Solving Instance Detection from an Open-World Perspective

Supplementary Material

Outline

In this supplemental material, we provide additional experimental results, more detection visualizations, and open-source code. Below is the outline of this document.

- **Section A**. We provide demo code for our proposed method IDOW in both CID and NID using Jupyter Notebook.
- **Section B**. We conduct additional ablation studies of each strategy involved in our IDOW.
- **Section C**. We provide additional qualitative visualizations of detection results from different methods.

A. Open-Source Code

In the project page (https://shenqq377.github.io/IDOW/), we release open-source code in the form of Jupyter Notebook plus Python files.

Why Jupyter Notebook? We prefer to release the code using Jupyter Notebook (https://jupyter.org) because it allows for interactive demonstration for education purposes. In case the reader would like to run Python script, using the following command can convert a Jupyter Notebook file `xxx.ipynb` into a Python script file `xxx.py`:

```
jupyter nbconvert --to script xxx.ipynb
```

Requirement. Running our code requires some common packages. We installed Python and most packages through Anaconda. A few other packages might not be installed automatically, such as Pandas, torchvision, and PyTorch, which are required to run our code. Below are the versions of Python and PyTorch used in our work.

- Python version: 3.9.16 [GCC 7.5.0]
- PyTorch version: 2.0.0

We suggest assigning >30GB space to run all the files.

License. We release open-source code under the MIT License to foster future research in this field.

Demo. The Jupyter Notebook files below demonstrate our proposed method IDOW in both CID and NID settings. During the training stage, we finetune DINOv2 with visual references from HR-InsDet/RoboTools dataset (in the CID setting) or OWID dataset (in the NID setting). We use GroundingDINO to detect instance-agnostic proposals for a given testing image. We feed these proposals into DINOv2 for feature representation, just like how we represent visual references of object instances. Over the features, we use stable matching on the cosine similarities between proposals and visual references to find the best match, yielding the final detection results.

- `demo_CID_InsDet.ipynb`
  Running this file finetune DINOv2 with visual references of 100 object instances from HR-InsDet dataset, and compute feature representation for GroundingDINO-detected proposals and visual references. The final detection is still performed using the stable matching algorithm with cosine similarities. This file presents the proposed approach $\text{IDOW}_{\text{GroundingDINO}}$ in the CID setting.
- `demo_NID_RoboTools.ipynb`
  Running this file finetune DINOv2 with visual references of 9691 object instances from OWID dataset, and compute feature representation for GroundingDINO-detected proposals and visual references. The final detection is still performed using the stable matching algorithm with cosine similarities. This file presents the proposed approach $\text{IDOW}_{\text{GroundingDINO}}$ in the NID setting.

B. Ablation Study and Further Analysis

We include additional ablation studies to supplement the results in the main paper. Specifically, we study: (1) performance of IDOW w.r.t different loss functions; (2) correlation plot between open-world sampled distractors and visual references; (3) performance of IDOW w.r.t different model backbones. Lastly, we present quantitative results evaluated by average recall (AR) and precision-recall curves of different methods to better understand the performance improvements of our proposed approach.

Performance of IDOW w.r.t different loss functions. There are various loss functions in deep learning to learn effective feature representations, each designed with a particular goal, e.g., learn features that discriminate between classes by cross entropy (CE) loss [42], structure the feature space based on similarity and dissimilarity by contrastive loss [7, 52], etc. We ablate three choices of finetuning FMs in IDOW, specifically CE loss, contrastive loss and triplet loss, in this supplement. To keep the comparison fair, we keep all the settings same except the loss function. As shown in Table 4, we find that adapting FMs with triplet loss performs better than finetuning the model with either a contrastive loss or a cross-entropy loss in both CID and NID settings, e.g., 55.52 AP by CE loss vs. 56.01 AP by triplet loss in NID. This could be attributed to the advantages of triplet loss in handling hard examples in IDOW.

Correlation plot between open-world sampled distractors and visual references. The experimental results in the main paper show that leveraging open-world sampled distrators leads to performance improvements. However, it is less clear how these distractors correlate with visual

Table 4. **Comparisons with different losses in foundation model adaptation.** We carry out the study on the HR-InsDet dataset in both CID and NID settings. Clearly, adapting FMs with triplet loss performs better than finetuning the model either by a contrastive loss or a cross entropy (CE) loss. This could be attributed to the advantages of triplet loss in handling hard examples in IDOW.

Setting	Loss	AP						AP_{50}	AP_{75}
		avg	hard	easy	small	medium	large		
CID	CE Loss	55.72	39.12	63.02	34.16	60.66	72.23	67.70	61.41
	Contrastive Loss	53.94	38.65	60.89	32.11	58.86	71.07	65.46	59.34
	Triplet Loss	57.01	40.74	64.36	35.25	62.98	73.64	69.33	62.84
NID	CE Loss	55.52	39.94	62.58	34.09	61.41	72.83	67.30	61.19
	Contrastive Loss	55.23	39.64	62.38	33.75	61.38	71.93	67.19	60.91
	Triplet Loss	56.01	40.42	63.36	35.14	62.22	72.55	68.11	61.75

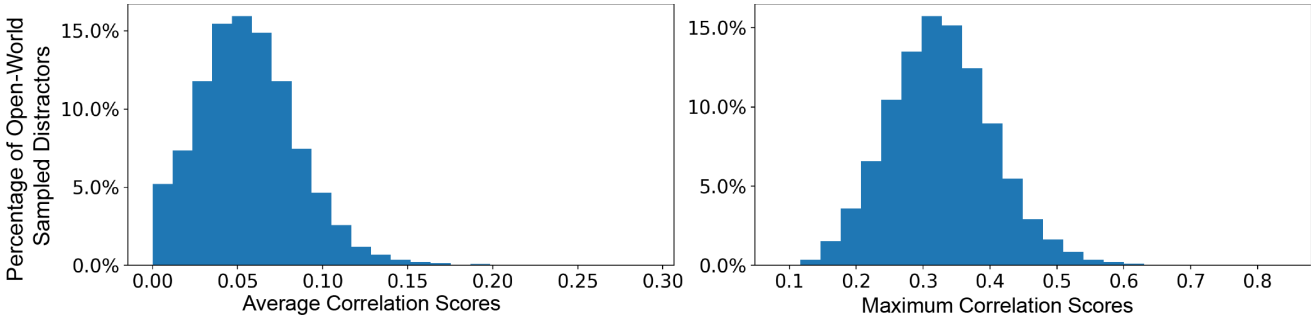

Figure 8. **Correlation plot between open-world sampled distractors and visual references.** We plot the correlation measured by average (left) and maximum (right) cosine similarity between sampled distractors and visual references on the HR-InsDet dataset. The average score measures the average distance between each distractor and all visual references while the maximum score finds the highest score. We find the majority of distractors have high enough maximum correction scores (≥ 0.3), indicating open-world sampled distractors are "hard" for FMs to distinguish from at least one instance visual reference. This observation, together with our hard example sampling strategy, demonstrates the effectiveness of approach InsDet with open-world sampled distractors.

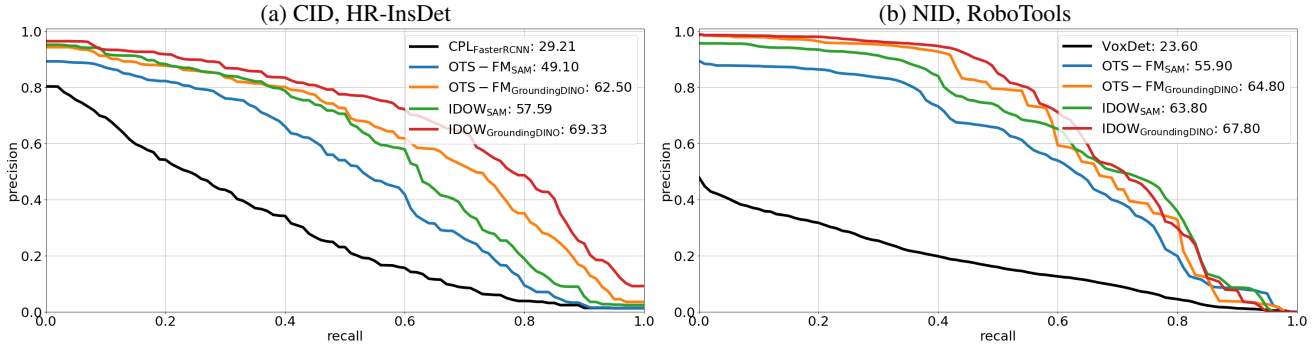

Figure 9. **Precision-recall curves with IoU=0.5 (AP_{50}) under both CID and NID settings.** We find that (1) our proposed method IDOW outperforms the previous state-of-the-art OTS-FM under both settings in terms of both recall and precision, and (2) IDOW maintains more stable precision accuracy when increasing recall (e.g., IDOW$_{GroundingDINO}$ v.s. OTS-FM$_{GroundingDINO}$), indicating that it has higher stability and stronger robustness in distinguishing between positive and negative samples.

references and thus contribute to better adapt FMs for InsDet. We plot the correlations measured by cosine similarities between distractors and visual references. Specifically, we report average and maximum cosine similarity scores. The former one describes the average score of each distractor w.r.t all visual references while the later one represents the maximum score. As shown in Fig. 8, the majority of distractors have maximum correction scores ≥ 0.3 despite the average score is low. This observation suggests that the majority of distractors are "hard" for FMs to distinguish from at least one instance. This explains adding open-world sampled distractors improve the performance together with our hard example sampling strategy. Nonetheless, it is worth for future work to explore how to better extract and leverage

Table 5. **Ablation study of different DINOv2 backbone involved in our IDOW**. We carry out the study in the CID setting on the HR-InsDet dataset. We compare different input resolutions and DINOv2 backbones, and make two salient conclusions. First, stronger backbones lead to better performance. Second, the increasing input resolution boosts the detection performance.

DINOv2 backbone	# Param	Input size	AP						AP_{50}	AP_{75}
			avg	hard	easy	small	medium	large		
ViT-s/14	21M	224×224	54.00	36.49	62.48	33.10	61.18	72.80	65.18	59.84
ViT-s/14	21M	448×448	56.84	40.67	64.49	35.22	62.54	74.01	68.85	62.90
ViT-s/14	21M	518×518	57.01	40.74	64.36	35.25	62.98	73.64	69.33	62.84
ViT-b/14	86M	518×518	58.63	41.45	66.41	36.92	65.11	77.07	71.28	64.78

Table 6. **Benchmarking results w.r.t average recall (AR) for *small*, *medium* and *large* instances**. "AR@max10" means AR within the top-10 ranked detections. In computing AR, we rank detections by using the detection confidence scores of the learning-based methods (e.g., FasterRCNN) or similarity scores in the non-learned methods OTS-FM. AR_s, AR_m, and AR_l are breakdowns of AR for small, medium, and large testing object instances. Results show that (1) SAM or GroundingDINO generally recalls more instances than others, (2) methods with high AR typically achieve better AP (cf. Table 1), and (3) all methods suffer from small instances.

	AR@max10	AR@max100	AR_s@max100	AR_m@max100	AR_l@max100
$CPL_{FasterRCNN}$ [9, 45]	26.24	39.24	14.83	44.87	60.05
$CPL_{RetinaNet}$ [9, 27]	26.33	49.38	22.04	56.76	69.69
$CPL_{CenterNet}$ [9, 64]	23.55	44.72	17.84	52.03	64.58
CPL_{FCOS} [9, 54]	25.82	46.28	22.09	52.85	64.11
CPL_{DINO} [9, 59]	29.84	54.22	32.00	59.43	72.92
$OTS\text{-}FM_{SAM}$ [22, 48]	40.02	63.06	31.11	70.40	90.36
$IDOW_{GroundingDINO}$	40.29	77.09	53.53	83.73	94.06

these open-world distractors.

Choice of FM backbones and input resolutions in IDOW. The experiments present in the main paper demonstrate the effectiveness of IDOW using DINOv2 with ViT small (ViT-s) as the backbone. Here, we explore the performance change of IDOW by adopting different DINOv2 backbones. According to [38], authors increase the resolution of images to 518×518 during a short period at the end of pretraining. Additionally, input resolutions also correlates with the performance of FMs from the recent literature [48]. Therefore, we further explore the effect of increasing input resolutions in Table 5. We summarize two conclusions: *(1) Stronger backbones lead to better performance.* Comparing ViT-s and ViT-b, it is clear that adopting a deeper backbone achieves > 4 AP improvements. *(2) Input resolution matters and increasing resolution boosts the performance.* Comparing the ViT-s backbone with 224, 448 and 512 input resolutions, inputs with 512×512 shows $\sim$3 AP improvements over 224×224.

Quantitative results w.r.t average recall (AR). In addition to the experimental results of average precision (AP) in the main paper, we also report the average recall (AR) of different competitors as a supplement in Table 6. We carry out the experiments in the CID setting on HR-InsDet dataset. From Table 6, we make three conclusions. (1) Non-learned FMs are better at proposing instances, e.g., 40.29 AR in GroundingDINO vs. 26.24 AR in FasterRCNN. (2) Comparing with AP in Table 1, methods with high AR typically achieve better AP. (3) All methods suffer from small instances.

Precision-recall curve of different methods. We present the precision-recall curves of different competitors in the CID setting on the HR-InsDet dataset (ref. Table 1) and in the NID setting on the RoboTools dataset (ref. Table 2) in Fig. 9. We make two observations. (1) Our proposed method $IDOW_{GroundingDINO}$ outperforms the previous state-of-the-arts in terms of both recall and precision. (2) IDOW maintains more stable precision accuracy when increasing recall (e.g., $IDOW_{GroundingDINO}$ v.s. $OTS\text{-}FM_{GroundingDINO}$), indicating that it has higher stability and stronger robustness in distinguishing between positive and negative samples.

Runtime/Inference speed. We provide inference time of our $IDOW_{GroundingDINO}$ averaged over HR-InsDet / RoboTools testing images w.r.t each step in Table 7: proposal detection using GroundingDINO, feature extraction with DINOv2, and feature matching including proposal-reference similarity computation and stable matching. In HR-InsDet / RoboTools, testing image resolution is 1024x2048 / 1080x1920; each instance has 24 / 100 visual references, with GroundingDINO generating 40 / 35 proposals per image on average. Our method is quite efficient in inference. Moreover, our hard example mining uses a simple min operation to focus on the hardest example in each training batch for each anchor. This min operation does not add additional

Table 7. **Runtime of IDOW averaged over HR-InsDet / RoboTools testing images**.

Dataset	proposal det.	fea. extraction	fea. matching	Total
HR-InsDet	0.123	0.018	0.042	0.183
RoboTools	0.125	0.019	0.018	0.162

compute cost.

C. Additional Visualizations

Prediction visualizations. We present more detection comparisons under the CID setting on the HR-InsDet dataset in Fig. 10, and under the NID setting on the RoboTools dataset in Fig. 11. We attach instance IDs to ground-truth and predictions to highlight whether the instance recognition is correct compared to the visual references. Our proposed method IDOW is compared with two previous arts, Cut-Paste-Learn [9] and OTS-FM [48]. We observe that our proposed method is more robust under small size, similar appearance, pose variation and serve occlusion by embracing foundation models with NeRF augmentation. For example, in Fig. 10, although instances (No.3, No.13, No.44 in the top row, and No.9, No.14, No.20, No.21, No.36, No.37 in the bottom row) in HR-InsDet are partially distracted, our IDOW can accurately identify them. In the RoboTools benchmark, most instances are very small and placed with arbitrary pose variations in the cluttered scenes. Moreover, the testing scene images are blurred. From Fig. 11, we observe that our IDOW can accurately detect more instances than VoxDet and OTS-FM although the visual references are not seen during adaptation in the NID setting.

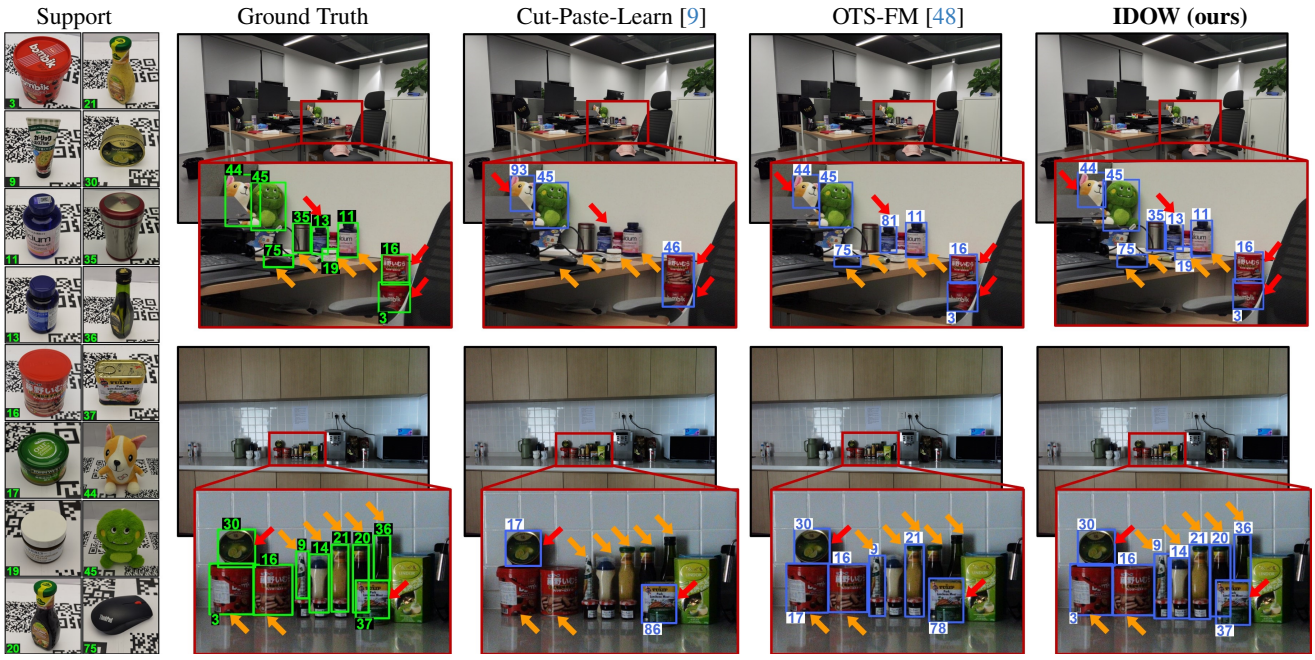

Figure 10. **Visual comparison of InsDet results by different methods in the CID setting** on HR-InsDet. We mark the ground-truth and predictions using green and blue boxes, respectively. Compared with Cut-Paste-Learn and OTS-FM, our IDOW detects more instances (see orange arrows) with better accuracy (see red arrows), when instances are partially occluded and illumination conditions change. Compared with OTS-FM, IDOW adapting a foundation model (i.e., DINOv2 used by both) yields better features, enabling robust predictions in InsDet.

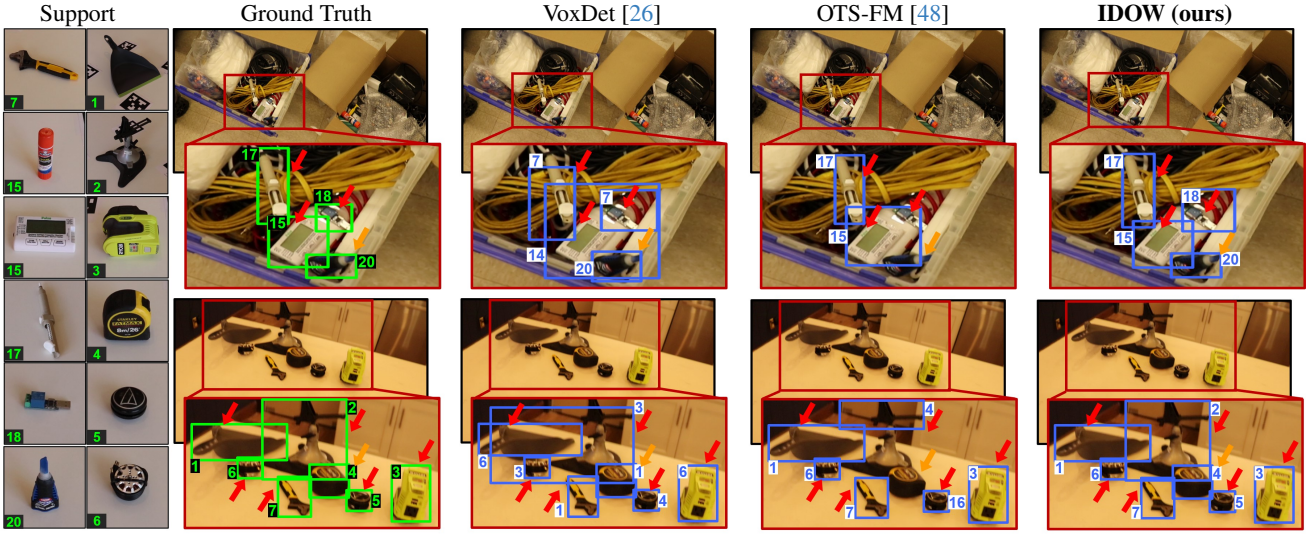

Figure 11. **Visual comparison of InsDet results by different methods in the NID setting** on RoboTools. We mark the ground-truth and predictions using green and blue boxes, respectively. Compared with VoxDet, our IDOW accurately detect instances (see red arrows) in a blurred and cluttered testing scene. Compared with OTS-FM, our IDOW detects more instances (see orange arrows).