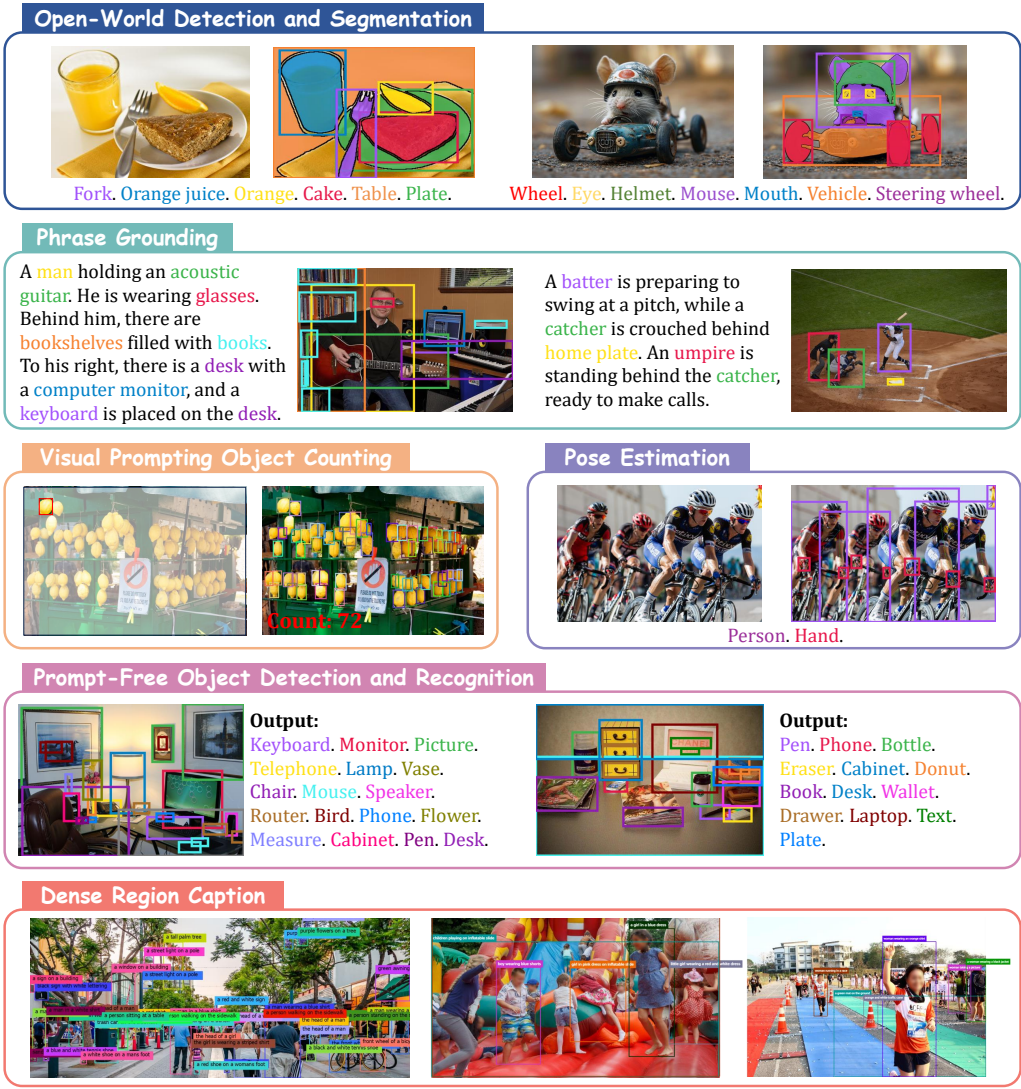


DINO-X: A Unified Vision Model for Open-World Object Detection and Understanding

IDEA Research Team

International Digital Economy Academy (IDEA), IDEA Research
<https://deepdataspace.com/home>



DINO-X

Figure 1: DINO-X is a unified object-centric vision model which supports various open-world perception and object-level understanding tasks, including Open-World Object Detection and Segmentation, Phrase Grounding, Visual Prompt Counting, Pose Estimation, Prompt-Free Object Detection and Recognition, Dense Region Caption, etc.

Abstract

In this paper, we introduce DINO-X, which is a unified object-centric vision model developed by IDEA Research with the best open-world object detection performance to date. DINO-X employs the same Transformer-based encoder-decoder architecture as Grounding DINO 1.5 [47] to pursue an object-level representation for open-world object understanding. To make long-tailed object detection easy, DINO-X extends its input options to support text prompt, visual prompt, and customized prompt. With such flexible prompt options, we develop a universal object prompt to support *prompt-free* open-world detection, making it possible to detect anything in an image without requiring users to provide any prompt. To enhance the model’s core grounding capability, we have constructed a large-scale dataset with over 100 million high-quality grounding samples, referred to as Grounding-100M, for advancing the model’s open-vocabulary detection performance. Pre-training on such a large-scale grounding dataset leads to a foundational object-level representation, which enables DINO-X to integrate multiple perception heads to simultaneously support multiple object perception and understanding tasks, including detection, segmentation, pose estimation, object captioning, object-based QA, etc. DINO-X encompasses two models: the Pro model, which provides enhanced perception capabilities for various scenarios, and the Edge model, which is optimized for faster inference speed and better suited for deployment on edge devices. Experimental results demonstrate the superior performance of DINO-X. Specifically, the DINO-X Pro model achieves 56.0 AP, 59.8 AP, and 52.4 AP on the COCO, LVIS-minival, and LVIS-val zero-shot object detection benchmarks, respectively. Notably, it scores 63.3 AP and 56.5 AP on the rare classes of LVIS-minival and LVIS-val benchmarks, both improving the previous SOTA performance by 5.8 AP. Such a result underscores its significantly improved capacity for recognizing long-tailed objects. Our demo and API will be released at <https://github.com/IDEA-Research/DINO-X-API>.

1 Introduction

In recent years, object detection has gradually evolved from closed-set detection models [74, 28, 4] to open-set detection models [33, 29, 76], which can identify objects corresponding to user-provided prompt. Such models have numerous practical applications, such as enhancing the adaptability of robots in dynamic environments, assisting autonomous vehicles in rapidly locating and reacting to new objects, improving the perceptual capabilities of multimodal large language models (MLLMs), reducing their hallucinations, and increasing the reliability of their responses.

In this paper, we introduce DINO-X, which is a unified object-centric vision model developed by IDEA Research with the best open-world object detection performance to date. Building upon Grounding DINO 1.5 [47], DINO-X employs the same Transformer encoder-decoder architecture and adopts open-set detection as its core training task. To make long-tailed object detection easy, DINO-X incorporates a more comprehensive prompt design at the model’s input stage. Traditional text prompt-only models [33, 47, 29], while having made great progress, still struggle to cover a sufficient range of long-tailed detection scenarios due to the difficulty of collecting sufficiently diverse training data to cover various applications. To overcome this shortage, in DINO-X, we extend the model architecture to support the following three types of prompts. (1) Text Prompt: This involves identifying desired objects based on user-provided text input, which can cover most of the detection scenarios. (2) Visual Prompt: Beyond text prompts, DINO-X also supports visual prompts as in T-Rex2 [18], further covering detection scenarios that cannot be well described by text alone. (3) Customized Prompt: To enable more long-tailed detection problems, we particularly introduce customized prompt in DINO-X, which can be implemented as either pre-defined or user-tuned prompt embeddings for customized needs. Through prompt-tuning, we can create domain-customized prompts for different domains or function-specific prompts to address various functional needs. For instance, in DINO-X, we develop a universal object prompt to support *prompt-free* open-world object detection, making it possible to detect any objects in a given image without requiring users to provide any prompt.

To achieve a strong grounding performance, we collected and curated over 100 million high-quality grounding samples from diverse sources, termed as Grounding-100M. Pre-training on such a large-scale grounding dataset leads to a foundational object-level presentation, which enables DINO-X to integrate multiple perception heads to simultaneously support multiple object perception and understanding tasks. Beyond the box head for object detection, DINO-X has implemented three additional heads: (1) Mask Head for predicting segmentation masks for the detected objects, (2) Keypoint Head for predicting more semantically meaningful keypoint for specific categories, and (3) Language Head for generating fine-grained descriptive captions for each detected object. By integrating these heads, DINO-X could provide more detailed object-level understanding of an input image. In Figure 1, we list various examples to illustrate the object-level vision tasks supported by DINO-X.

Similar to Grounding DINO 1.5, DINO-X also encompasses two models: the DINO-X Pro model, which provides enhanced perception capabilities for various scenarios, and the DINO-X Edge model, which is optimized for faster inference speed and better suited for deployment on edge devices. Experimental results demonstrate the superior performance of DINO-X. As illustrated in Figure 2, our DINO-X Pro model achieves 56.0 AP, 59.8 AP, and 52.4 AP on the COCO, LVIS-minival, and LVIS-val zero-shot transfer benchmarks, respectively. Notably, it scores 63.3 AP and 56.5 AP on the rare classes of the LVIS-minival and LVIS-val benchmarks, showing improvements of 5.8 AP and 5.0 AP over Grounding DINO 1.6 Pro, and 7.2 AP and 11.9 AP over Grounding DINO 1.5 Pro, highlighting its significantly improved ability to recognize long-tailed objects.

2 Approach

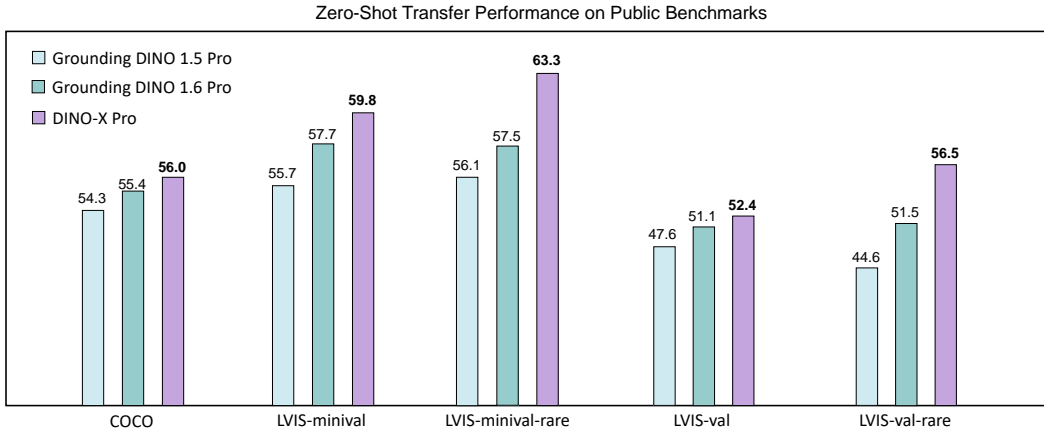


Figure 2: DINO-X Pro zero-shot performance on public detection benchmarks. Comparing with Grounding DINO 1.5 Pro and Grounding DINO 1.6 Pro, DINO-X Pro achieves new state-of-the-art (SOTA) performance on COCO, LVIS-minival, and LVIS-val zero-shot benchmarks. Furthermore, it outperforms other models with larger margins in detecting rare classes of objects on LVIS-minival and LVIS-val, demonstrating its exceptional capability of recognizing long-tailed objects.

2.1 Model Architecture

The overall framework of DINO-X is shown in Fig. 3. Following Grounding DINO 1.5, we also develop two variants of DINO-X models: a more powerful and comprehensive "Pro" version, DINO-X Pro, as well as a faster "Edge" version, termed DINO-X Edge, which will be introduced in details in Sections 2.1.1 and 2.1.2, respectively.

2.1.1 DINO-X Pro

The core architecture of the DINO-X Pro model is similar to Grounding DINO 1.5 [47]. We utilize a pre-trained ViT [12] model as its primary vision backbone and employ a deep early fusion strategy during the feature extraction stage. Different from Grounding DINO 1.5, to further extend the model's

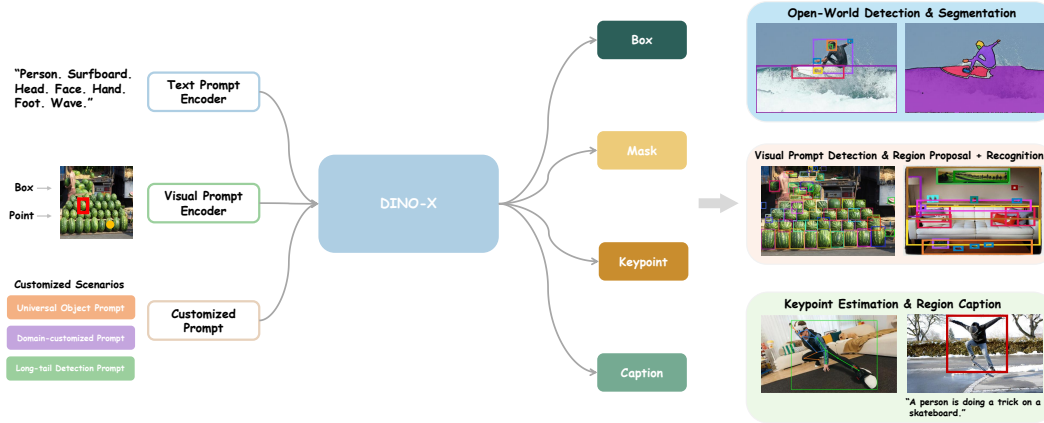


Figure 3: DINO-X is designed to accept text prompt, visual prompt, and customized prompt, and is capable of simultaneously generating outputs ranging from coarse-level representations, such as bounding boxes, to fine-grained details, including masks, keypoints, and object captions.

capability of detecting long-tailed objects, we have broadened the prompt support in DINO-X Pro at the input stage. Besides text prompts, we extend DINO-X Pro to also support visual prompts and customized prompts to cover various detection needs. Text prompts can cover the majority of object detection scenarios commonly encountered in daily life, while visual prompts enhance the model’s detection capability in situations where text prompts fall short due to data scarcity and descriptive limitations [18]. Customized prompts are defined as a series of specialized prompts that can be fine-tuned through prompt-tuning [26] techniques to expand the model’s ability to detect objects in more long-tailed, domain-specific, or function-specific scenarios without compromising other capabilities. By performing large-scale grounding pre-training, we obtain a foundational object-level representation from the encoder output of DINO-X. Such a robust representation enables us to seamlessly support multiple object perception or understanding tasks by introducing different perception heads. As a result, DINO-X is capable of generating outputs across different semantic levels, ranging from coarse-level, such as bounding boxes, to more fine-grained level, including masks, keypoints, and object captions.

We will first introduce the supported prompts in DINO-X in the following paragraphs.

Text Prompt Encoder: Both Grounding DINO [33] and Grounding DINO 1.5 [47] employ BERT [9] as text encoder. However, the BERT model is trained solely on text data, which limits its effectiveness for perception tasks requiring multimodal alignment, such as open-world detection. Therefore, in DINO-X Pro, we utilize a pre-trained CLIP [65] model as our text encoder, which has pre-trained on extensive multimodal data, thereby further enhancing the model’s training efficiency and performance across various open-world benchmarks.

Visual Prompt Encoder: We adopt the visual prompt encoder from T-Rex2 [18], integrating it to enhance object detection by utilizing user-defined visual prompts in both box and point formats. These prompts are converted into position embeddings using a sine-cosine layer and then projected into a unified feature space. The model separates box and point prompts using different linear projections. Then we employ the same multi-scale deformable cross-attention layers as in T-Rex2 to extract visual prompt features from multi-scale feature maps, conditioned on the user-provided visual prompts.

Customized Prompt: In practical use cases, it is common to encounter the need for fine-tuning models for customized scenarios. In DINO-X Pro, we define a series of specialized prompts, termed customized prompt, which can be fine-tuned through prompt-tuning [26] techniques to cover more long-tailed, domain-specific, or function-specific scenarios in a resource-efficient and cost-effective manner without compromising other capabilities. For instance, we developed a universal object prompt to support *prompt-free* open-world detection, making it possible to detect any objects within an image, thereby expanding its potential applications in areas such as screen parsing [35], etc.

Given an input image and a user-provided prompt, no matter it is textual, visual, or a customized prompt embedding, DINO-X performs deep feature fusion between the prompt and the visual features extracted from the input image and then apply different heads for different perception tasks. More specifically, the implemented heads are introduced in the following paragraphs.

Box Head: Following Grounding DINO [33], we adopt the language-guided query selection module to select features that are most relevant to the input prompt as decoder object queries. Each query is then fed into the Transformer decoder and updated layer-by-layer, followed by a simple MLP layer that predicts the corresponding bounding box coordinates for each object query. Similar to Grounding DINO, we employ L1 loss and G-IoU [49] loss for bounding box regression, while using contrastive loss to align each object query with the input prompt for classification.

Mask Head: Following the core design of Mask2Former [4] and Mask DINO [28], we construct the pixel embedding map by fusing the 1/4 resolution backbone feature and the upsampled 1/8 resolution feature from the Transformer encoder. Then we perform dot-product between each object query from the Transformer decoder and the pixel embedding map to get the mask output of the query. In order to improve the training efficiency, the 1/4 resolution feature map from the backbone was only used in mask prediction. And we also follow [24, 4] to only compute the mask loss for sampled points in the final mask loss calculation.

Keypoint Head: The keypoint head takes keypoint-related detection outputs from DINO-X, e.g. person or hand, as input and utilize a separate decoder to decode object keypoints. Each detection output is treated as a query and expanded into a number of keypoints, which are then sent to multiple deformable Transformer decoder layers to predict the desired keypoint positions and their visibilities. This process can be regarded as a simplified ED-Pose [68] algorithm, which does not need to consider the object detection task but only focuses on keypoint detection. In DINO-X, we instantiate two keypoint heads for person and hand, which have 17 and 21 pre-defined keypoints, respectively.

Language Head: The language head is a task-promptable generative small language model to enhance DINO-X’s ability to comprehend regional context and perform perception tasks beyond localization, such as object recognition, region captioning, text recognition, and region-based visual question answering (VQA). The architecture of our model is depicted in Figure 4. For any detected object from DINO-X, we first extract its region features from the DINO-X backbone features using the RoIAlign [15] operator, combined with its query embedding to form our object tokens. Then, we apply a simple linear projection to ensure their dimensions aligned with the text embedding. The lightweight language decoder integrates these regional representations with task tokens to generate outputs in an auto-regressive manner. The learnable task tokens empower the language decoder to handle a variety of tasks.

2.1.2 DINO-X Edge

Following Grounding DINO 1.5 Edge [47], DINO-X Edge also utilizes EfficientViT [1] as backbone for efficient feature extraction and incorporates a similar Transformer encoder-decoder architecture. To further enhance DINO-X Edge model’s performance and computational efficiency, we employ several improvements to the model architecture and training techniques in the following aspects:

Stronger Text Prompt Encoder: To achieve more effective region-level multi-modal alignment, DINO-X Edge adopts the same CLIP text encoder as our Pro model. In practice, text prompt embeddings can be pre-computed for most cases and do not affect the inference speed of the visual encoder and decoder. Using a stronger text prompt encoder generally leads to better results.

Knowledge Distillation: In DINO-X Edge, we distill the knowledge from the Pro model to enhance the Edge model’s performance. Specifically, we utilize both feature-based distillation and response-based distillation, which align the feature and prediction logits between the Edge model and the Pro model, respectively. This knowledge transfer enables DINO-X Edge to achieve a stronger zero-shot capability compared to Grounding DINO 1.6 Edge.

Improved FP16 Inference: We employ a normalization technique for floating-point multiplication, enabling model quantization into FP16 without compromising accuracy. This results in an inference

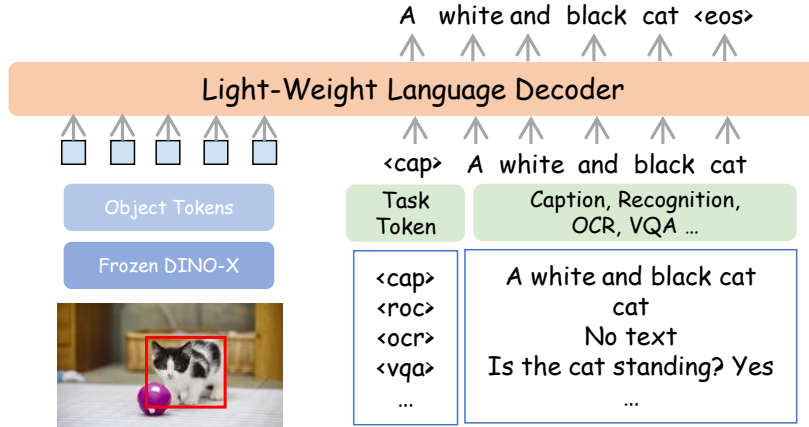


Figure 4: The detailed design of language head in DINO-X. It involves using a frozen DINO-X to extract object tokens, and a linear projection aligns its dimensions with the text embeddings. The lightweight language decoder then integrates these object and task tokens to generate response outputs in an autoregressive manner. The task tokens equip the language decoder with the capability of tackling different tasks.

speed of 20.1 FPS, a 33% increase from 15.1 FPS compared to Grounding DINO 1.6 Edge, and a 87% improvement from 10.7 FPS compared to Grounding DINO 1.5 Edge.

3 Dataset Construction and Model Training

Data Collection: To ensure the core open-vocabulary object detection capability, we developed a high-quality and semantic-rich grounding dataset, which consists of over 100 million images collected from the web, termed Grounding-100M. We used the training data from T-Rex2 with some additional industrial scenario data for visual prompt-based grounding pre-training. We used open-source segmentation models, such as SAM [23] and SAM2 [46], to generate pseudo mask annotations for a portion of the Grounding-100M dataset, which serves as the main training data for our mask head. we sampled a subset of high-quality data from the Grounding-100M dataset and utilized their box annotations as our *prompt-free* detection training data. We also collected over 10 million region understanding data, covering object recognition, region captioning, OCR, and region-level QA scenarios for language head training.

Model Training: To overcome the challenge of training multiple vision tasks, we adopt a two-stage strategy. In the first stage, we conducted joint training for text-prompt-based detection, visual-prompt-based detection, and object segmentation. In this training phase, we did not incorporate any images or annotations from COCO [32], LVIS [14], and V3Det [57] datasets, so that we can evaluate the model’s zero-shot detection performance on these benchmarks. Such a large-scale grounding pre-training ensures an outstanding open-vocabulary grounding performance of DINO-X and results in a foundational object-level representation. In the second stage, we froze the DINO-X backbone and added two keypoint heads (for person and hand) and a language head, each being trained separately. By adding more heads, we greatly expand DINO-X’s ability to perform more fine-grained perception and understanding tasks, such as pose estimation, region captioning, object-based QA, etc. Subsequently, we leveraged prompt-tuning techniques and trained a universal object prompt, allowing for prompt-free any-object detection while preserving the model’s other capabilities. Such a two-stage training approach has several advantages: (1) it ensures that the model’s core grounding capability is not affected by introducing new abilities, and (2) it also validates that large-scale grounding pre-training can serve as a robust foundation for an object-centric model, allowing for seamless transfer to other open-world understanding tasks.

4 Evaluation

In this section, we compare the various capabilities of our DINO-X series model with its related works. The best and the second best results are indicated in **bold** and with underline

4.1 DINO-X Pro

4.1.1 Open-World Detection and Segmentation

Evaluation on Zero-Shot Object Detection and Segmentation Benchmarks: Following Grounding DINO 1.5 Pro [47], we evaluate the zero-shot object detection and segmentation capability of DINO-X Pro on the COCO [32] benchmark, which includes 80 common categories, and the LVIS benchmark, which features a richer and more extensive long-tail distribution of categories. As shown in Table 1, DINO-X Pro shows a significant performance improvement compared to previous state-of-the-art methods. Specifically, on the COCO benchmark, DINO-X Pro achieves an increase of 1.7 box AP and 0.6 box AP compared to Grounding DINO 1.5 Pro and Grounding DINO 1.6 Pro, respectively. On the LVIS-minival and LVIS-val benchmarks, DINO-X Pro achieves 59.8 box AP and 52.4 box AP, respectively, surpassing the previously best-performing Grounding DINO 1.6 Pro model by 2.0 AP and 1.1 AP, respectively. Notably, for the detection performance on LVIS rare classes, DINO-X achieves 63.3 AP on LVIS-minival and 56.5 AP on LVIS-val, significantly surpassing the previous SOTA Grounding DINO 1.6 Pro model by 5.8 AP and 5.0 AP, respectively, demonstrating the exceptional capability of DINO-X in long-tailed object detection scenarios. In terms of segmentation metrics, we compared DINO-X with the most commonly used general segmentation model, Grounded SAM [48] series, on the COCO and LVIS zero-shot instance segmentation benchmarks. Using Grounding DINO 1.5 Pro for zero-shot detection and SAM-Huge [23] for segmentation, Grounded SAM achieves the best zero-shot performance on the LVIS instance segmentation benchmarks. DINO-X achieves mask AP scores of 37.9, 43.8, and 38.5 on the COCO, LVIS-minival, and LVIS-val zero-shot instance segmentation benchmarks, respectively. Compared to Grounded SAM, there is still a notable performance gap for DINO-X to catch up, which shows the challenge of training a unified model for multiple tasks. Nevertheless, DINO-X significantly improves the segmentation efficiency by generating corresponding masks for each region without requiring multiple complex inference steps. We will further optimize the performance of the mask head in our future work.

Evaluation on Visual-Prompt Based Detection Benchmarks: To assess the visual prompt object detection capability of DINO-X, we conduct experiments on the few-shot object counting benchmarks. In this task, each test image is accompanied by three visual exemplar boxes representing the target object, and the model is required to output the count of the target object. We evaluate the performance using the FSC147 [45] and FSCD-LVIS [40] datasets, which both feature scenes densely populated with small objects. Specifically, FSC147 primarily consists of single-target scenes, where only one type of object is present per image, whereas FSCD-LVIS focuses on multi-target scenes containing multiple object categories. For FSC147, we report the Mean Absolute Error (MAE) metric, and for FSCD-LVIS, we use the Average Precision (AP) metric. Following prior work [17, 18], the visual exemplar boxes are employed as interactive visual prompts. As shown in Table 2, DINO-X achieves state-of-the-art performance, demonstrating its strong capability in practical visual prompt object detection.

4.1.2 Keypoint Detection

Evaluation on Human 2D Keypoint Benchmarks: We present a comparison of DINO-X with other related works on the COCO [32], CrowdPose [52], and Human-Art [20] benchmarks, as shown in Table 3. We employ the OKS-based Average Precision (AP) [52] as the main metrics. Note that the pose head was trained jointly on MSCOCO, CrowdPose, and Human-Art. Hence the evaluation is not a zero-shot setting. But as we froze the backbone of DINO-X and trained only the pose head, the evaluation on object detection and segmentation still follows the zero-shot setting. Training on multiple pose datasets, our model can effectively predicts keypoints across various person styles, including everyday scenarios, crowded environments, occlusions, and artistic representations. While our model achieves an AP that is 1.6 lower than ED-Pose (primarily due to the limited number of trainable parameters in the pose head), it outperforms existing models on CrowdPose and Human-Art

Table 1: The performance of DINO-X Pro on the COCO, LVIS-minival and LVIS-val benchmarks compared to previous methods. Gray numbers indicate that the training dataset includes images or annotations from the COCO or LVIS datasets.

Method	Backbone	COCO-val		LVIS-minival							LVIS-val									
				Box AP			Mask AP				Box AP			Mask AP						
		AP _{box}	AP _{mask}	AP _{all}	AP _r	AP _c	AP _f	AP _{all}	AP _r	AP _c	AP _f	AP _{all}	AP _r	AP _c	AP _f	AP _{all}	AP _r	AP _c	AP _f	
<i>Supervised Model (Pretraining data includes COCO, LVIS, etc.)</i>																				
GLIPv2 [76]	Swin-H	60.6	-	50.1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
Grounding DINO [33]	Swin-L	60.7	-	33.9	22.2	30.7	38.8	-	-	-	-	-	-	-	-	-	-	-	-	
APE (B) [51]	ViT-L	57.7	48.6	62.5	-	-	-	55.4	-	-	-	-	57.0	-	-	-	50.5	-	-	
APE (D) [51]	ViT-L	58.3	49.3	64.7	-	-	-	57.5	-	-	-	-	59.6	-	-	-	53.0	-	-	
GLEE-Pro [63]	ViT-L	62.0	54.2	-	-	-	-	-	-	-	-	-	55.7	49.2	-	-	49.9	44.3	-	
DINOv [27]	Swin-T	47.0	42.7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
DINOv [27]	Swin-L	54.2	50.4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
<i>Zero-shot Transfer Model</i>																				
OWL-ViT [39]	ViT-L	42.2	-	-	-	-	-	-	-	-	-	-	34.6	31.2	-	-	-	-	-	
MDETR [21]	ResNet101	-	-	22.5	7.4	22.7	25.0	-	-	-	-	-	-	-	-	-	-	-	-	
GLIP [29]	Swin-L	49.8	-	37.3	28.2	34.3	41.5	-	-	-	-	-	26.9	17.1	23.3	35.4	-	-	-	
Grounding DINO [33]	Swin-T	48.4	-	27.4	18.1	23.3	32.7	-	-	-	-	-	-	-	-	-	-	-	-	
Grounding DINO [33]	Swin-L	52.5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
OpenSeeD [75]	Swin-L	-	-	23.0	-	-	-	21.0	-	-	-	-	-	-	-	-	-	-	-	
UniDetector [61]	ResNet50	-	-	-	-	-	-	-	-	-	-	-	19.8	18.0	19.2	21.2	-	-	-	
OmDet-Turbo-B [79]	ConvNeXt-B	53.4	-	34.7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
OWL-ST [38]	CLIP L/14	-	-	40.9	41.5	-	-	-	-	-	-	-	35.2	36.2	-	-	-	-	-	
MQ-GLIP [66]	Swin-L	-	-	43.4	34.5	41.2	46.9	-	-	-	-	-	34.7	26.9	32.0	41.3	-	-	-	
MM-Grounding-DINO [80]	Swin-T	50.4	-	41.4	34.2	37.4	46.2	-	-	-	-	-	31.9	23.6	27.6	40.5	-	-	-	
MM-Grounding-DINO [80]	Swin-L	53.0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
DetCLIP [70]	Swin-L	-	-	38.6	36.0	38.3	39.3	-	-	-	-	-	28.4	25.0	27.0	31.6	-	-	-	
DetCLIPv2 [69]	Swin-L	-	-	44.7	43.1	46.3	43.7	-	-	-	-	-	36.6	33.3	36.2	38.5	-	-	-	
DetCLIPv3 [71]	Swin-L	-	-	48.8	49.9	49.7	47.8	-	-	-	-	-	41.4	41.4	40.5	42.3	-	-	-	
YOLO-World [6]	YOLOv8-L	45.1	-	35.4	27.6	34.1	38.0	-	-	-	-	-	-	-	-	-	-	-	-	
OV-DINO [56]	Swin-T	50.2	-	40.1	34.5	39.5	41.5	-	-	-	-	-	32.9	29.1	30.4	37.4	-	-	-	
T-Rex2 (visual) [18]	Swin-L	46.5	-	47.6	45.4	46.0	49.5	-	-	-	-	-	45.3	43.8	42.0	49.5	-	-	-	
T-Rex2 (text) [18]	Swin-L	52.2	-	54.9	49.2	54.8	<u>56.1</u>	-	-	-	-	-	45.8	42.7	43.2	<u>50.2</u>	-	-	-	
<i>Assembled General Perception Model</i>																				
SAM (ViTDet-H prompt) [23]	-	-	-	46.5	-	-	-	-	-	-	-	-	-	-	-	-	44.7	-	-	
Grounded SAM (1.5 Pro + Huge) [48, 23]	-	-	-	<u>44.3</u>	-	-	-	47.7	50.2	51.7	43.8	-	-	-	-	-	41.8	46.0	42.3	39.5
Grounded SAM 2 (1.5 Pro + Large) [48, 23]	-	-	-	44.7	-	-	-	<u>46.2</u>	<u>50.1</u>	<u>50.1</u>	<u>42.0</u>	-	-	-	-	-	<u>40.5</u>	<u>44.6</u>	<u>41.0</u>	<u>38.1</u>
<i>Object-Centric Vision Model</i>																				
Grounding DINO 1.5 Pro [47]	ViT-L	54.3	-	55.7	56.1	57.5	54.1	-	-	-	-	-	47.6	44.6	47.9	48.7	-	-	-	
Grounding DINO 1.6 Pro [47]	ViT-L	<u>55.4</u>	-	<u>57.7</u>	<u>57.5</u>	<u>60.5</u>	55.3	-	-	-	-	-	<u>51.1</u>	<u>51.5</u>	52.0	50.1	-	-	-	
DINO-X Pro	ViT-L	56.0	37.9	59.8	63.3	61.7	57.5	43.8	46.7	47.5	40.0	52.4	56.5	<u>51.1</u>	51.9	38.5	44.4	38.4	36.1	

Table 2: The performance of DINO-X Pro on few-shot object counting benchmarks.

Type	Method	FSC147-test		FSCD-LVIS-test
		MAE	RMSE	AP
Density Map Regression	FamNet [45]	22.1	99.5	-
	BMNet+ [53]	14.6	91.8	-
	Counting-DETR [40]	12.0	49.8	22.7
Detection	T-Rex [17]	<u>8.72</u>	-	40.3
	T-Rex2 [18]	10.9	<u>36.7</u>	<u>43.4</u>
	DINO-X Pro	5.6	27.4	44.8

by 3.4 AP and 1.8 AP, respectively, showing its remarkable generalization ability on more diverse scenarios.

Evaluation on Human Hand 2D Keypoint Benchmarks: In addition to evaluating human pose, we also present hand pose results on the HInt benchmark [42] with Percentage of Correctly Localized Keypoints (PCK) as the measurement. PCK is a metric used to evaluate the accuracy of keypoint localization. A keypoint is considered correct if the distance between its predicted and ground truth locations is below a specified threshold. We use a threshold of 0.05 box size, *i.e.* PCK@0.05. During training, we combine the HInt, COCO, and OneHand10K [59] training dataset (a subset of the compared method HaMer [42]), and evaluate the performance on the HInt test set. As shown in Table 4, DINO-X achieves the best performance on the PCK@0.05 metrics, indicating its strong capability on highly accurate hand pose estimation.

Table 3: Comparisons with state-of-the-art methods on COCO-val, CrowdPose-test, and Human-Art-val benchmarks. † denotes the flipping test. The OKS-based Average Precision (AP) is employed as evaluation metric on the datasets. **TD**, **BU**, **OS**, **PT** mean top-down, bottom-up, one-stage and pre-trained methods, respectively.

Method	Type	COCO-val			CrowdPose-test			Human-Art-val		
		AP	AP ₅₀	AP ₇₅	AP	AP ₅₀	AP ₇₅	AP	AP ₅₀	AP ₇₅
Sim.Base.[64]†	TD	70.4	88.6	78.3	60.8	81.4	65.7	-	-	-
HRNet[54]†		74.4	90.5	81.9	71.3	91.1	77.5	39.9	54.5	42.0
HrHRNet[5]†	BU	67.1	86.2	73.0	65.9	86.4	70.6	34.6	-	-
DEKR[13]†		68.0	86.7	74.5	65.7	85.7	70.4	-	-	-
SWAHR[36]†		68.9	87.8	74.9	71.6	88.5	77.6	-	-	-
PETR[52]†	OS	64.8	85.1	70.2	71.6	90.4	78.3	-	-	-
ED-Pose[68]		75.8	92.3	82.9	<u>76.6</u>	<u>92.4</u>	<u>83.3</u>	72.3	-	-
DINO-X Pro	PT	<u>74.4</u>	<u>90.7</u>	<u>81.1</u>	80.0	88.0	84.4	74.1	90.7	81.1

Table 4: Comparisons with state-of-the-art methods on HInt dataset. We use PCK@0.05 as the main metrics.

Method	All joints			Visible joints			Occluded joints		
	New Days	VISOR	Ego4D	New Days	VISOR	Ego4D	New Days	VISOR	Ego4D
FrankMocap [50]	16.1	16.8	13.1	20.1	20.4	16.3	9.2	11.0	8.4
METRO [30]	14.7	16.8	13.2	19.2	19.7	15.8	7.0	10.2	8.1
Mesh Graphormer [31]	16.8	19.1	14.6	22.3	23.6	18.4	7.9	10.9	8.3
HandOccNet (param) [41]	9.1	8.1	7.7	10.2	8.5	7.3	7.2	7.4	8.0
HandOccNet (no param) [41]	13.7	12.4	10.9	15.7	13.1	11.2	9.8	9.9	9.6
ViTPose-Hands [67]	32.2	40.0	23.3	44.0	55.7	35.0	13.9	21.2	10.3
Hamba [10]	48.7	47.2	-	61.2	61.4	-	28.2	29.9	-
HaMeR [42]	<u>51.6</u>	<u>56.5</u>	<u>46.9</u>	<u>62.9</u>	<u>66.5</u>	<u>59.1</u>	<u>33.2</u>	<u>42.6</u>	<u>33.1</u>
DINO-X Pro	54.3	63.0	66.0	69.3	78.0	81.1	34.4	48.0	49.1

4.1.3 Object-Level Vision-Language Understanding

Evaluation on Object Recognition: We verify the effectiveness of our language head with related works on object recognition benchmarks, which need to recognize the category of the object in a specified region of an image. Following Osprey[73], we use Semantic Similarity (SS) and Semantic IoU (S-IoU)[8], to evaluate the object recognition capability of the language head on the object-level LVIS-val[14] and the part-level PACO-val[44] datasets. As shown in Table 5, Our model achieves 71.25% in SS and 41.15% in S-IoU, surpassing Osprey by 6.01% in SS and 2.06% in S-IoU on the LVIS-val dataset. On the PACO dataset, our model is inferior to Osprey. Note that we did not include LVIS and PACO in our language head training and the performance of our model is achieved in a zero-shot manner. The lower performance on PACO might be due to the discrepancy between our training data and PACO. And our model only has 1% trainable parameters compared with Osprey.

Table 5: Results on referring object classification benchmarks. We use Semantic Similarity (SS) and Semantic-IoU (S-IoU) scores to measure the region classification quality.

Method	Visual Encoder	Language Decoder	LVIS		PACO	
			SS	S-IoU	SS	S-IoU
Kosmos-2 [43]	ViT-L	LM-1.3B [43]	38.95	8.67	32.09	4.79
Shikra [2]	ViT-L	Vicuna-7B[7]	49.65	19.82	43.64	11.42
GPT4RoI [77]	ViT-L	Vicuna-7B[7]	51.32	11.99	48.04	12.08
Ferret [72]	ViT-L	Vicuna-7B[7]	63.78	36.57	58.68	25.96
Osprey [73]	ConvNeXt-L	Vicuna-7B[7]	<u>65.24</u>	<u>38.19</u>	73.06	52.72
DINO-X Pro	ViT-L	OPT-125M[78]	71.25	41.15	<u>66.67</u>	<u>39.39</u>

Table 6: Results on region captioning benchmarks. We report METEOR and CIDEr scores to measure the region caption quality.

Method	Visual Encoder	Language Decoder	Visual Genome		RefCOCOg	
			CIDEr	METEOR	CIDEr	METEOR
GRIT [62]	ViT-B	Small-43M [62]	142.0	17.2	71.6	15.2
GPT4RoI [77]	ViT-L	Vicuna-7B[7]	145.2	17.4	-	-
ASM [58]	ViT-G	Husky-7B[22]	145.1	18.0	<u>103.0</u>	20.8
AlphaCLIP [55]	ViT-L	Vicuna-7B[7]	<u>160.3</u>	<u>18.9</u>	109.2	<u>16.7</u>
SCA [16]	SAM-H	Llama-3B[11]	149.8	17.4	74.0	15.6
DINO-X Pro (zero-shot)	ViT-L	OPT-125M[78]	143.2	17.5	55.7	12.2
DINO-X Pro (fine-tuned)	ViT-L	OPT-125M[78]	201.8	20.1	86.3	15.1

Evaluation on Region Captioning: We evaluate our model’s region caption quality on Visual Genome[25] and RefCOCOg[37]. The evaluation results are presented in Table 6. Remarkably, based on object-level features extracted by a frozen DINO-X backbone and without utilizing any Visual Genome training data, our model achieves a 142.1 CIDEr score on the Visual Genome benchmark in a zero-shot manner. Further, after fine-tuning on Visual Genome dataset, we set a new state-of-the-art result with 201.8 CIDEr score with only a light-weight language head.

4.2 DINO-X Edge

Table 7: Zero-shot Performance of DINO-X Edge on COCO, LVIS-minival, and LVIS-val object detection benchmarks compared with related works.

Method	Backbone	Test Size	COCO-val		LVIS-minival				LVIS-val				FPS (A100)		FPS (Orin NX)	
			AP _{box}	AP _{all}	AP _r	AP _c	AP _f	AP _{all}	AP _r	AP _c	AP _f	Pytorch/TensorRT	FP32	FP16	TensorRT	FP32/FP16
<i>End-to-End Open-Set Object Detection</i>																
GLIP [29]	Swin-T [34]	800 × 1333	46.3	26.0	20.8	21.4	31.0	-	-	-	-	-	-	-	-	-
Grounding DINO [33]	Swin-T [34]	800 × 1333	48.4	27.4	18.1	23.3	32.7	-	-	-	-	9.4 / 42.6	-	1.1/-	-	
<i>Real-time End-to-End Open-Set Object Detection Models</i>																
YOLO-Worldv2-S† [6]	YOLOv8-S [19]	640 × 640	-	22.7	16.3	20.8	25.5	17.3	11.3	14.9	22.7	47.4 / -	-	-	-	
YOLO-Worldv2-M† [6]	YOLOv8-M [19]	640 × 640	-	30.0	25.0	27.2	33.4	23.5	17.1	20.0	30.1	42.7 / -	-	-	-	
YOLO-Worldv2-L† [6]	YOLOv8-L [19]	640 × 640	-	33.0	22.6	32.0	35.8	26.0	18.6	23.0	32.6	37.4 / -	-	-	-	
YOLO-Worldv2-L† [6]	YOLOv8-L [19]	640 × 640	-	32.9	25.3	31.1	35.8	26.1	20.6	22.6	32.3	37.4 / -	-	-	-	
OmDet-Turbo-T [79]	Swin-T [34]	640 × 640	42.5	30.3	-	-	-	-	-	-	-	21.5 / 140.0	-	-	-	
OVLW-DETR-L [60]	LW-DETR-L [3]	640 × 640	-	33.5	26.5	33.9	34.4	-	-	-	-	- / -	-	-	-	
<i>Efficient Object-Centric Vision Model</i>																
Grounding DINO 1.5 Edge [47]	EfficientViT-L1 [1]	640 × 640	42.9	33.5	28.0	34.3	33.9	27.3	26.3	25.7	29.6	21.7 / 111.6	-	10.7/-	-	
Grounding DINO 1.5 Edge [47]	EfficientViT-L1 [1]	800 × 1333	45.0	36.2	33.2	36.6	36.3	29.3	28.1	27.6	31.6	18.5 / 75.2	-	5.5/-	-	
Grounding DINO 1.6 Edge [47]	EfficientViT-L1 [1]	800 × 800	44.8	36.9	34.6	39.1	35.4	31.0	31.6	30.5	31.4	20.81/152.7	-	10.0/15.1	-	
Grounding DINO 1.6 Edge [47]	EfficientViT-L1 [1]	1024 × 1024	46.5	40.1	36.8	42.0	39.0	33.3	32.6	32.8	34.3	19.4/108.1	-	7.6/10.5	-	
DINO-X Edge	EfficientViT-L2 [1]	640 × 640	48.7	<u>44.5</u>	<u>41.4</u>	<u>47.3</u>	<u>42.6</u>	<u>38.4</u>	<u>38.9</u>	<u>38.3</u>	<u>38.2</u>	19.8/138.6	-	10.0/20.1	-	
DINO-X Edge	EfficientViT-L2 [1]	800 × 1333	50.9	48.3	47.6	50.2	46.6	42.0	43.1	41.7	41.8	15.1/74.5	-	4.5/9.1	-	

Evaluation on Zero-Shot Object Detection Benchmarks: To evaluate the zero-shot object detection capability of DINO-X Edge, we conduct tests on the COCO and LVIS benchmarks after pre-training on Grounding-100M. As shown in Table 7, DINO-X Edge outperforms existing real-time open-set detectors on COCO benchmark by a large margin. DINO-X Edge also achieves 48.3 AP and 42.0 AP on LVIS-minival and LVIS-val, respectively, demonstrating excellent zero-shot detection capability in long-tailed detection scenarios.

We evaluate the inference speed DINO-X Edge using both FP32 and FP16 TensorRT models on NVIDIA Orin NX, measuring the performance in terms of frames per second (FPS). The FPS results for the PyTorch model and the FP32 TensorRT model on an A100 GPU were also included. †denotes that the YOLO-World results were reproduced using the latest official codes.

Leveraging the normalization technique in floating-point multiplication, we can quantize the model to FP16 without sacrificing the performance. With an input size of 640×640, DINO-X Edge achieves an inference speed of 20.1 FPS, marking a 33% improvement compared to Grounding DINO 1.6 Edge (increasing from 15.1 FPS to 20.1 FPS).

5 Case Analysis and Qualitative Visualization

In this section, we visualize the different capabilities of DINO-X models across various real-world scenarios. The images are primarily sourced from COCO [32], LVIS [14], V3Det [57], SA-1B [23], and other publicly available resources. We are deeply grateful for their contributions, which have significantly benefited the community.

5.1 Open-World Object Detection

As illustrated in Figure 5, DINO-X demonstrates the capability to detect any objects based on the given text prompt. It can identify a wide range of objects, from common categories to long-tailed classes and dense object scenarios, showcasing its robust open-world object detection capabilities.

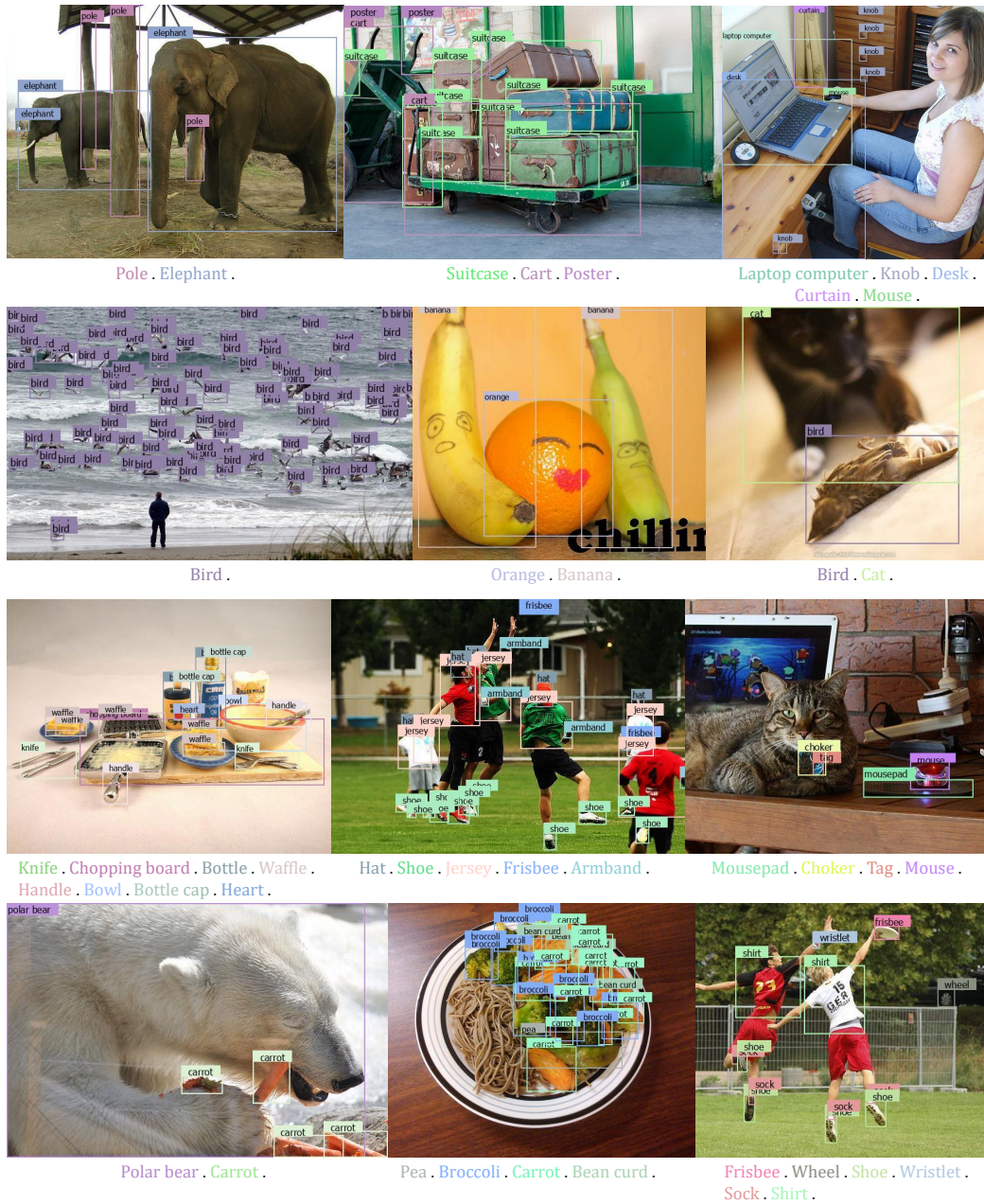


Figure 5: Open-world object detection with DINO-X

5.3 Open-World Object Segmentation and Visual Prompt Counting

As shown in Figure 7, beyond Grounding DINO 1.5 [47], DINO-X not only enables open-world object detection based on text prompts but also generates the corresponding segmentation mask for each object, providing richer semantic outputs. Furthermore, DINO-X also supports detection based on user-defined visual prompts by drawing bounding boxes or points on target objects. This capability demonstrates exceptional usability in object counting scenarios.

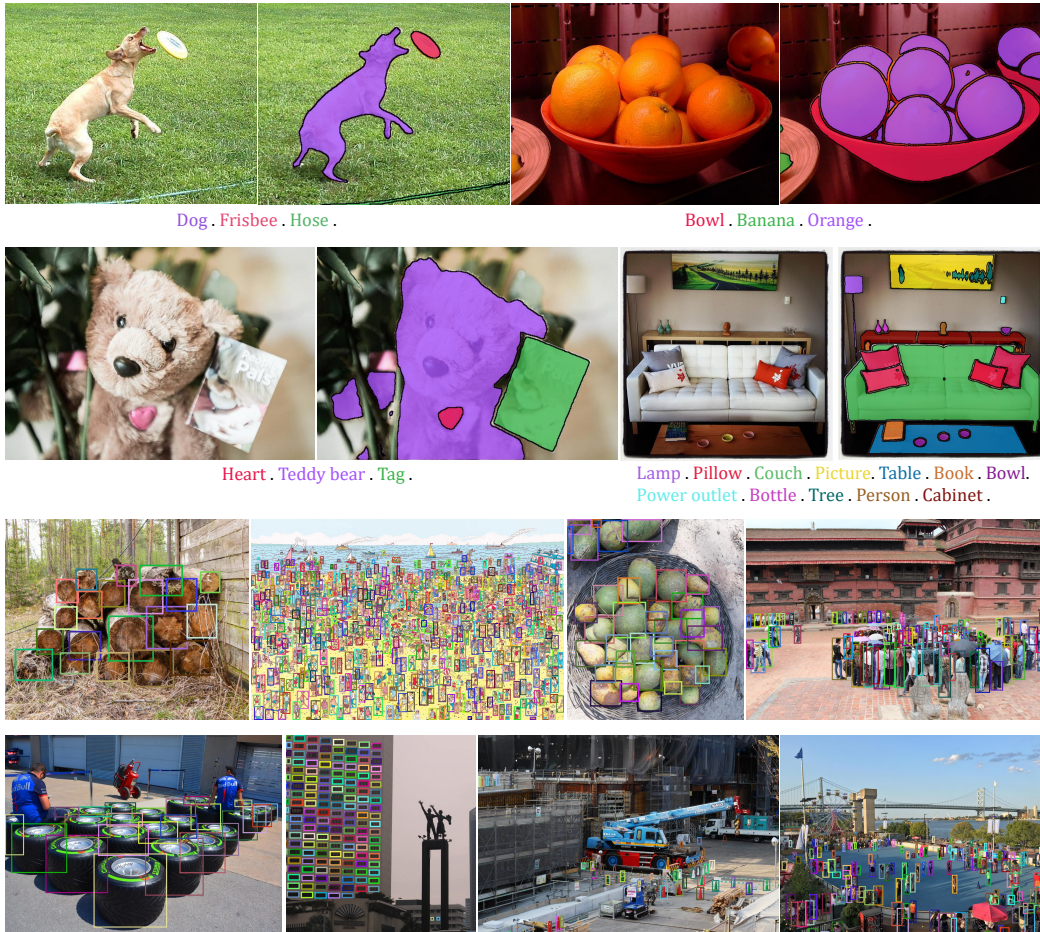


Figure 7: Open-world object segmentation and visual prompt object counting with DINO-X

5.4 Prompt-Free Object Detection and Recognition

In DINO-X, we developed a highly practical feature named *prompt-free* object detection, which allows users to detect any objects in an input image without providing any prompts. As shown in Figure 8 When combined with DINO-X’s language head, this feature enables seamless detection and identification of all objects in the image without requiring any user input.

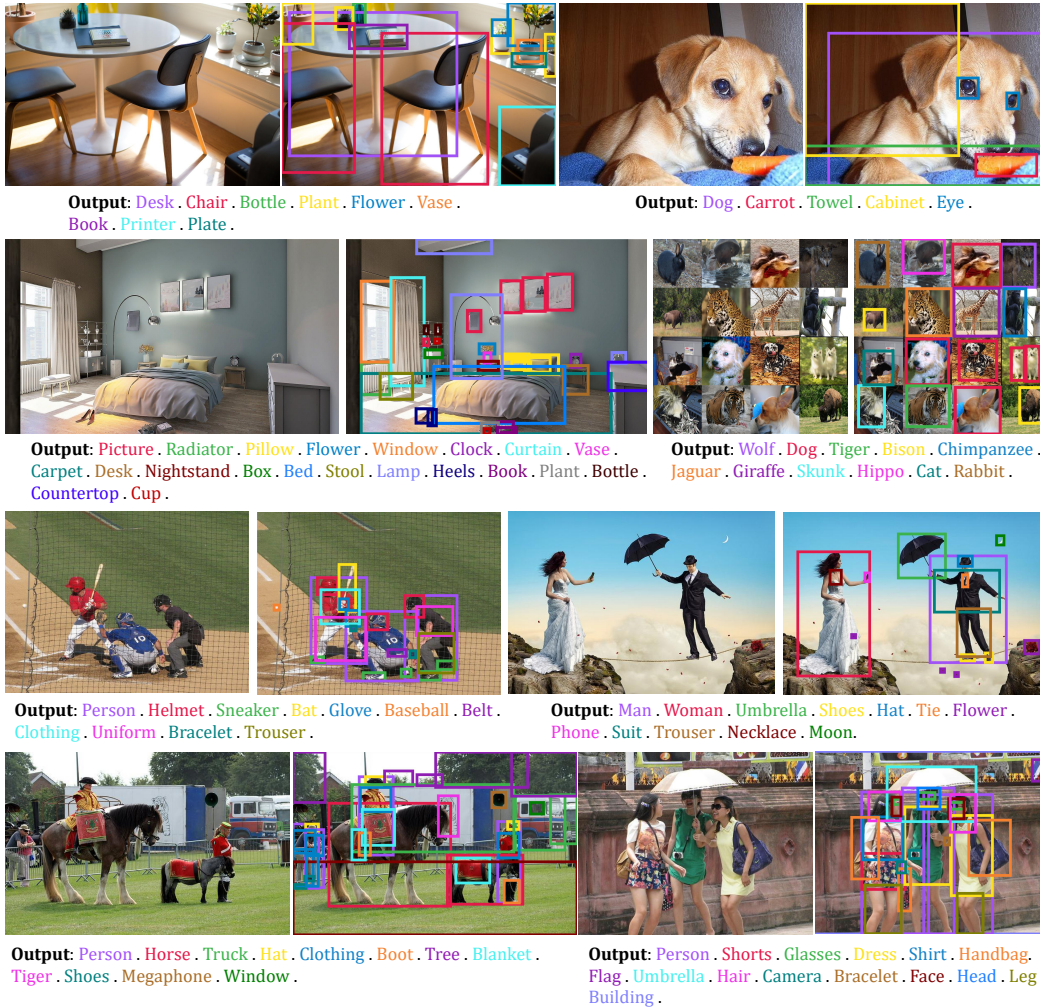


Figure 8: Prompt-free object detection and recognition with DINO-X

5.5 Dense Region Caption

As illustrated in Figure 9, DINO-X can generate more fine-grained captions for any specified region. Furthermore, with DINO-X's language head, we can also perform tasks such as region-based QA and other region understanding tasks. Currently, this feature is still in the development stage and will be released in our next version.



Figure 9: Dense Region Caption with DINO-X

5.6 Human Body and Hand Pose Estimation

As shown in Figure 10, DINO-X can predict keypoints for specific categories through the keypoint heads based on the text prompts. Trained on a combination of COCO, CrowdHuman, and Human-Art datasets, DINO-X is capable of predicting human body and hand keypoints across various scenarios.

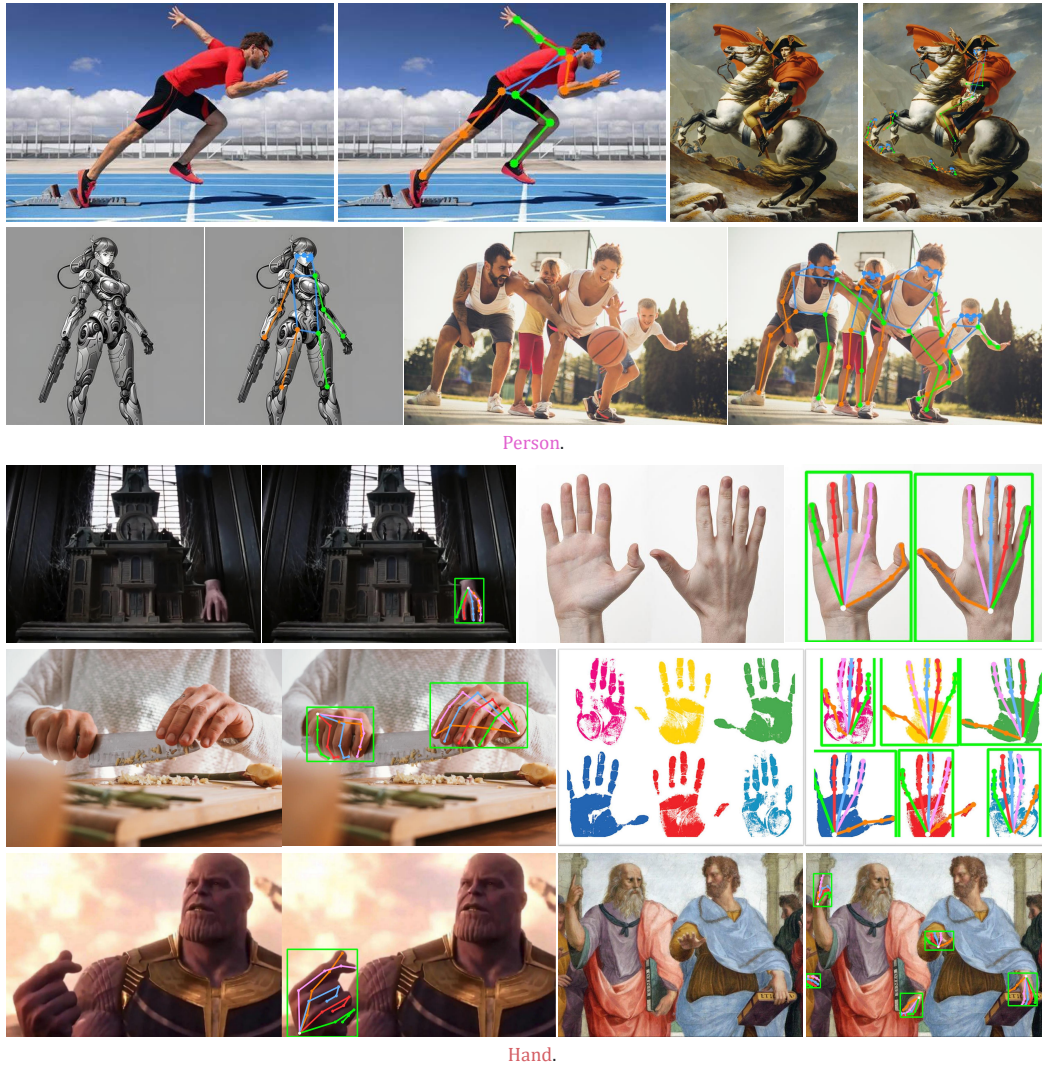


Figure 10: Pose estimation on human body and human hand with DINO-X

5.7 Side-by-side comparison with Grounding DINO 1.5 Pro

We conducted a side-by-side comparison of DINO-X with previous state-of-the-art models, Grounding DINO 1.5 Pro and Grounding DINO 1.6 Pro. As shown in Figure 11, built upon the foundation of Grounding DINO 1.5, DINO-X further enhances its language comprehension capabilities while delivering a remarkable performance in dense object detection scenarios.

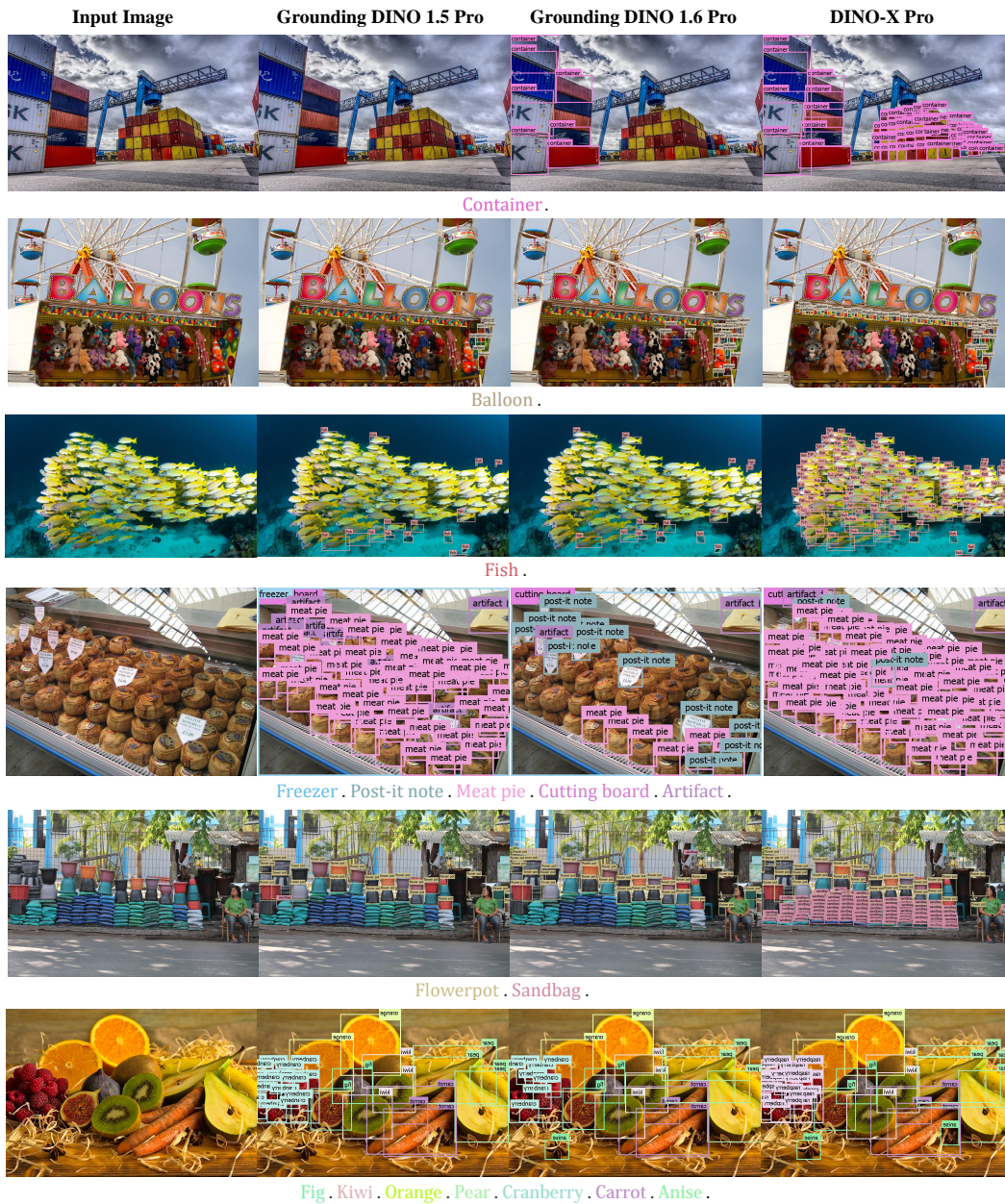


Figure 11: Comparison of Grounding DINO 1.5 Pro, Grounding DINO 1.6 Pro and DINO-X

6 Conclusion

This paper has presented DINO-X, a strong object-centric vision model to advance the field of open-set object detection and understanding. The flagship model, DINO-X Pro, has established new records on the COCO and LVIS zero-shot benchmarks, showing a remarkable improvement in detection accuracy and reliability. To make long-tailed object detection easy, DINO-X not only supports open-world detection based on text prompts but also enables object detection with visual prompts and customized prompts for customized scenarios. Moreover, DINO-X extends its capabilities from detection to a broader range of perception tasks, including segmentation, pose estimation, and object-level understanding tasks. To enable real-time object detection for more applications on edge devices, we also developed the DINO-X Edge model, which further expands the practical utility of the DINO-X series models.

7 Contributions and Acknowledgments

We would like to express our gratitude to everyone involved in the DINO-X project. The contributions are as follows (in no particular order):

- **DINO-X Pro:** Yihao Chen, Tianhe Ren, Qing Jiang, Zhaoyang Zeng, and Yuda Xiong.
- **Mask Head:** Tianhe Ren, Hao Zhang, Feng Li, and Zhaoyang Zeng.
- **Visual Prompt & Prompt-Free Detection:** Qing Jiang.
- **Pose Head:** Xiaoke Jiang, Xingyu Chen, Zhuheng Song, and Yuhong Zhang.
- **Language Head:** Wenlong Liu, Zhengyu Ma, Junyi Shen, Yuan Gao, and Yuda Xiong.
- **DINO-X Edge:** Hongjie Huang, Han Gao, and Qing Jiang.
- **Grounding-100M:** Yuda Xiong, Yihao Chen, Tianhe Ren, Qing Jiang, Zhaoyang Zeng, and Shilong Liu.
- **Language Head and DINO-X Edge Lead:** Kent Yu.
- **Overall Project Lead:** Lei Zhang.

We would also like to thank everyone involved in the DINO-X playground and API support, including application lead Wei Liu, product manager Qin Liu and Xiaohui Wang, front-end developers Yuanhao Zhu, Ce Feng, and Jiongrong Fan, back-end developers Zhiqiang Li and Jiawei Shi, UX designer Zijun Deng, operation intern Weijian Zeng, tester Jiangyan Wang, and Peng Xiao for providing suggestions and feedbacks on customized scenarios.

References

- [1] Han Cai, Junyan Li, Muyan Hu, Chuang Gan, and Song Han. EfficientViT: Lightweight Multi-Scale Attention for High-Resolution Dense Prediction. *ICCV*, 2023. 5, 10
- [2] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing Multimodal LLM’s Referential Dialogue Magic. *arXiv preprint arXiv:2306.15195*, 2023. 9
- [3] Qiang Chen, Xiangbo Su, Xinyu Zhang, Jian Wang, Jiahui Chen, Yunpeng Shen, Chuchu Han, Ziliang Chen, Weixiang Xu, Fanrong Li, et al. LW-DETR: A Transformer Replacement to YOLO for Real-Time Detection. *arXiv preprint arXiv:2406.03459*, 2024. 10
- [4] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention Mask Transformer for Universal Image Segmentation. *CVPR*, 2022. 2, 5
- [5] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S Huang, and Lei Zhang. HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose Estimation. *CVPR*, 2020. 9
- [6] Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. YOLO-World: Real-Time Open-Vocabulary Object Detection. *CVPR*, 2024. 8, 10
- [7] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality, 2023. 9, 10
- [8] Alessandro Conti, Enrico Fini, Massimiliano Mancini, Paolo Rota, Yiming Wang, and Elisa Ricci. Vocabulary-free Image Classification. *NeurIPS*, 2023. 9
- [9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL*, 2019. 4
- [10] Haoye Dong, Aviral Chharia, Wenbo Gou, Francisco Vicente Carrasco, and Fernando De la Torre. Hamba: Single-view 3D Hand Reconstruction with Graph-guided Bi-Scanning Mamba. *NeurIPS*, 2024. 9
- [11] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*, 2024. 10
- [12] Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. EVA-02: A Visual Representation for Neon Genesis. *arXiv preprint arXiv:2303.11331*, 2023. 3
- [13] Zigang Geng, Ke Sun, Bin Xiao, Zhaoxiang Zhang, and Jingdong Wang. Bottom-Up Human Pose Estimation Via Disentangled Keypoint Regression. *CVPR*, 2021. 9
- [14] Agrim Gupta, Piotr Dollár, and Ross Girshick. LVIS: A Dataset for Large Vocabulary Instance Segmentation. *CVPR*, 2019. 6, 9, 11
- [15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. *ICCV*, 2017. 5
- [16] Xiaoke Huang, Jianfeng Wang, Yansong Tang, Zheng Zhang, Han Hu, Jiwen Lu, Lijuan Wang, and Zicheng Liu. Segment and Caption Anything. *CVPR*, 2024. 10
- [17] Qing Jiang, Feng Li, Tianhe Ren, Shilong Liu, Zhaoyang Zeng, Kent Yu, and Lei Zhang. T-Rex: Counting by Visual Prompting. *arXiv preprint arXiv:2311.13596*, 2023. 7, 8
- [18] Qing Jiang, Feng Li, Zhaoyang Zeng, Tianhe Ren, Shilong Liu, and Lei Zhang. T-Rex2: Towards Generic Object Detection via Text-Visual Prompt Synergy. *ECCV*, 2024. 2, 4, 7, 8
- [19] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics YOLOv8. 2023. 10
- [20] Xuan Ju, Ailing Zeng, Jianan Wang, Qiang Xu, and Lei Zhang. Human-Art: A Versatile Human-Centric Dataset Bridging Natural and Artificial Scenes. *CVPR*, 2023. 7
- [21] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. MDETR - Modulated Detection for End-to-End Multi-Modal Understanding. *ICCV*, 2021. 8
- [22] Joongwon Kim, Bhargavi Paranjape, Tushar Khot, and Hannaneh Hajishirzi. Husky: A Unified, Open-Source Language Agent for Multi-Step Reasoning. *arXiv preprint arXiv:2406.06469*, 2024. 10

- [23] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment Anything. *ICCV*, 2023. 6, 7, 8, 11
- [24] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. PointRend: Image Segmentation as Rendering. *CVPR*, 2020. 5
- [25] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *IJCV*, 2017. 10
- [26] Brian Lester, Rami Al-Rfou, and Noah Constant. The Power of Scale for Parameter-Efficient Prompt Tuning. *EMNLP*, 2021. 4
- [27] Feng Li, Qing Jiang, Hao Zhang, Tianhe Ren, Shilong Liu, Xueyan Zou, Huaizhe Xu, Hongyang Li, Chunyuan Li, Jianwei Yang, et al. Visual In-Context Prompting. *CVPR*, 2024. 8
- [28] Feng Li, Hao Zhang, Huaizhe xu, Shilong Liu, Lei Zhang, Lionel M. Ni, and Heung-Yeung Shum. Mask DINO: Towards A Unified Transformer-based Framework for Object Detection and Segmentation. *CVPR*, 2023. 2, 5
- [29] Liunian Harold Li*, Pengchuan Zhang*, Haotian Zhang*, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. Grounded Language-Image Pre-training. *CVPR*, 2022. 2, 8, 10
- [30] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-End Human Pose and Mesh Reconstruction with Transformers. *CVPR*, 2021. 9
- [31] Kevin Lin, Lijuan Wang, and Zicheng Liu. Mesh Graphormer. *ICCV*, 2021. 9
- [32] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common Objects in Context. *ECCV*, 2014. 6, 7, 11
- [33] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, 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. 2, 4, 5, 8, 10
- [34] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. *ICCV*, 2021. 10
- [35] Yadong Lu, Jianwei Yang, Yelong Shen, and Ahmed Awadallah. OmniParser for Pure Vision Based GUI Agent. *arXiv preprint arXiv:2408.00203*, 2024. 4
- [36] Zhengxiong Luo, Zhicheng Wang, Yan Huang, Tieniu Tan, and Erjin Zhou. Rethinking the Heatmap Regression for Bottom-up Human Pose Estimation. *CVPR*, 2021. 9
- [37] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and Comprehension of Unambiguous Object Descriptions. *CVPR*, 2016. 10
- [38] Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling Open-Vocabulary Object Detection. *NeurIPS*, 2023. 8
- [39] 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. Simple Open-Vocabulary Object Detection with Vision Transformers. *ECCV*, 2022. 8
- [40] Thanh Nguyen, Chau Pham, Khoi Nguyen, and Minh Hoai. Few-shot Object Counting and Detection. *ECCV*, 2022. 7, 8
- [41] JoonKyu Park, Yeonguk Oh, Gyeongsik Moon, Hongsuk Choi, and Kyoung Mu Lee. HandOccNet: Occlusion-Robust 3D Hand Mesh Estimation Network. *CVPR*, 2022. 9
- [42] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing Hands in 3D with Transformers. *CVPR*, 2024. 8, 9
- [43] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Grounding Multimodal Large Language Models to the World. *ICLR*, 2024. 9

- [44] Vignesh Ramanathan, Anmol Kalia, Vladan Petrovic, Yi Wen, Baixue Zheng, Baishan Guo, Rui Wang, Aaron Marquez, Rama Kovvuri, Abhishek Kadian, et al. PACO: Parts and Attributes of Common Objects. *CVPR*, 2023. 9
- [45] Viresh Ranjan, Udbhav Sharma, Thu Nguyen, and Minh Hoai. Learning to Count Everything. *CVPR*, 2021. 7, 8
- [46] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. SAM 2: Segment Anything in Images and Videos. *arXiv preprint arXiv:2408.00714*, 2024. 6
- [47] Tianhe Ren, Qing Jiang, Shilong Liu, Zhaoyang Zeng, Wenlong Liu, Han Gao, Hongjie Huang, Zhengyu Ma, Xiaoke Jiang, Yihao Chen, Yuda Xiong, Hao Zhang, Feng Li, Peijun Tang, Kent Yu, and Lei Zhang. Grounding DINO 1.5: Advance the "Edge" of Open-Set Object Detection. *arXiv preprint arXiv:2405.10300*, 2024. 2, 3, 4, 5, 7, 8, 10, 13
- [48] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded SAM: Assembling Open-World Models for Diverse Visual Tasks. *arXiv preprint arXiv:2401.14159*, 2024. 7, 8
- [49] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression. *CVPR*, 2019. 5
- [50] Yu Rong, Takaaki Shiratori, and Hanbyul Joo. FrankMocap: A Monocular 3D Whole-Body Pose Estimation System via Regression and Integration. *ICCV*, 2021. 9
- [51] Yunhang Shen, Chaoyou Fu, Peixian Chen, Mengdan Zhang, Ke Li, Xing Sun, Yunsheng Wu, Shaohui Lin, and Rongrong Ji. Aligning and Prompting Everything All at Once for Universal Visual Perception. *CVPR*, 2024. 8
- [52] Dahu Shi, Xing Wei, Liangqi Li, Ye Ren, and Wenming Tan. End-to-End Multi-Person Pose Estimation with Transformers. *CVPR*, 2022. 7, 9
- [53] Min Shi, Hao Lu, Chen Feng, Chengxin Liu, and Zhiguo Cao. Represent, Compare, and Learn: A Similarity-Aware Framework for Class-Agnostic Counting. *CVPR*, 2022. 8
- [54] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep High-Resolution Representation Learning for Human Pose Estimation. *CVPR*, 2019. 9
- [55] Zeyi Sun, Ye Fang, Tong Wu, Pan Zhang, Yuhang Zang, Shu Kong, Yuanjun Xiong, Dahua Lin, and Jiaqi Wang. Alpha-CLIP: A CLIP Model Focusing on Wherever You Want. *CVPR*, 2024. 10
- [56] Hao Wang, Pengzhen Ren, Zequn Jie, Xiao Dong, Chengjian Feng, Yinlong Qian, Lin Ma, Dongmei Jiang, Yaowei Wang, Xiangyuan Lan, and Xiaodan Liang. OV-DINO: Unified Open-Vocabulary Detection with Language-Aware Selective Fusion. *arXiv preprint arXiv:2407.07844*, 2024. 8
- [57] Jiaqi Wang, Pan Zhang, Tao Chu, Yuhang Cao, Yujie Zhou, Tong Wu, Bin Wang, Conghui He, and Dahua Lin. V3Det: Vast Vocabulary Visual Detection Dataset. *ICCV*, 2023. 6, 11
- [58] Weiyun Wang, Min Shi, Qingyun Li, Wenhai Wang, Zhenhang Huang, Linjie Xing, Zhe Chen, Hao Li, Xizhou Zhu, Zhiguo Cao, et al. The All-Seeing Project: Towards Panoptic Visual Recognition and Understanding of the Open World. *ICLR*, 2024. 10
- [59] Yangang Wang, Cong Peng, and Yebin Liu. Mask-Pose Cascaded CNN for 2D Hand Pose Estimation From Single Color Image. *TCSVT*, 2018. 8
- [60] Yu Wang, Xiangbo Su, Qiang Chen, Xinyu Zhang, Teng Xi, Kun Yao, Errui Ding, Gang Zhang, and Jingdong Wang. OVLW-DETR: Open-Vocabulary Light-Weighted Detection Transformer. *arXiv preprint arXiv:2407.10655*, 2024. 10
- [61] Zhenyu Wang, Yali Li, Xi Chen, Ser-Nam Lim, Antonio Torralba, Hengshuang Zhao, and Shengjin Wang. Detecting Everything in the Open World: Towards Universal Object Detection. *CVPR*, 2023. 8
- [62] Jialian Wu, Jianfeng Wang, Zhengyuan Yang, Zhe Gan, Zicheng Liu, Junsong Yuan, and Lijuan Wang. GRiT: A Generative Region-to-text Transformer for Object Understanding. *ECCV*, 2024. 10

- [63] Junfeng Wu, Yi Jiang, Qihao Liu, Zehuan Yuan, Xiang Bai, and Song Bai. General Object Foundation Model for Images and Videos at Scale. *CVPR*, 2024. 8
- [64] Bin Xiao, Haiping Wu, and Yichen Wei. Simple Baselines for Human Pose Estimation and Tracking. *ECCV*, 2018. 9
- [65] Hu Xu, Saining Xie, Xiaoqing Ellen Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. Demystifying CLIP Data. *ICLR*, 2024. 4
- [66] Yifan Xu, Mengdan Zhang, Chaoyou Fu, Peixian Chen, Xiaoshan Yang, Ke Li, and Changsheng Xu. Multi-modal Queried Object Detection in the Wild. *NeurIPS*, 2023. 8
- [67] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. ViTPose: Simple Vision Transformer Baselines for Human Pose Estimation. *NeurIPS*, 2022. 9
- [68] Jie Yang, Ailing Zeng, Shilong Liu, Feng Li, Ruimao Zhang, and Lei Zhang. Explicit Box Detection Unifies End-to-End Multi-Person Pose Estimation. *ICLR*, 2023. 5, 9
- [69] Lewei Yao, Jianhua Han, Xiaodan Liang, Dan Xu, Wei Zhang, Zhenguo Li, and Hang Xu. Det-CLIPv2: Scalable Open-Vocabulary Object Detection Pre-training via Word-Region Alignment. *CVPR*, 2023. 8
- [70] Lewei Yao, Jianhua Han, Youpeng Wen, Xiaodan Liang, Dan Xu, Wei Zhang, Zhenguo Li, Chunjing Xu, and Hang Xu. DetCLIP: Dictionary-Enriched Visual-Concept Paralleled Pre-training for Open-world Detection. *NeurIPS*, 2022. 8
- [71] Lewei Yao, Renjie Pi, Jianhua Han, Xiaodan Liang, Hang Xu, Wei Zhang, Zhenguo Li, and Dan Xu. DetCLIPv3: Towards Versatile Generative Open-vocabulary Object Detection. *CVPR*, 2024. 8
- [72] Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. Ferret: Refer and Ground Anything Anywhere at Any Granularity. *ICLR*, 2024. 9
- [73] Yuqian Yuan, Wentong Li, Jian Liu, Dongqi Tang, Xinjie Luo, Chi Qin, Lei Zhang, and Jianke Zhu. Osprey: Pixel Understanding with Visual Instruction Tuning. *CVPR*, 2024. 9
- [74] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M. Ni, and Heung-Yeung Shum. DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection. *ICLR*, 2023. 2
- [75] Hao Zhang, Feng Li, Xueyan Zou, Shilong Liu, Chunyuan Li, Jianwei Yang, and Lei Zhang. A Simple Framework for Open-Vocabulary Segmentation and Detection. *ICCV*, 2023. 8
- [76] Haotian Zhang, Pengchuan Zhang, Xiaowei Hu, Yen-Chun Chen, Liunian Harold Li, Xiyang Dai, Lijuan Wang, Lu Yuan, Jenq-Neng Hwang, and Jianfeng Gao. GLIPv2: Unifying Localization and Vision-Language Understanding. *NeurIPS*, 2022. 2, 8
- [77] Shilong Zhang, Peize Sun, Shoufa Chen, Min Xiao, Wenqi Shao, Wenwei Zhang, Kai Chen, and Ping Luo. GPT4RoI: Instruction Tuning Large Language Model on Region-of-Interest. *arXiv preprint arXiv:2307.03601*, 2023. 9, 10
- [78] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. OPT: Open Pre-trained Transformer Language Models. *arXiv preprint arXiv:2205.01068*, 2022. 9, 10
- [79] Tiancheng Zhao, Peng Liu, Xuan He, Lu Zhang, and Kyusong Lee. Real-time Transformer-based Open-Vocabulary Detection with Efficient Fusion Head. *arXiv preprint arXiv:2403.06892*, 2024. 8, 10
- [80] Xiangyu Zhao, Yicheng Chen, Shilin Xu, Xiangtai Li, Xinjiang Wang, Yining Li, and Haiyan Huang. An Open and Comprehensive Pipeline for Unified Object Grounding and Detection. *arXiv preprint arXiv:2401.02361*, 2024. 8