Solving Instance Detection from an Open-World Perspective

Qianqian Shen[1,*] Yunhan Zhao[2,*] Nahyun Kwon[3] Jeeeun Kim[3] Yanan Li[4,†] Shu Kong[5,6,†]
[1]Zhejiang University, [2]UC Irvine, [3]Texas A&M University, [4]Zhejiang Lab
[5]University of Macau, [6]Institute of Collaborative Innovation
website and code: https://shenqq377.github.io/IDOW

Abstract

Instance detection (InsDet) aims to localize specific object instances within a novel scene imagery based on given visual references. Technically, it requires proposal detection to identify all possible object instances, followed by instance-level matching to pinpoint the ones of interest. Its open-world nature supports its broad applications from robotics to AR/VR but also presents significant challenges: methods must generalize to unknown testing data distributions because (1) the testing scene imagery is unseen during training, and (2) there are domain gaps between visual references and detected proposals. Existing methods tackle these challenges by synthesizing diverse training examples or utilizing off-the-shelf foundation models (FMs). However, they only partially capitalize the available open-world information. In contrast, we approach InsDet from an Open-World perspective, introducing our method IDOW. We find that, while pretrained FMs yield high recall in instance detection, they are not specifically optimized for instance-level feature matching. Therefore, we adapt pretrained FMs for improved instance-level matching using open-world data. Our approach incorporates metric learning along with novel data augmentations, which sample distractors as negative examples and synthesize novel-view instances to enrich the visual references. Extensive experiments demonstrate that our method significantly outperforms prior works, achieving >10 AP over previous results on two recently released challenging benchmark datasets in both conventional and novel instance detection settings.

1. Introduction

Instance detection (InsDet) aims to localize object instances of interest in novel scene imagery (Fig. 1), where these objects are specified by some visual references (as known as support templates) [3, 26, 34, 48, 53]. Its open-world nature supports its wide-ranging applications in robotics and AR/VR. For example, a robot can be commanded to search for a customer's suitcase at an airport [26], or *my-keys* in my cluttered bedroom [62]. Its open-world nature also presents significant challenges: during training, one has no knowledge of testing data distribution, which can be instantiated by the unknown scene imagery [9], novel object instances encountered only in testing [26], and domain gaps between visual references and detected proposals [48].

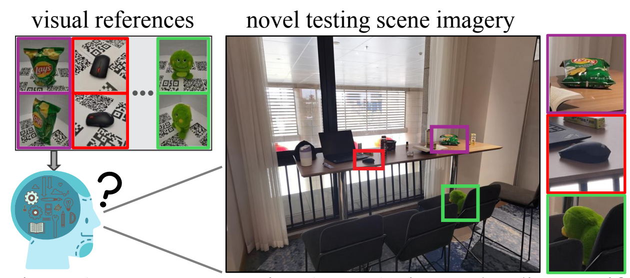

Figure 1. Instance Detection (InsDet) aims to localize specific object instances in novel scene imagery based on given visual references (*aka* support templates). It is a challenging problem due to its open-world nature: the testing scene images are unseen during training and thus are unknown to InsDet models, and visual references and detected proposals have domain gaps (e.g., due to occlusions and arbitrary lighting conditions in the latter).

Status quo. The literature of InsDet has two settings and existing InsDet methods attempt to address the open world in different aspects (Fig. 2). The *conventional instance detection (CID)* setting provides visual references of object instances for training. Yet, the testing scene is still unknown in training time. Therefore, early method Cut Paste Learn (CPL) collects random background images to model diverse testing scenes (Fig. 2a), cuts instances from the visual references, and pastes them on these background images to construct a synthetic training set [9, 11]. It is worth noting that background image sampling is utilizing open-world information in a way of data collection in the open world. On the other hand, the *novel instance detection (NID)* setting, introduced recently [26], studies detecting novel object instances that are specified only in testing time where a trained model is not allowed to be tuned further. To approach NID, VoxDet [26] uses large-scale synthetic data (Fig. 2b) to learn

[*]Equal contributions; [†]Corresponding authors.

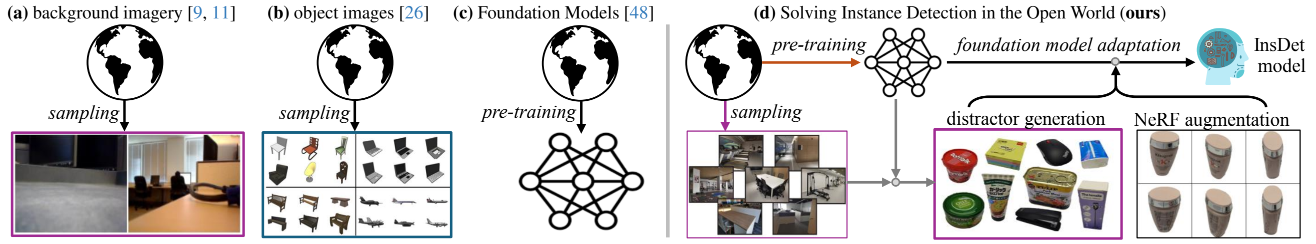

Figure 2. Existing InsDet methods leverage the open-world information in different aspects, such as **(a)** background image sampling (from the open world) to synthesize training data [9, 11], **(b)** object image sampling (from the open world) to learn feature representations [26], and **(c)** foundation model utilization (pretrained in the open world) for proposal detection and instance-level feature matching [48]. **(d)** As FMs are not specifically designed for instance-level feature matching required by InsDet, we propose to adapt them by leveraging rich data sampled from the open world. We gather data from multiple sources: (1) any available visual references of instances in the CID setting, (2) abundant multi-view object images sampled in the open world similar to **(b)**, (3) synthetic data by training NeRF [1, 36] to generate novel-view images based on the given instances, (4) *distractors* by running FMs (esp. SAM [22]) on random open-world imagery to generate random object-like proposals (Fig. 5). We use the data above to adapt FM through metric learning. The technical novelty of our work lies in (3) and (4), as well as the design choice of metric learning to adapt FMs for InsDet.

3D voxel representations from visual references and a matching function between 2D proposals and 3D voxels. Recent work OTS-FM [48] uses off-the-shelf pretrained foundation models (Fig. 2c) for InsDet. Specifically, it uses SAM [22] to detect object proposals in scene imagery, and DINOv2 [38] to represent visual references and proposals for instance-level feature matching. OTS-FM is a non-learned method and hence is applicable in both CID and NID settings. Importantly, it achieves the state-of-the-art performance on the recently released benchmark HR-InsDet [48].

The open-world perspective. As illustrated in Fig. 1, the challenges of InsDet lie in its open-world nature: the unknown testing data distribution, and domain gaps between visual references and detected proposals. To address the open-world challenges, existing methods take different strategies to exploit the open-world information (Fig. 2a-c). For example, to address the unknown testing scene imagery, some methods sample diverse background images [9, 11], demonstrating a practice of sampling data in the open world. To mitigate the domain gap between detected proposals and visual references, [26] attempts to learn features for instance-level matching by utilizing external datasets (ShapeNet [5] and ABO [8]). This data utilization can be considered as data collection in the open world. For better proposal detection and more discriminative feature representations, [48] adopts off-the-shelf pretrained foundation models (FMs) SAM [22] and DINOv2 [38], respectively. One can think of FMs as being pretrained in the open world. Different from existing methods which have not fully capitalized the open-world information, we take the open-world perspective as the first principle and solve *InsDet* by exploiting the *Open World* (Fig. 2d). We call our approach *IDOW*.

Insights and novelties. Our insights are rooted in the understanding of InsDet's open-world nature and the practice of existing methods. Existing methods address open-world issues to some extent (Fig. 2), particularly by sampling data in the open world [9, 23, 26] and adopting foundation models (FMs) pretrained in the open world [48]. These motivate us to develop InsDet methods in the open world (*IDOW*). We find that open-world detectors yield high recall on object instances (see precision-recall curves in Fig. 9 of the Appendix) but FMs are not tailored to instance-level matching between proposals and visual references – directly using an FM for InsDet is sub-optimal. Therefore, we adapt FMs to instance-level feature representation by metric learning. We propose novel and simple techniques for data augmentation to enhance FM adaptation and feature learning: distractor sampling, and multi-view synthesis. The former helps learn more discriminative features for instance comparison, and the latter enhances visual references particularly when they are few. With these techniques, our IDOW achieves significant boosts over previously reported results on recently released challenging benchmarks (Table 1 and 2).

Contributions. We make three major contributions:

1. We elaborate on the open-world challenge of InsDet and motivate the necessity of developing methods in the open world. We solve InsDet by exploiting diverse *open data* available in the open world and FMs pretrained therein.
2. We introduce simple and effective techniques to adapt FMs to InsDet: a metric learning loss, and data augmentation with distractor sampling and novel-view synthesis.
3. We extensively compare our approach with existing ones on two recently released datasets, demonstrating that our approach outperforms them by >10 AP in both conventional and novel instance detection settings.

2. Related Work

Instance detection. Early methods use local features such as SIFT [31] and SURF [2] to match visual references [43] and image regions to localize instances [16]. Recent methods train deep neural networks for instance-agnostic proposal detection and proposal-instance matching, achieving significant improvements [26, 34, 48]. In the *con-*

ventional instance detection (CID) setting, visual references of object instances are given during training. To approach CID, prevalent method synthesize training data in a cut-paste-learn (CPL) strategy [9, 11]: cutting instances from visual references and pasting them on random background images, then learning a detector on such synthetic data. Other works strive to obtain more training samples by rendering realistic instance examples [17, 19], using data augmentation [9] and synthesizing better training images [9, 12, 25]. Moreover, recent literature introduces the *novel instance detection* (NID) setting [26], which requires detecting novel instances specified *only* in testing and does not allow for further model finetuning during testing. VoxDet [26] utilizes a pretrained open-world detector [20] for proposal detection. Nowadays, the open-world perspective advances proposal detection in a way of learning more generalizable detectors and carrying out open-vocabulary recognition [29]. While a foundational open-vocabulary detector can attach a textual description (e.g., a short phrase) to each proposal, simply using it is insufficient for instance-level matching. [48] realizes this issue although foundational detectors can yield nearly perfect recall. As a result, the recent method OTS-FM uses another FM DINOv2 [38] for proposal-instance matching [48]. Differently, we solve InsDet from the open-world perspective, exploiting open FMs and open data to learn more discriminative instance-level feature representations.

Foundation models (FMs) are pre-trained on web-scale data, which can be thought of as being sampled in the *open world*. Different FMs are trained on different types of data and can shine in different downstream tasks related to visual perception, natural language processing, or both. As this work concerns about visual perception, we briefly review Vision-Language Models (VLMs) and Visual Foundation Models (VFMs). VLMs are pretrained on web-scale image-text paired data [18, 28, 29, 44], demonstrating impressive results in high-level tasks such as visual grounding and image captioning. VFMs are pretrained primarily on visual data [4, 7, 14, 22, 38, 55, 58, 63] and can yield impressive visual perception results such as proposal or open-set object detection [29, 59]. In InsDet, the mainstream framework requires proposal detection and visual feature matching. In this framework, open-world detectors appear to achieve high recall on the instances of interest, owing to that the open-world pretrained detectors generalize quite well in the open world [29, 46, 59]. In contrast, feature matching still remains a challenge in improving InsDet. Therefore, we focus on improving features for InsDet by adapting VFMs using data sampled in the open world. [48] shows that using an off-the-shelf FM DINOv2 [38] for feature matching greatly enhances InsDet detection; yet, FM adaptation has been still under-explored in InsDet, although it is a well studied in other areas such as few-shot perception [30, 32, 51] and zero-shot perception [13, 40, 41]. Our work introduces simple and novel techniques to adapt FMs to improve InsDet.

The open-world data is publicly available for pretraining FMs. Coping with such data is also often challenging. For example, data between training and testing has distribution shifts or domain gaps [30, 49], as clearly instantiated in InsDet by the unknown testing scene imagery, in addition to the domain gaps between (clean) visual references and (occluded) proposals (Fig. 1). Speaking of distribution shifts, some existing methods sample open-world data, demonstrating a need to develop robust models in the open world [15, 23]. Existing InsDet methods also sample open-world data such as diverse background images and instance examples (Fig. 2) [9, 11]. Even so, while open-world data is abundant, most of it is not directly useful for training models for specific downstream tasks. In contrast, data augmentation aims to synthesize more tailored data to enhance models' robustness and generalization, including geometry-aware augmentation (e.g., random cropping and flipping) [7, 24, 61], photometric-aware augmentation (e.g., color jittering) [21, 56, 57], and robustness-aware augmentation (e.g., adversarial perturbation) [60]. It is worth noting that Neural Radiance Field (NeRF) [35], a recent rendering technique, can be used to synthesize novel-view images. In this work, we particularly adopt NeRF to synthesize novel-view visual references of object instances and use them to adapt FMs towards more tailored features to InsDet. To the best of our knowledge, using NeRF is under-explored in the literature of InsDet, but we find that doing so remarkably improves InsDet performance.

3. Instance Detection: Protocols and Methods

We first introduce the protocols of InsDet, including the problem definition, evaluation, and two settings. We then present our techniques to enhance foundation model adaptation to InsDet.

3.1. Protocols

Problem definition. Instance Detection (InsDet) aims to detect specific object instances from a 2D scene image (Fig. 1), where the objects of interest are defined by a "support set" of some visual references [9, 26, 48]. Previous works capture the visual references from multiple camera views, with QR code being pasted to help estimate camera poses (see visual references in Fig. 1).

Challenges. InsDet is a challenging problem due to its open-world nature: one has zero knowledge about the testing data distribution which is instantiated by unknown testing scenes, never-before-seen matters, arbitrary occlusions, background clutters, etc. In particular, object instances of interest can be novel and are only defined during testing, as emphasized by the novel instance detection setting below. Nevertheless, InsDet models must be capable of detecting

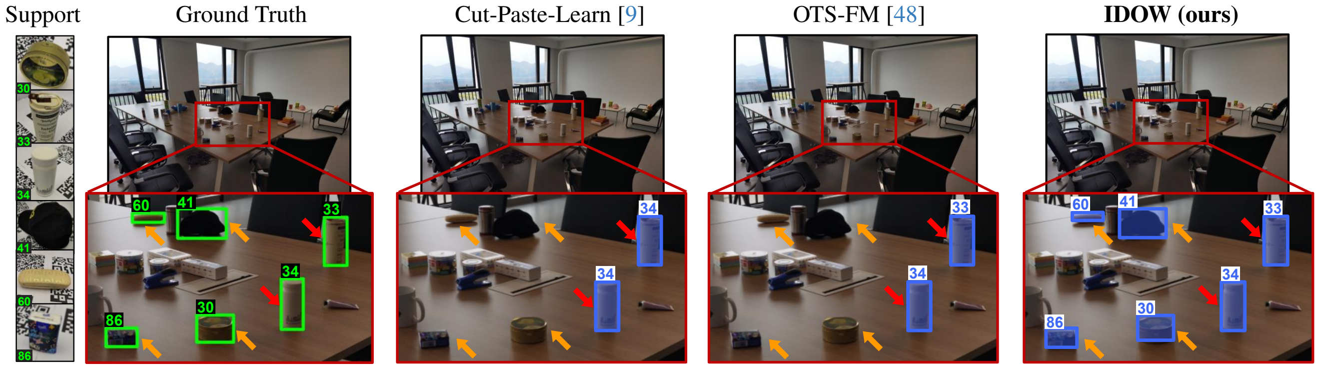

Figure 3. **Visual comparison of InsDet results by different methods in the CID setting** on HR-InsDet [48]. The testing scene image contains sparse placement of small instances with challenging illumination. We mark the ground-truth and predictions using green and blue boxes, respectively. We attach instance IDs to them to highlight whether the instance recognition is correct compared to the visual references (i.e., the leftmost references). Compared with Cut-Paste-Learn and OTS-FM, our IDOW detects more instances (see orange arrows) with better accuracy (see red arrows). Comparison between OTS-FM and IDOW suggests suggests that adapting the FM DINOv2 yields better features for IDOW to perform much better in front of domain shifts, i.e., challenging illumination conditions in this testing scene image.

any object instances of interest and robustly comparing them with visual references.

Two settings. There are two settings of InsDet, which differ in whether object instances are pre-defined during training or defined on the fly in testing time.

- *Conventional Instance Detection (CID).* In this setting, all object instances of interest are pre-defined during training [9]. That said, one can use their visual references as training data to train detectors. Yet, only visual references are available and testing scene data is still unknown. Hence, InsDet methods in CID often choose to sample diverse background images in the open world [9, 12]. This setting simulates application scenarios in AR/VR. For instance, robots should memorize the items concerned by the customer so as to provide better customized services.
- *Novel Instance Detection (NID).* This setting requires InsDet models to detect *novel* object instances defined only in testing time and does not allow the models to be finetuned further during testing [26]. In this setting, the *de facto* practice to train InsDet models turns to external data, which can be either synthetic data or those sampled in the open world. This setting simulates some real-world scenarios, for instance, robots must search for a never-before-seen luggage of a customer at an airport.

3.2. Learning Instance Detector in Open World

We present our solution of instance detection from an open-world perspective (IDOW), focusing on foundation model adaptation and data augmentation.

3.2.1. Foundation Model Adaptation

As an open-world detector yields quite high recall [26, 48], we focus on adapting a foundation model to enhance feature representations for InsDet. Although a foundational feature model (e.g., DINOv2 [38]) offers feature representations applicable to various downstream tasks, it is not tailored to InsDet. To improve feature representation, one can finetune this FM over relevant data, such as those sampled in the open world [26]. Intuitively, the finetuned feature is expected to match proposals and visual references better if they are from the same object instances; otherwise, distinguish them. Therefore, we propose to use a metric learning loss for FM adaptation. Moreover, we sample *distractors* from imagery in the open world. The distractors serve as negatives in metric learning.

We denote an FM as f_θ, parameterized by θ. This FM serves as the feature extractor $f_\theta(\mathbf{I}) : \mathbf{I} \rightarrow \mathbb{R}^q$ to produce q-dimensional features for any given visual reference image or proposal $\mathbf{I}$. We finetune $f_\theta(\cdot)$ to better serve InsDet such that features of reference images from the same instance should present high similarity, otherwise low similarity. We achieve this by finetuning f_θ using a simple metric learning loss ℓ. Specifically, we construct triplets of training data $(\mathbf{I}_a, \mathbf{I}_p, \mathbf{I}_n)$, where $\mathbf{I}_a$, $\mathbf{I}_p$ and $\mathbf{I}_n$ represent an anchor visual reference, positive and negative samples, respectively. The training data contains examples sampled or synthesized in the open world [26], as well as visual references available in the CID setting. Negative examples are cross-instance visual examples and distractors (detailed later). Below is the loss:

$$\ell = \Big[d(f_\theta(\mathbf{I}_a), f_\theta(\mathbf{I}_p)) - d(f_\theta(\mathbf{I}_a), f_\theta(\mathbf{I}_n)) + \alpha\Big]_+, \quad (1)$$

where α is a hyper-parameter determining the margin between an instance and other negative data. $d(\cdot)$ measures the distance between two examples, e.g., inverse cosine similarity used in our work. After training, we use the learned features to represent both proposals and visual references. We follow the instance-proposal matching pipeline [48] to produce the final InsDet results: computing pairwise similarities between each proposal and each visual reference, running stable matching algorithm [10, 33], and returning

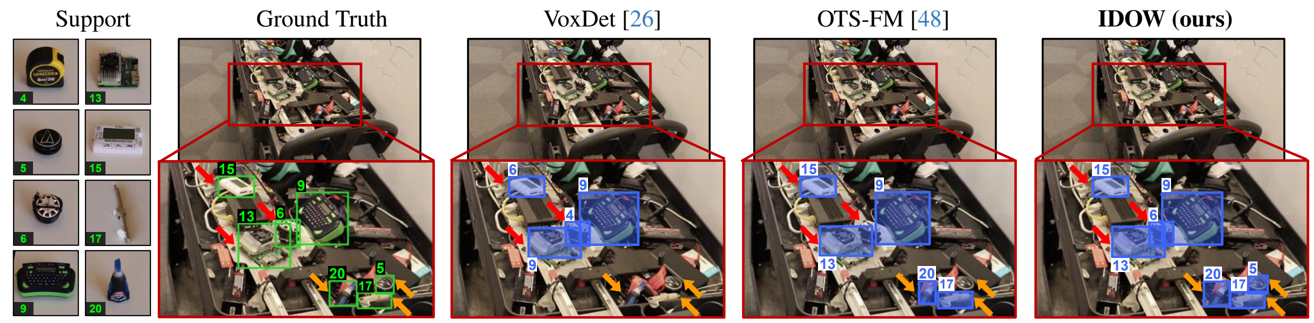

Figure 4. **Visual comparison of InsDet results by different methods in the NID setting** on RoboTools [26]. On the cluttered testing scene image, we mark the ground-truth and predictions using green and blue boxes, respectively. We attach instance IDs to them to highlight whether the instance recognition is correct compared to the visual references (i.e., the leftmost references). Compared with VoxDet and OTS-FM, our IDOW detects more instances (see orange arrows) with better accuracy (see red arrows). Comparing OTS-FM and our IDOW suggests that, although having not seen the visual references during adaption in this NID setting, features by our IDOW are a better representation for InsDet than the off-the-shelf foundation model (i.e., DINOv2) used in OTS-FM.

the matched pairs if they have similarity scores greater than a predefined threshold (0.4 as used in [48]).

Hard example sampling. To construct training batches of triplets, it is important to effectively sample negative examples, easy negative examples do not provide sufficient gradients during training [50]. In this work, we adopt a batch-level hard negative sampling strategy. In each training batch, we first sample a reference image as an anchor $\mathbf{I}_a$. We then sample a random example from the same instance as the positive one $\mathbf{I}_p$. We finally sample a hard negative example from the union of the reference images of all the other instances and the distractors S_{dt} (explained in the next subsection). Mathematically, we find hard negative patch $\mathbf{I}_{hn}$ with the objective: $\arg\min_{\mathbf{I}_n} d(f_\theta(\mathbf{I}_a), f_\theta(\mathbf{I}_n)),\ \mathbf{I}_n \in \{C(\mathbf{I}_i) \neq C(\mathbf{I}_a), \text{for } i = 1 \ldots N\} \cup S_{dt}$, where $C(\mathbf{I}_i)$ represents the instance ID of the data $\mathbf{I}_i$, and N indicates the total number of visual reference images.

3.2.2. The Proposed Data Augmentation

To better adapt FM for instance-level feature matching, we introduce two data augmentation techniques below.

Distractor sampling aims to sample patches of random background images as universal negative data to all object instances. It is a data augmentation technique, expected to define the open space and help features better characterize meaningful object instances. Distractor sampling has been adopted in the literature. For example, open-set recognition samples distractors as a separate class, known as "the-other-class" or "background-class", to define the open space. In InsDet, the cut-paste-learn strategy [9] samples random background photos to define the background content. In this work, we run SAM [22] on random background photos and use the segments as distractors. One might worry about the possibility of sampled distractors being exactly some instances of interest and including them as negatives may destabilize model adaptation. Yet, given a large number of distractors, this rarely becomes an issue as demonstrated by the success of various self-supervised learning methods [38] which may sample examples serving as both positive and negative data at the same time.

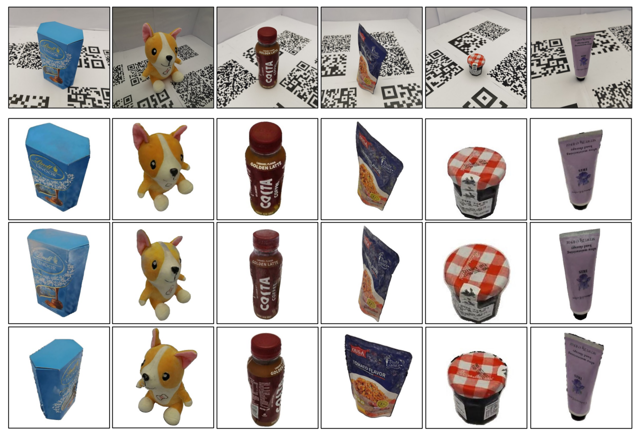

Figure 5. We use NeRF to synthesize novel-view object images to augment the limited given visual references. In this work, we train per-instance Zip-NeRF [1]. We visualize synthesized images at different angles (in row 2-4) together with the raw visual references (in the first row). The visual references are from HR-InsDet dataset [48], where the QR code is used for camera pose estimation. Overall, we find synthesized images show high visual quality.

Novel-view synthesis. Each object instance might be represented by only a few visual examples, especially in the CID setting which provides limited real visual references for instances of interest. To augment available visual references for better adapting FM, we learn a NeRF [35] for each instance and use NeRF to synthesize novel-view images. In this process, we estimate relative camera poses (required by NeRF) using COLMAP [47], as commonly done in NeRF methods [1, 35] and InsDet approaches [26, 48]. Fig. 5 displays some random NeRF-generated novel-view images with good quality in visuals. It is worth noting that a NeRF needs to be trained only once on the visual references of an object instance. Once it is trained, we use it to generate

Table 1. **Benchmarking results in the CID setting** on the HR-InsDet dataset. We compare our IDOW with state-of-the-arts and make three salient conclusions. First, IDOW significantly outperforms previous methods, e.g., $\text{IDOW}_{\text{GroundingDINO}}$ (57.01 AP) > $\text{OTS-FM}_{\text{GroundingDINO}}$ (51.68 AP) > CPL_{DINO} (27.99 AP). This confirms the importance of addressing InsDet from the open-world perspective. Second, adapting FMs by our IDOW further boosts the performance by 5-7 AP, e.g., IDOW_{SAM} (48.75 AP) > $\text{OTS-FM}_{\text{SAM}}$ (41.61 AP). Third, IDOW and OTS-FM are applicable to different pretrained FMs and adopting stronger FMs achieves better performance, e.g., using GroundingDINO yields >8 AP than SAM in IDOW.

Method	**Venue & Year**	**AP**						**AP$_{50}$**	**AP$_{75}$**
		avg	hard	easy	small	medium	large		
$\text{CPL}_{\text{FasterRCNN}}$ [9, 45]	NeurIPS 2015	19.54	10.26	23.75	5.03	22.20	37.97	29.21	23.26
$\text{CPL}_{\text{RetinaNet}}$ [9, 27]	ICCV 2017	22.22	14.92	26.49	5.48	25.80	42.71	31.19	24.98
$\text{CPL}_{\text{CenterNet}}$ [9, 64]	CVPR 2019	21.12	11.85	25.70	5.90	24.15	40.38	32.72	23.60
CPL_{FCOS} [9, 54]	ICCV 2019	22.40	13.22	28.68	6.17	26.46	38.13	32.80	25.47
CPL_{DINO} [9, 59]	ICLR 2023	27.99	17.89	32.65	11.51	31.60	48.35	39.62	32.19
$\text{OTS-FM}_{\text{SAM}}$ [22, 48]	NeurIPS 2023	41.61	28.03	47.57	14.58	45.83	69.14	49.10	45.95
$\text{OTS-FM}_{\text{GroundingDINO}}$ [29, 48]	NeurIPS 2023	51.68	37.23	58.72	28.79	58.55	69.22	62.50	56.78
IDOW_{SAM}		48.75	32.09	56.50	20.75	55.26	73.43	57.59	54.06
$\text{IDOW}_{\text{GroundingDINO}}$		**57.01**	**40.74**	**64.36**	**35.25**	**62.98**	**73.64**	**69.33**	**62.84**

more visual references not only for training but for testing. In particular, such NeRF-synthesized novel-view images can be stored together with the original visual references in testing for proposal-instance matching. Doing so significantly improves InsDet performance (Table 3 and Fig. 7). To the best of our knowledge, our work is the first that exploits NeRF to synthesize data for InsDet.

4. Experiments

In this section, we first introduce the experimental setup, then evaluate our IDOW by comparing existing methods in both CID and NID settings, and finally conduct ablation studies to analyze our approach.

4.1. Experimental Setup

Datasets. We use two recently published datasets in our experiments. HR-InsDet [48] is developed for the CID setting, containing 100 daily object instances, 160 high-resolution testing images from 14 indoor scenarios (see Fig. 3 for an example). RoboTools [26] is developed for the NID setting, containing 20 robotic tool instances and 1,581 testing images from 24 indoor scenarios (see Fig. 4 for an example). For a fair comparison, we use the OWID dataset released by [26] to adapt FM in the NID setting.

Metrics. We use metrics adopted in both datasets mentioned above, including average precision (AP) at IoU thresholds from 0.5 to 0.95 with the step size 0.05 as the primary metric and also report AP$_{50}$ and AP$_{75}$ averaged over all instances with IoU threshold as 0.5 and 0.75, respectively. The HR-InsDet dataset contains tags for hard/easy testing scenes and small/medium/large object instances. We follow [48] to break down results on these tags.

Compared methods. We compare previous methods including CPL, OTS-FM, and VoxDet. These methods have options to use different components (so do ours). For example, CPL can train a detector using one of the popular architectures Faster RCNN [45], RetinaNet [27], CenterNet [64], FCOS [54], and DINO [59]. OTS-FM can use SAM or GroundingDINO for proposal detection. We implement these methods by using different backbones to improve their performance (Table 1). We use subscripts to mark the backbones used in each method. Moreover, we compare more methods in the NID setting, including OS2D [39], DTOID [34], and $\text{OLN}_{\text{Corr.}}$ [20].

Implementation details. We carry out all experiments on a single NVIDIA 3090 GPU and implement methods with PyTorch. Unless otherwise specified, our IDOW uses GroundingDINO [29] as the proposal detector and DINOv2 [38] as the feature foundation model to finetune. We use the pretrained [CLS] token to extract features and class embedding to compute similarity scores for matching. When finetuning, we set hyper-parameter $\alpha = 0.5$ in the metric learning loss. We use Adam optimizer with a learning rate 1e-3 and weight decay 0.5. We set the batch size as 100 and finetune the model for 10 epochs.

4.2. Benchmarking Results

Quantitative results. We evaluate our method against prior approaches on the InsDet and RoboTools dataset. Table 1 and 2 list detailed results in the CID and NID settings, respectively. We summarize three salient observations:

First, *approaching InsDet from the open-world perspective significantly outperforms previous methods.* From both CID and NID settings, we find that, despite not being explicitly trained on the instances of interests, OTS-FM already achieves >10 AP than traditional detector-based methods which particularly train on such instances. Adapting FMs by our IDOW boosts performance by 3-5 AP.

Second, *IDOW significantly outperforms state-of-the-art*

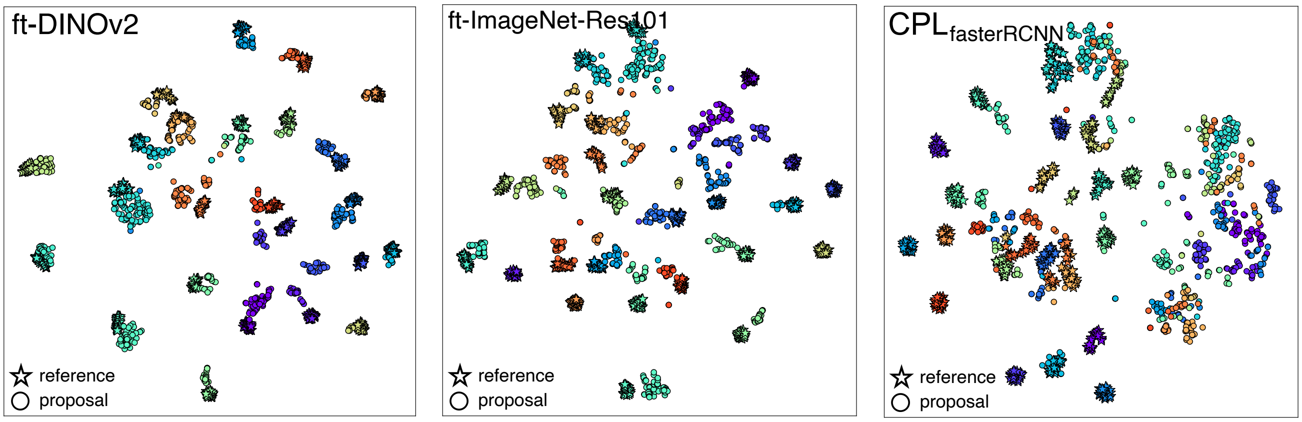

Figure 6. **t-SNE visualization of features** learned by different methods. We compare our finetuned DINOv2 against a finetuned ImageNet-pretrained ResNet101 model and the baseline instance detector CPL with a FasterRCNN architecture. We color feature points w.r.t instance IDs and overlay star-marked visual references (zoom-in to see better). Visually, the finetuned DINOv2 extracts more discriminative features.

Table 2. **Benchmarking results in the NID setting** on RoboTools. We compare our IDOW with state-of-the-arts and make three salient conclusions. First, IDOW significantly outperforms previous methods, e.g., IDOW$_{\text{GroundingDINO}}$ (59.4 AP) > OTS-FM$_{\text{GroundingDINO}}$ (56.7 AP) > VoxDet (18.7 AP). This demonstrates the effectiveness of approaching InsDet from the open-world perspective. Second, adapting FMs by our IDOW further boosts the performance by 3-5 AP, e.g., IDOW$_{\text{SAM}}$ (51.9 AP) > OTS-FM$_{\text{SAM}}$ (46.5 AP). Third, adopting stronger FMs achieves better performance, e.g., IDOW$_{\text{GroundingDINO}}$ (59.4 AP) > IDOW$_{\text{SAM}}$ (51.9 AP) in IDOW.

Method	**Venue & Year**	**AP**	**AP$_{50}$**	**AP$_{75}$**
OS2D [39]	ECCV 2020	2.9	6.5	2.0
DTOID [34]	WACV 2021	3.6	9.0	2.0
OLN$_{\text{Corr.}}$ [20]	RA-L 2022	14.4	18.1	15.7
VoxDet [26]	NeurIPS 2023	18.7	23.6	20.5
OTS-FM$_{\text{SAM}}$ [48]	NeurIPS 2023	46.5	55.9	50.4
OTS-FM$_{\text{GroundingDINO}}$ [48]	NeurIPS 2023	56.7	64.8	59.0
IDOW$_{\text{SAM}}$		51.9	63.8	56.5
IDOW$_{\text{GroundingDINO}}$		**59.4**	**67.8**	**61.8**

InsDet methods, achieving over 5 AP higher on average than methods like OTS-FM in both CID and NID settings, demonstrating the effectiveness of our techniques.

Third, *IDOW is versatile and applicable to different open-world pretrained FMs.* By comparing IDOW$_{\text{SAM}}$ and IDOW$_{\text{GroundingDINO}}$, we find that adopting stronger FMs (GroundingDINO vs. SAM) achieves better performance.

Qualitative Results. Fig. 3 and 4 compares InsDet results by different methods in the CID and NID settings, respectively. We can see that our IDOW has higher precision and detects more wanted instances, which are small and under low lighting conditions. Fig. 6 uses tSNE to visualize features of visual references and detected proposals (in the HR-InsDet dataset) computed by our finetuned DINOv2 (ft-DINOv2) in the left panel, and other non-FM methods including a finetuned ImageNet-pretrained ResNet101 backbone (ft-ImageNet-Res101) and CPL$_{\text{FasterRCNN}}$ in the middle and right panels, respectively. Clearly, ft-DINOv2 matches proposals and references much better than other models.

4.3. Ablation Study and Further Analysis

In Table 3, we ablate the proposed techniques of NeRF-synthesis and distractor sampling. Results shows that adapting the FM DINOv2 greatly boosts InsDet performance (from 51.68 AP by the baseline to 53.94 AP). Importantly, including NeRF-synthesized images in both training and testing significantly improves the performance. Yet, using them in testing brings more performance gains than training (refer to the next study in Fig. 7). Eventually, applying our techniques boosts AP to 57.01. The supplemental material contains more quantitative and qualitative results.

Following the above, we study the affects of using different number of NeRF-synthesized images in training and test. Fig. 7 depicts the results. (1) *Adding NeRF-generated images helps foundation model adaptation for InsDet*. Specifically, adding <24 generated images boosts the performance, especially when 0 synthesized images are used in testing. However, adding >24 generated images for training induces diminishing performance gains, likely because that the adapted FM overfits to artifacts in NeRF generated images. (2) *Adopting more synthetic images in testing consistently improves performance.* Yet, the performance by adding more (e.g., >36) in testing saturates. The peak performance occurs when 36 NeRF-synthesized images are used in testing.

5. Discussions

Broader Impacts. Approaching InsDet in the open-world has a significant impact on various downstream applications, such as robotics and AR/VR. Detecting object instances in RGB images is considered the first step of contemporary perception algorithms. A stronger InsDet model provides richer information for the following steps, such

Table 3. **Ablation study of each strategy involved in our IDOW.** We carry out the study in the CID setting on the HR-InsDet dataset. We use OTS-$FM_{GroundingDINO}$ as a baseline, over which we incrementally add each strategy. "Train" means foundation model adaptation through finetuning on the available data. **DA** denotes **D**ata **A**ugmentation with NeRF-generated novel-views; **DS** denotes **D**istractor **S**ampling. Results clearly demonstrate that all the four strategies help achieve better InsDet performance. Finetuning FMs on the given visual references enhances detection performance, cf. Train (53.94 AP) > baseline (51.68 AP). Moreover, NeRF-based data augmentation improves the final detection performance, particularly when used in testing, cf. Train+DA@Test (56.44 AP) > Train+DA@Train (54.48 AP) > baseline (51.68 AP). Lastly, applying distractor sampling (DS) improves the final performance further, cf. Train+DS (54.10 AP) > Train (53.94 AP).

Strategies				**AP**						**AP_{50}**	**AP_{75}**
DA@Test	Train	DA@Train	DS	avg	hard	easy	small	medium	large		
baseline: OTS-$FM_{GroundingDINO}$				51.68	37.23	58.72	28.79	58.55	69.22	62.50	56.78
	✓			53.94	37.54	61.52	30.18	60.98	71.60	65.18	59.36
	✓	✓		54.48	38.31	61.71	31.18	61.10	72.55	66.06	59.98
	✓		✓	54.10	37.73	61.65	30.27	61.19	71.27	65.37	59.57
	✓	✓	✓	54.92	38.28	62.41	32.00	61.61	74.21	66.58	60.50
✓				55.24	39.64	62.81	34.00	61.20	72.38	66.99	60.88
✓	✓			56.44	39.71	64.24	34.72	63.13	74.42	68.50	62.31
✓	✓	✓		56.92	39.85	64.85	34.66	62.74	75.48	68.89	62.80
✓	✓		✓	56.51	39.70	64.36	34.34	61.65	75.89	68.71	62.38
✓	✓	✓	✓	57.01	40.74	64.36	35.25	62.98	73.64	69.33	62.84

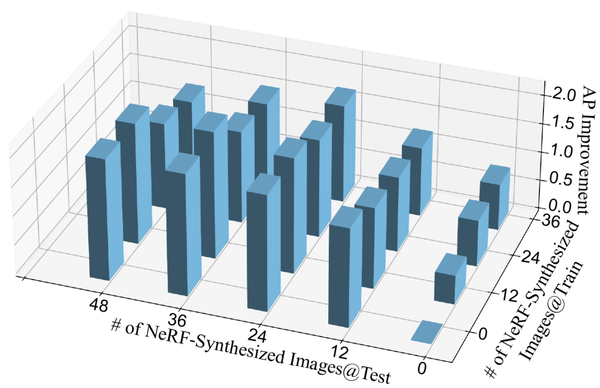

Figure 7. **Analysis of the number of added NeRF-synthesized images in training and testing w.r.t AP improvement** over the "Train" method (Table 3), which by default finetunes DINOv2 using the available visual references (24 per instance). We carry out this analysis in the CID setting on the HR-InsDet dataset. On the one hand, adding NeRF-synthesized images helps finetune FM for InsDet, but adding more (e.g., >24) hurts the performance, likely due to overfitting to the artifacts of NeRF-synthesized images. On the other hand, adding more NeRF images in testing consistently improves performance.

as grasping or trajectory forecasting in robotics. Additionally, our work shows the promising performance of adapting FMs to InsDet, which will foster future research in studying different properties of FMs, such as adapting FMs to other tasks. Nonetheless, our research utilizes FMs that are potentially pretrained on private data. At this point, it is unclear how such FMs might have bias or ethical issues. As a result, we have no safeguard measure to ensure that the finetuned/adapted FMs are fair, even though we find no visible issues in the image-only dataset used for finetuning. Further, adopting SAM may result in missed detections of critical instances relevant to specific downstream applications, such as medical devices or other rare objects. Such potential issues are not addressed in this work.

Limitations and future work. Experiments demonstrate that our proposed techniques effectively leverage FMs to address InsDet across various settings. Nonetheless, we note several limitations for future improvements. First, the current pipeline relies on the proposals generated from GroundingDINO. Despite its high recall, it is not yet ready for real-time applications. Future work should consider faster proposal detection methods. Second, the NeRF augmentation step is also slow since it requires training a NeRF for every instance, which takes 1 hour and 500MB space using one NVIDIA 3090 GPU. While developing efficient NeRF methods is beyond our scope, we point out that efficient NeRF methods can be readily used in our pipeline, such as [37] which trains a NeRF in seconds and [6] which can train a single NeRF for multiple objects.

6. Conclusion

Instance detection is highly applicable in various scenarios such as in robotics and AR/VR. We elaborate on its open-world nature, necessitating the development of InsDet models in the open world. We solve InsDet from the open world perspective, embracing foundation models (FMs) and data sampled in the open world. In particular, we propose simple and effective techniques such as metric learning, distractor sampling, and novel-view synthesis. Extensive experiments in different settings validate that our techniques in adapting FMs significantly boost InsDet performance.

Acknowledgments

This work was supported by FDCT (0067/2024/ITP2), the University of Macau (SRG2023-00044-FST), the Institute of Collaborative Innovation and NSFC (No.62206256).

References

[1] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *ICCV*, 2023. 2, 5

[2] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *ECCV*, 2006. 2

[3] Richard Bormann, Xinjie Wang, Markus Völk, Kilian Kleeberger, and Jochen Lindermayr. Real-time instance detection with fast incremental learning. In *ICRA*. IEEE, 2021. 1

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 3

[5] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv:1512.03012*, 2015. 2

[6] Hansheng Chen, Jiatao Gu, Anpei Chen, Wei Tian, Zhuowen Tu, Lingjie Liu, and Hao Su. Single-stage diffusion nerf: A unified approach to 3d generation and reconstruction. In *ICCV*, 2023. 8

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 3, 1

[8] Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, et al. Abo: Dataset and benchmarks for real-world 3d object understanding. In *CVPR*, 2022. 2

[9] Debidatta Dwibedi, Ishan Misra, and Martial Hebert. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *ICCV*, 2017. 1, 2, 3, 4, 5, 6

[10] David Gale and Lloyd S Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962. 4

[11] Georgios Georgakis, Arsalan Mousavian, Alexander C. Berg, and Jana Kosecka. Synthesizing training data for object detection in indoor scenes. In *Proceedings of Robotics: Science and Systems*, 2017. 1, 2, 3

[12] Georgios Georgakis, Arsalan Mousavian, Alexander C Berg, and Jana Košecká. Synthesizing training data for object detection in indoor scenes. *Robotics: Science and Systems (RSS)*, 2017. 3, 4

[13] Sachin Goyal, Ananya Kumar, Sankalp Garg, Zico Kolter, and Aditi Raghunathan. Finetune like you pretrain: Improved finetuning of zero-shot vision models. In *CVPR*, 2023. 3

[14] Sunan He, Taian Guo, Tao Dai, Ruizhi Qiao, Chen Wu, Xiujun Shu, and Bo Ren. Vlmae: Vision-language masked autoencoder. *arXiv:2208.09374*, 2022. 3

[15] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. In *ICLR*, 2019. 3

[16] Stefan Hinterstoisser, Cedric Cagniart, Slobodan Ilic, Peter Sturm, Nassir Navab, Pascal Fua, and Vincent Lepetit. Gradient response maps for real-time detection of textureless objects. *IEEE transactions on pattern analysis and machine intelligence*, 34(5):876–888, 2011. 2

[17] Tomáš Hodaň, Vibhav Vineet, Ran Gal, Emanuel Shalev, Jon Hanzelka, Treb Connell, Pedro Urbina, Sudipta N Sinha, and Brian Guenter. Photorealistic image synthesis for object instance detection. In *IEEE international conference on image processing (ICIP)*, 2019. 3

[18] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021. 3

[19] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In *ICCV*, 2017. 3

[20] Dahun Kim, Tsung-Yi Lin, Anelia Angelova, In So Kweon, and Weicheng Kuo. Learning open-world object proposals without learning to classify. *IEEE Robotics and Automation Letters*, 7(2):5453–5460, 2022. 3, 6, 7

[21] Eun Kyeong Kim, Hansoo Lee, Jin Yong Kim, and Sungshin Kim. Data augmentation method by applying color perturbation of inverse psnr and geometric transformations for object recognition based on deep learning. *Applied Sciences*, 10(11): 3755, 2020. 3

[22] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, 2023. 2, 3, 5, 6

[23] Shu Kong and Deva Ramanan. Opengan: Open-set recognition via open data generation. In *ICCV*, 2021. 2, 3

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *NeurIPS*, 25, 2012. 3

[25] Kevin Lai, Liefeng Bo, and Dieter Fox. Unsupervised feature learning for 3d scene labeling. *ICRA*, 2014. 3

[26] Bowen Li, Jiashun Wang, Yaoyu Hu, Chen Wang, and Sebastian Scherer. Voxdet: Voxel learning for novel instance detection. *NeurIPS*, 2024. 1, 2, 3, 4, 5, 6, 7

[27] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 6, 3

[28] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *NeurIPS*, 2023. 3

[29] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *ECCV*, 2024. 3, 6

[30] Tian Liu, Huixin Zhang, Shubham Parashar, and Shu Kong. Few-shot recognition via stage-wise retrieval-augmented finetuning. In *CVPR*, 2025. 3

[31] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60:91–110, 2004. 2

[32] Anish Madan, Neehar Peri, Shu Kong, and Deva Ramanan. Revisiting few-shot object detection with vision-language models. In *NeurIPS Datasets & Benchmark Track*, 2024. 3
[33] David G McVitie and Leslie B Wilson. The stable marriage problem. *Communications of the ACM*, 14(7):486–490, 1971. 4
[34] Jean-Philippe Mercier, Mathieu Garon, Philippe Giguere, and Jean-Francois Lalonde. Deep template-based object instance detection. In *WACV*, 2021. 1, 2, 6, 7
[35] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 3, 5
[36] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2
[37] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022. 8
[38] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv:2304.07193*, 2023. 2, 3, 4, 5, 6
[39] Anton Osokin, Denis Sumin, and Vasily Lomakin. Os2d: One-stage one-shot object detection by matching anchor features. In *ECCV*, 2020. 6, 7
[40] Shubham Parashar, Zhiqiu Lin, Yanan Li, and Shu Kong. Prompting scientific names for zero-shot species recognition. In *EMNLP*, 2023. 3
[41] Shubham Parashar, Zhiqiu Lin, Tian Liu, Xiangjue Dong, Yanan Li, Deva Ramanan, James Caverlee, and Shu Kong. The neglected tails in vision-language models. In *CVPR*, 2024. 3
[42] Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen, and Sundaraja S Iyengar. A survey on deep learning: Algorithms, techniques, and applications. *ACM computing surveys (CSUR)*, 51(5):1–36, 2018. 1
[43] A Quadros, James Patrick Underwood, and Bertrand Douillard. An occlusion-aware feature for range images. In *ICRA*, 2012. 2
[44] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 3
[45] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 6, 3
[46] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv:2401.14159*, 2024. 3
[47] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *CVPR*, 2016. 5
[48] Qianqian Shen, Yunhan Zhao, Nahyun Kwon, Jeeeun Kim, Yanan Li, and Shu Kong. A high-resolution dataset for instance detection with multi-view object capture. In *NeurIPS Datasets & Benchmark Track*, 2023. 1, 2, 3, 4, 5, 6, 7
[49] Jia Shi, Gautam Gare, Jinjin Tian, Siqi Chai, Zhiqiu Lin, Arun Vasudevan, Di Feng, Francesco Ferroni, and Shu Kong. Lca-on-the-line: Benchmarking out-of-distribution generalization with class taxonomies. In *ICML*, 2024. 3
[50] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *CVPR*, 2016. 5
[51] Julio Silva-Rodriguez, Sina Hajimiri, Ismail Ben Ayed, and Jose Dolz. A closer look at the few-shot adaptation of large vision-language models. In *CVPR*, 2024. 3
[52] Afrina Tabassum, Muntasir Wahed, Hoda Eldardiry, and Ismini Lourentzou. Hard negative sampling strategies for contrastive representation learning. *arXiv:2206.01197*, 2022. 1
[53] Peng Tang, Xinggang Wang, Xiang Bai, and Wenyu Liu. Multiple instance detection network with online instance classifier refinement. In *CVPR*, 2017. 1
[54] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, 2019. 6, 3
[55] Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. In *ECCV*, 2022. 3
[56] Riccardo Volpi, Pietro Morerio, Silvio Savarese, and Vittorio Murino. Adversarial feature augmentation for unsupervised domain adaptation. In *CVPR*, 2018. 3
[57] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. *NeurIPS*, 2018. 3
[58] XuDong Wang, Jingfeng Yang, and Trevor Darrell. Segment anything without supervision. In *NeurIPS*, 2024. 3
[59] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel Ni, and Harry Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. In *ICLR*, 2022. 3, 6
[60] Xinyu Zhang, Qiang Wang, Jian Zhang, and Zhao Zhong. Adversarial autoaugment. In *ICLR*, 2020. 3
[61] Yunhan Zhao, Shu Kong, and Charless Fowlkes. Camera pose matters: Improving depth prediction by mitigating pose distribution bias. In *CVPR*, 2021. 3
[62] Yunhan Zhao, Haoyu Ma, Shu Kong, and Charless Fowlkes. Instance tracking in 3d scenes from egocentric videos. In *CVPR*, 2024. 1
[63] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. Image bert pre-training with online tokenizer. In *ICLR*, 2022. 3
[64] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. In *arXiv:1904.07850*, 2019. 6, 3