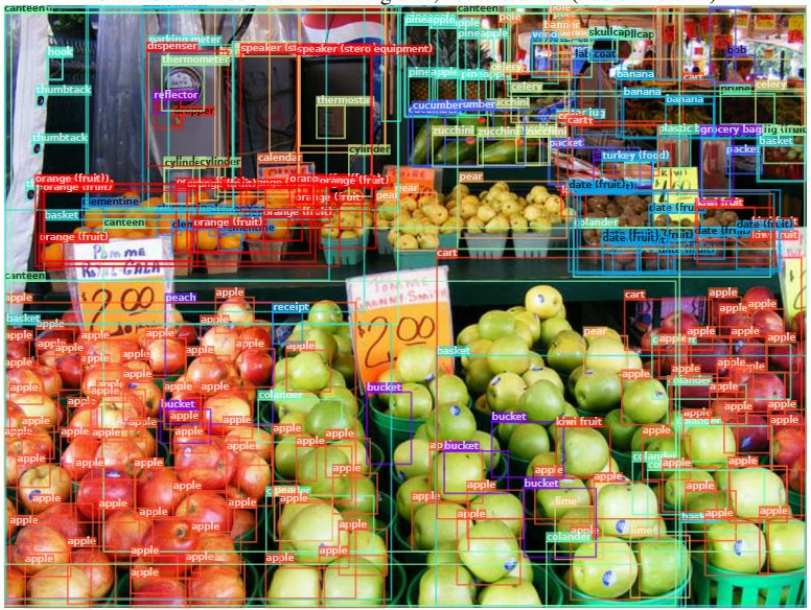
Figure A2: Example pseudo-annotations on WebLI [3]. Image-associated text (from the HTML `alt_text` tag) is shown above the images. If the text is not in English, an automatically generated translation is used. N-grams are extracted from these texts to generate queries for the annotator model. Pseudo-annotations were filtered as for our main experiments: To be included, boxes must have a score of at least 0.1, and images must have at least one box with a score above 0.3. All images from Wikimedia Commons.

- **Experiments**

Figure A3: Training inputs after pre-processing. **Top:** A 4×4 mosaic of randomly resized and padded images as used for self-training. **Bottom:** The same mosaic after dropping the 50% of patches with lowest pixel variance (image size: 1008×1008; patch size: 14×14). Most dropped patches belong to padding areas or uniform image backgrounds. All images from Wikimedia Commons.

- **Experiments**

OWL-ST L/14 self-trained on N-grams, not fine-tuned (Table 1 row 12)



OWL-ST+FT L/14 self-trained on N-grams and fine-tuned on $\text{LVIS}_{\text{base}}$ (Table 1 row 15)

Figure A4: Qualitative example for OWLv2 L/14 from the LVIS val set. For the visualization, all LVIS classes were used as prompts. $\text{LVIS}_{\text{rare}}$ classes are labeled in black. **Top:** OWL-ST self-trained on N-grams, not fine-tuned (Table 1 row 12). **Bottom:** OWL-ST+FT self-trained on N-grams and fine-tuned on $\text{LVIS}_{\text{base}}$ (Table 1 row 15). Boxes above score 0.08 (top) or 0.3 (bottom) are shown.