Exploiting Unlabeled Data with Vision and Language Models for Object Detection Supplementary Document

Shiyu Zhao[1,⋆], Zhixing Zhang[1,⋆], Samuel Schulter[2], Long Zhao[3], Vijay Kumar B.G[2], Anastasis Stathopoulos[1], Manmohan Chandraker[2,4], and Dimitris Metaxas[1]

[1]Rutgers University, [2]NEC Labs America, [3]Google Research, [4]UC San Diego

The supplemental material first provides additional experiments on open-vocabulary detection (OVD) on the LVIS dataset in Sec. A.1, a faster version of VL-PLM to speed-up pseudo label (PL) extraction in Sec. A.2, and additional experiments on semi-supervised detection (SSOD) in Sec. A.3. Then, we give additional analysis on fusing pseudo labels in SSOD in Sec. B.1, the quality of PLs in Sec. B.2, how to model the background category in PL generation in Sec. B.3, and the generalization ability of the proposal generator in Sec. B.4. Finally, qualitative results of PLs and the final OVD detector are given in Sec. C.

A Additional Experiments

A.1 Open-vocabulary detection results on LVIS

In addition to our open-vocabulary detection (OVD) experiments on COCO [6] in Sec. 4.1 of the main paper, we also evaluate our model on the LVIS [4] dataset. LVIS is a large-vocabulary dataset with 1203 categories and shares images with COCO [6]. We follow the experimental setup of ViLD [3], the state-of-the-art method on LVIS for OVD (LVIS-OVD): All categories are divided into three sets, namely, frequent, common, and rare, based on the numbers of their objects. Following [3], we take frequent and common categories as the base categories and regard rare categories as the novel categories. We leverage base categories to train our two-stage class-agnostic proposal generator and adopt VL-PLM to generate PLs for novel categories. Then, a standard OVD detector was trained with both the ground truth of base categories and our PLs.

Comparison with ViLD: Table 1 compares our detector via VL-PLM with *Supervised* and the state-of-the-art method ViLD. *Supervised* is the supervised baseline model trained on the whole LVIS with repeat factor sampling [4,7]. We report the box mAP to better indicate the performance on detection. For ViLD [3], we took the model provided by the authors and re-ran the evaluation to ensure a fair comparison. For the *Supervised* baseline, we adopted the numbers from [3]. As shown, our method outperforms ViLD on all splits. We gain +0.6 AP_r for rare categories (novel) and +2.6 AP_c/+3.5 AP_f for common/frequent

⋆ Equal contribution.

Table 1. Evaluations for open vocabulary detection on LVIS-v1 [4].

Method	Training data	AP_r	AP_c	AP_f	mAP
Supervised	Base + Novel	12.3	**24.3**	32.4	25.4
ViLD [3]	Base	16.6	21.1	31.6	24.4
VL-PLM (Ours)	Base	**17.2**	23.7	**35.1**	**27.0**

categories (base). This indicates that training with our PLs has less influence on base categories than the distillation in ViLD does. We observed a similar trend on COCO [6] in Sec. 4.1. Compared with the improvement on base categories, the improvement on novel categories is relatively small, likely due to the long-tailed distribution of novel categories. Still, VL-PLM outperforms *Supervised* by a large margin in terms of AP_r. A possible explanation is that our PLs provide more annotations for rare categories. In general, our PLs provide more (but noisy) annotation than the grounded truth "federated" annotations of LVIS [4], where only subsets of categories are annotated per image. This may explain why VL-PLM even outperforms *Supervised* in mAP. Although those annotations are not fully accurate, they still provide useful information for rare categories in the training, e.g., the texture of objects of rare categories.

A.2 Fast VL-PLM and Multi-scale Fast VL-PLM

This section provides more details about Fast VL-PLM and Multi-scale Fast VL-PLM that are mentioned in the discussion of **Time efficiency** in Sec. 4.3 of the main paper. Table 2 compares original VL-PLM with the two variants in terms of time cost and pseudo label quality. As shown, Fast VL-PLM reduces runtime by 5× with a slight drop in PL quality. Multi-scale Fast VL-PLM almost entirely removes the accuracy drop and still reduces runtime by 3×. Our Fast VL-PLM and Multi-scale Fast VL-PLM are only designed for ResNet-based CLIP not for ViT-based CLIP, which are described as follows.

Table 2. Average time to get pseudo labels per image and their quality.

	CLIP Backbone	Time (s)	AP@PL	#@PL
Original VL-PLM	ResNet50	0.5413	15.9	7.31
Fast VL-PLM	ResNet50	0.1199	13.5	7.39
Multiscale Fast VL-PLM	ResNet50	0.1685	15.4	7.38

For Fast VL-PLM, we feed the whole input image into CLIP's ResNet50 to get shared feature maps. Then, we bound the coordinates of each region proposal to the close integer. For example, a proposal of $\{10.9, 50.2, 110.1, 100.9\}$ in xyxy format is converted into a box of $\{11, 50, 110, 101\}$. Third, based on the bounded

box, we crop features on the shared feature maps for the corresponding proposal. Finally, we ignore the positional embedding and feed the cropped features into the last attention layer of CLIP to output the region embedding for each proposal. Please refer to [8] for details on the structure of ResNet-based CLIP. Please note the difference to ROI-pooling, where each cropped region would be pooled into the same spatial dimensions. Here, the cropped feature size is proportional to the bounding box and we let the attention layer in CLIP "pool" the input into a fixed-size output. We tried ROI-pooling but observed worse performance.

Multi-scale Fast VL-PLM is a mutli-scale version of Fast VL-PLM. We first construct an image pyramid and feed those images into CLIP's ResNet50 to get shared multi-scale feature maps. Then, for region proposals of small size, we crop features on shared feature maps of a large scale so that more details are attained in the cropped feature maps. Shared feature maps of a small scale are for region proposals of large size. Specially, we resize the smallest dimension of input images into three scales, i.e., 224, $224 \cdot 3 = 672$, and $224 \cdot 5 = 1120$. Thus, the shared feature maps are in one scale among 7, $7 \cdot 3 = 21$, and $7 \cdot 5 = 35$. Region proposals are assigned to different scales by their areas. The area > 64 is for the first scale, the area between 16 and 64 for the second, and others for the third. Our design is inspired by FPN [5] and enjoys its advantages, as well.

A.3 Scaling up unlabeled images for SSOD

To better understand the impact of the ratio between labeled and unlabeled images, we continue our experiments on semi-supervised object detection (SSOD). In Sec. 4.2 of the main paper, we varied the fraction of labeled images. Here, we use a fixed amount of labeled images and vary the number of unlabeled images. We train Faster R-CNN models using 5% of labeled COCO images with different amounts of unlabeled images. We randomly select our unlabeled images from the unlabeled images provided by the COCO dataset [6]. All models are trained for 90k iterations with a batch size of 16. As shown in Table 3, as the amount of unlabeled data increases, the performance increases as well, but with diminishing returns. In future work, we want to explore this aspect more and evaluate PLs from VL-PLM in an omni-supervised setting [9].

Table 3. Detection accuracy in mAP for Faster R-CNN on 5% of labeled COCO images with varied amounts of unlabeled images.

# of labeled images	5764	5764	5764	5764	5764
# of unlabeled images	0	5764	28820	57640	115280
Ratio	1:0	1:1	1:5	1:10	1:20
mAP	17.7	21.1	22.7	23.1	23.7
mAP increase	-	+3.4	+5.0	+5.4	+6.0

Table 4. Pseudo label fusion for semi-supervised object detection on COCO 2017 [6].

Methods	1% COCO	2% COCO	5% COCO	10% COCO
SSL PLs only	11.18	14.88	21.20	25.98
VL-PLM w/o fusion	13.27	15.97	20.64	24.20
VL-PLM	**15.35**	**18.60**	**23.70**	**27.23**

B Additional Analysis and Discussion

B.1 Fusing Pseudo Labels from SSOD teacher with VL-PLM

This section provides more details for Sec. 3.3 of the main paper on how we merge PLs for SSOD. As illustrated in Fig. 1, we use the proposed VL-PLM to generate PLs and merge the PLs from the semi-supervised teacher. Then, we apply thresholding and NMS on the merged PLs to obtain the final PLs for SSOD. To validate the effectiveness of this fusion strategy, we consider the following baselines. (1) SSL PLs only: We only adopt the PLs from the semi-supervised teacher as the final PLs. (2) VL-PLM w/o fusion: We only pick the PLs from the vision and language model. (3) VL-PLM: The fused PLs. We base our experiments on 1%, 2%, 5% and 10% COCO splits [6,10] for SSOD and report the results in Table 4. As shown, compared with SSL PLs only, VL-PLM w/o fusion is better on 1% and 2% COCO splits but worse on 5% and 10% COCO splits. A possible explanation is that V&L models provide more useful information that boosts the performance when the amount of annotated data is smaller. Moreover, VL-PLM outperforms SSL PLs only and VL-PLM w/o fusion on all splits. This clearly demonstrates that our PLs from VL-PLM are better than the PLs from the teacher model. Our fusion method successfully improves the quality of the final PLs. Since putting PLs from the teacher and V&L model together brings about the best results even for 5% and 10% COCO splits, we believe that PLs from the teacher and the V&L model are complementary.

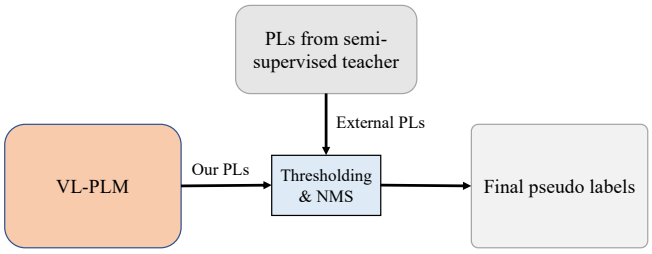


Fig. 1. Overview of pseudo labels (PLs) fusion for semi-supervised object detection (SSOD). We fuse our PLs with those from the semi-supervised teacher model before the thresholding and NMS.

Table 5. Relationship between the quality of pseudo labels and the performance of the final open vocabulary detectors on COCO 2017 [6].

	PL Setting	Pseudo Labels		Final Detector		
		AP@PL	#@PL	Base AP	Novel AP	Overall AP
PL v1	No RoI, $\tau = 0.05$	17.4	89.92	33.3	14.6	28.4
PL v2	No RoI, $\tau = 0.95$	14.6	2.88	56.1	26.0	48.2
PL v3	VL-PLM, $\tau = 0.05$	20.6	85.15	29.7	19.3	27.0
PL v4	VL-PLM, $\tau = 0.95$	18.0	2.93	55.4	**31.3**	**49.1**
PL v5	VL-PLM, $\tau = 0.99$	11.1	1.62	**56.7**	27.2	49.0

B.2 Analysis on the Quality of PLs

In this section, we provide a more detailed analysis and discussion for **Understanding the quality of PLs** of Sec. 4.3 in the main paper.

Quality of PLs and performance of final detectors: In this section, we provide more analysis for Table 5 which is also present in the main paper. We recall our 5 baselines as follows, (1) *PL v1*: We take the raw region proposals from region proposal network (RPN) without RoI refinement in our pseudo label generation and set $\tau = 0.05$. (2) *PL v2*: The same as *PL v1* but with $\tau = 0.95$. (3) *PL v3*: VL-PLM with $\tau = 0.05$. (4) *PL v4*: VL-PLM with $\tau = 0.95$. (5) *PL v5*: VL-PLM with $\tau = 0.99$. In Table 5, the evaluations are conducted on the zero-shot splits [1] on COCO [6] (COCO-ZS). We report the AP_{50} (AP@PL) and the number (#@PL) on novel categories for different PLs with the performance of detection models trained with corresponding PLs. Novel AP, Base AP, and Overall AP are provided to indicate the performance of detectors.

Based on Table 5, we have the following findings. First, compared with *PL v4*, *PL v1* shares similar AP@PL but has much more pseudo annotations. The final detector of *PL v4* significantly outperforms that of *PL v1*. Second, compared with *PL v4*, *PL v2* has nearly the same amount of PLs with a lower AP@PL. In terms of Novel AP, the final detector of *PL v4* outperforms that of *PL v1* by a large margin. Based on those facts, we conclude that neither AP@PL nor #@PL alone can decide the quality of PLs. We need to consider both AP@PL and #@PL. Good PLs come with high AP@PL and low #@PL. Third, based on the comparison between *PL v4* and *PL v5*, we find that an extremely high threshold τ harms the predictions of novel categories but results in a slightly better performance on base categories. Empirically, we find a reasonable $\tau \in [0.6, 0.95]$ and set $\tau = 0.8$ as default. Fourth, comparing *PL v2* and *PL v4*, we find that with RoI head refinement, our PLs gain a significant improvement. For the final detector, *PL v4* achieves similar performance on base categories as *PL v2* and much better results on novel categories, which clearly demonstrate the effectiveness of using RoI head as box refinement.

Table 6. The quality of pseudo labels generated with different ways to model the background. τ is tuned to keep similar #@PL. See text for more details.

	Novel	Novel+BG	Novel+Base	Novel+Base+BG	Novel+OV set
τ	0.80	0.80	0.59	0.59	0.501
AP@PL	25.5	25.7	26.3	26.4	25.1
#@PL	4.86	4.18	4.36	4.22	4.35

B.3 Modeling Background in PL Generation

Background is a latent category for the detection task and should be considered in our pseudo label generation, as well. In this section, we demonstrate how different ways of modeling the background affects the quality of PLs for OVD on COCO-ZS. Since there may be region proposals for base categories, we generalize the concept of background as categories that are not in the target categories, and consider 5 category spaces with different backgrounds as

1. *Novel*: The label space for pseudo labels only contains novel categories, no explicit modeling of background
2. *Novel+BG* The text "background" is used as one additional background category
3. *Novel+Base*: Both novel and base categories are used in the label space of PLs, where base categories should model the background (since those are annotated in the OVD setting)
4. *Novel+Base+BG*: Same as *Novel+Base*, but with the additional category of "background"
5. *Novel+OV*: We remove novel categories in COCO from the 1203 categories in LVIS [4]. The remaining categories are used to model background. We name this background set as OV.

Table 6 provides AP@PL and #@PL on novel categories for different category spaces of the background.

As shown, *Novel+BG* is slightly better than *Novel* with higher AP@PL and lower #@PL. *Novel+Base* and *Novel+Base+BG* result in the same observation. BG does improve the quality but the improvement is not significant. Moreover, *Novel+Base* gains a clear improvement over *Novel*, likely because it helps V&L models to identify objects of base categories which will be removed as the background, improving the quality of PLs for novel categories. Third, OV as the background decreases the quality of PLs based on the comparison between*Novel+OV* and *Novel+Base*. This is reasonable because V&L models may be influenced by the large amount of OV categories that are absent in the scene.

B.4 Generalization ability of the proposal generator

For OVD, we need to identify the objects of novel categories in the unlabeled data. Similar to [3], we study if the two-stage class-agnostic detector trained

Fig. 2. Visualization of pseudo labels. Only novel categories in the image are shown. Top: Good cases with multiple instances. Bottom: Failure cases with missing instances, grouped instances, part domination and redundant annotations.

on base categories generalizes to novel ones. Basing our experiments on COCO and LVIS datasets, we train a Faster R-CNN with either base or all categories (base+novel). Table 7 presents the top-N average recall (AR@N) of the RPN on novel categories. As shown, on COCO-ZS, RPN trained without novel categories suffers a clear performance drop. But it can still achieve a reasonable AR@1000, and we adopt top 1000 boxes from RPN in our pseudo label generation. On the contrary, models on LVIS achieve nearly the same average recall (AR) on novel categories with either base categories or base + novel categories as the training data. Possibly, there are more categories and instances in base categories of LVIS than in those of COCO. Thus, the proposal generator is able to learn a better concept about the objects in LVIS.

Table 7. The generalization ability of RPN of the proposal generator on COCO 2017 [6] and LVIS-v1 [4]. We report the average recall (AR) on novel categories.

Dataset	Training Data	AR@100	AR@300	AR@500	AR@1000
COCO	Base	34.5	43.4	47.2	51.7
	Base + Novel	54.2	59.3	61.1	62.8
LVIS	Base	33.3	42.3	45.9	50.5
	Base + Novel	33.7	43.0	46.6	50.5

C Qualitative Results

C.1 Visualization of Our Pseudo Labels

We illustrate good cases and failure cases of our PLs for OVD on COCO in Fig. 2 and Fig. 3. We only visualize boxes for target (novel) categories that are

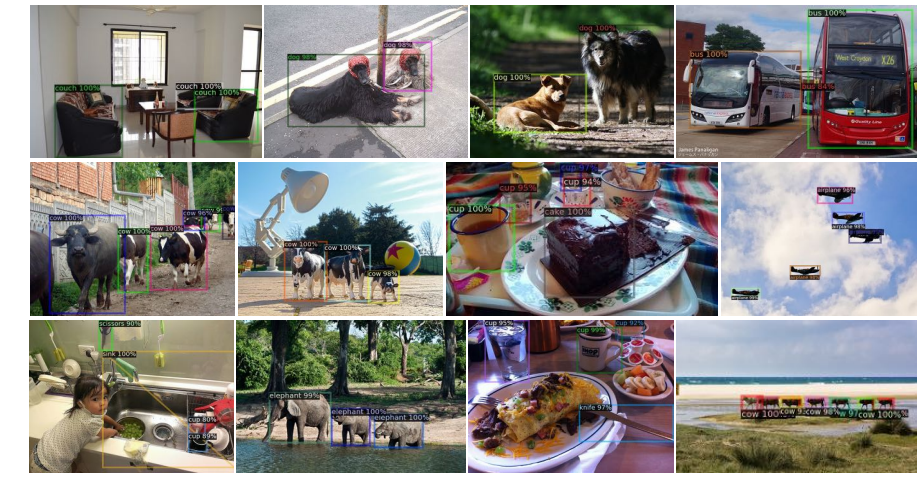
included in the scene. Good cases show that VL-PLM is able to locate multiple objects correctly. However, in the recent caption-based pseudo label generation method [2], it's a major issue to find multiple objects of the same categories. For failure cases, there are four major types, i.e., part domination, redundant boxes, missing instances, and grouped instances. We believe that part domination and redundant boxes are mainly caused by the poor localization ability of the adopted V&L model CLIP [8]. Missing and grouped instances usually happen when multiple instances are close to each other or only part of instances appears. Possibly, the major reason is that the proposal generator cannot provide correct region proposals, leading to poor quality of PLs. In this sense, an improvement on either the V&L models or the proposal generator in VL-PLM will boost PLs' quality.

C.2 Visualization of Our OVD Detector

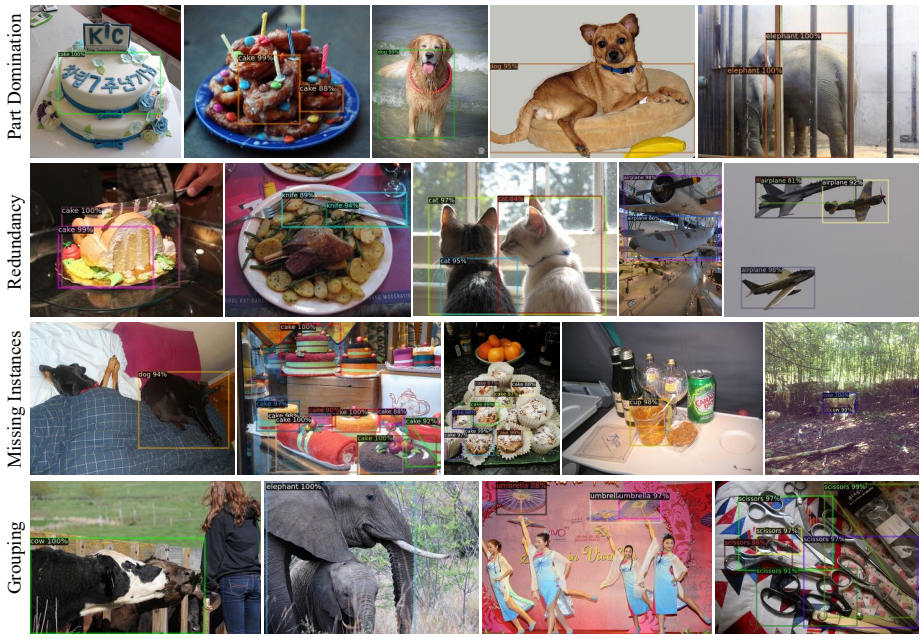
This section visualizes the good and failure cases of the final detector for OVD. Fig. 4 illustrates those cases on novel and base categories, respectively. As shown, the detector trained with our PLs is able to detect objects of novel categories. Moreover, unlike PLs, the results of the final detector mainly include three types of failure cases, i.e., missing instances, redundant boxes, and grouped instances. Possibly, the part domination that degrades the quality of PLs is reduced during the training with both ground truth and PLs.

References

1. Bansal, A., Sikka, K., Sharma, G., Chellappa, R., Divakaran, A.: Zero-shot object detection. In: ECCV. pp. 384–400 (2018)
2. Gao, M., Xing, C., Niebles, J.C., Li, J., Xu, R., Liu, W., Xiong, C.: Towards open vocabulary object detection without human-provided bounding boxes (2021)
3. Gu, X., Lin, T.Y., Kuo, W., Cui, Y.: Open-vocabulary Object Detection via Vision and Language Knowledge Distillation. In: ICLR (2022)
4. Gupta, A., Dollár, P., Girshick, R.: LVIS: A dataset for large vocabulary instance segmentation. In: CVPR (2019)
5. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature Pyramid Networks for Object Detection. In: CVPR (2017)
6. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common Objects in Context. In: ECCV (2014)
7. Mahajan, D., Girshick, R., Ramanathan, V., He, K., Paluri, M., Li, Y., Bharambe, A., van der Maaten, L.: Exploring the Limits of Weakly Supervised Pretraining. In: ECCV (2018)
8. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: ICML (2021)
9. Radosavovic, I., Dollár, P., Girshick, R., Gkioxari, G., He, K.: Data Distillation: Towards Omni-Supervised Learning. In: CVPR (2018)
10. Sohn, K., Zhang, Z., Li, C.L., Zhang, H., Lee, C.Y., Pfister, T.: A simple semi-supervised learning framework for object detection. In: arXiv:2005.04757 (2020)



(a)



(b)

Fig. 3. Visualizations of the pseudo labels (PLs) from VL-PLM. Only boxes for target categories in the scene are shown. (a) Good cases. All target objects are located with appropriate boxes. (b) The most common types of failure cases in our PLs, i.e., part domination, redundant boxes, missing instances, and grouped instances.

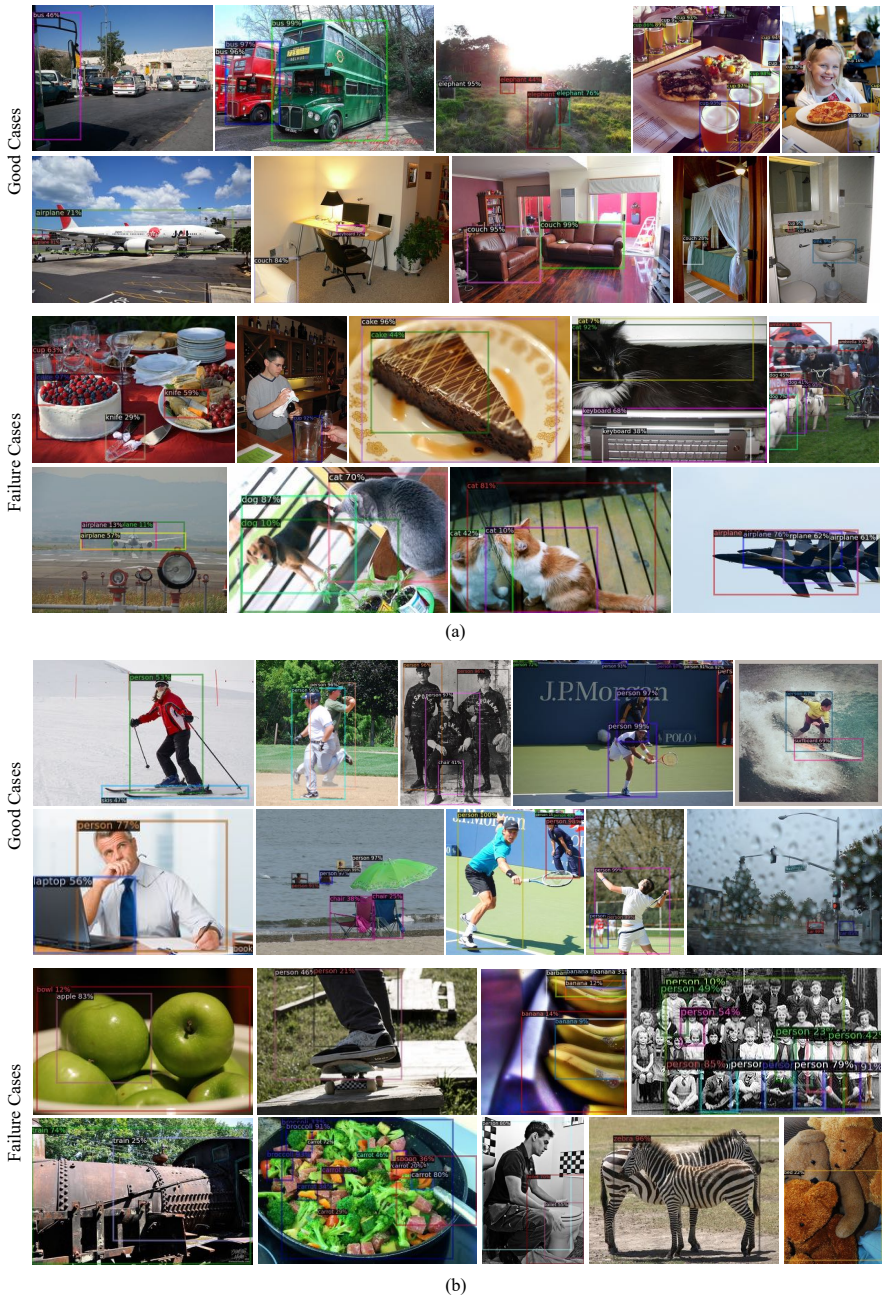


Fig. 4. Visualization of the final detection results. Only boxes for target categories in the scene are shown. (a) Novel categories as the target. (b) Base categories as the target. The major failure cases belong to three types, i.e., missing instances, redundant boxes, or grouped instances.