T-Rex2: Towards Generic Object Detection via Text-Visual Prompt Synergy

Qing Jiang[1,2§], Feng Li[2,3§], Zhaoyang Zeng[2], Tianhe Ren[2], Shilong Liu[2,4§], and Lei Zhang[1,2†]

[1] South China University of Technology
[2] International Digital Economy Academy (IDEA)
[3] The Hong Kong University of Science and Technology
[4] Tsinghua University

mountchicken@outlook.com , leizhang@idea.edu.cn

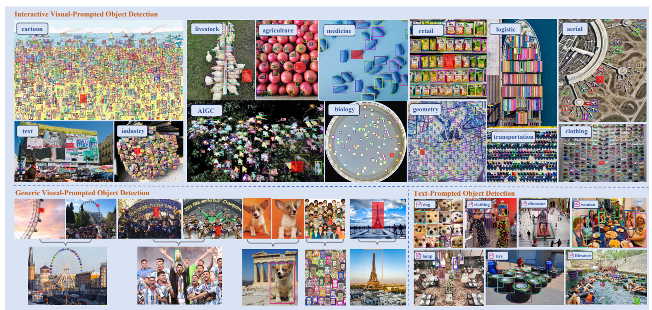


Fig. 1: *T-Rex2* is a promptable and interactive model for open-set object detection.

Abstract. We present *T-Rex2*, a highly practical model for open-set object detection. Previous open-set object detection methods relying on text prompts effectively encapsulate the abstract concept of common objects, but struggle with rare or complex object representation due to data scarcity and descriptive limitations. Conversely, visual prompts excel in depicting novel objects through concrete visual examples, but fall short in conveying the abstract concept of objects as effectively as text prompts. Recognizing the complementary strengths and weaknesses of both text and visual prompts, we introduce *T-Rex2* that synergizes both prompts within a single model through contrastive learning. *T-Rex2* accepts inputs in diverse formats, including text prompts, visual prompts, and the combination of both, so that it can handle different scenarios by switching between the two prompt modalities. Comprehensive experiments demonstrate that *T-Rex2* exhibits remarkable zero-shot object detection capabilities across a wide spectrum of scenarios. We show that text prompts and visual prompts can benefit from each other within the synergy, which is essential to cover massive and complicated real-world scenarios and pave the way towards generic object detection. API code is available at `https://github.com/IDEA-Research/T-Rex`.

Keywords: Open-Set Object Detection · Visual Prompt · Text Prompt.

---

§*This work was done when Qing, Feng, and Shilong were interns at IDEA.*
† *Corresponding author.*

1 Introduction

Object detection, a foundational pillar of computer vision, aims to locate and identify objects within an image. Traditionally, object detection was operated within a closed-set paradigm [1, 6, 17, 21, 24, 36, 37, 44, 50, 55, 56], wherein a predefined set of categories is known a prior, and the system is trained to recognize and detect objects from this set. Yet the ever-changing and unforeseeable nature of the real world demands a shift in object detection methodologies towards an open-set paradigm.

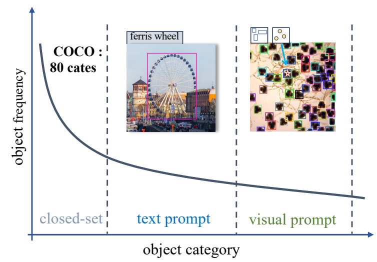


Fig. 2: Long-tailed curve between object frequency and the number of categories that can be detected.

Open-set object detection represents a significant paradigm shift, transcending the limitations of closed-set detection by empowering models to identify objects beyond a predetermined set of categories. A prevalent approach is to use text prompts for open-vocabulary object detection [5, 7, 11, 20, 25, 30, 54]. This approach typically involves distilling knowledge from language models like CLIP [34] or BERT [3] to align textual descriptions with visual representations.

While using text prompts has been predominantly favored in open-set detection for their capacity to abstractly describe objects, it still faces the following limitations. 1) *Long-tailed data shortage*. The training of text prompts necessitates modality alignment between visual representations, however, the scarcity of data for long-tailed objects may impair the learning efficiency. As depicted in Fig. 2, the distribution of objects inherently follows a long-tail pattern, i.e., as the variety of detectable objects increases, the available data for these objects becomes increasingly scarce. This data scarcity may undermine the capacity of models to identify rare or novel objects. 2) *Descriptive limitations*. Text prompts also fall short of accurately depicting objects that are hard to describe in language. For instance, as shown in Fig. 2, while a text prompt may effectively describe `ferris wheel`, it may struggle to accurately represent `the microorganisms in the microscope image` without biological knowledge.

Conversely, visual prompts [10, 12, 16, 18, 19, 30, 45, 49, 57] provide a more intuitive and direct method to represent objects by providing visual examples. For example, users can use visual prompts, such as points or boxes to mark the object for detection, even if they do not know what the object is. Additionally, visual prompts are not constrained by the need for cross-modal alignment, since they rely on visual similarities rather than linguistic correlations, enabling their application to novel objects that are not encountered during training.

Nonetheless, visual prompts also exhibit limitations, as they are less effective at capturing the general concept of objects compared to text prompts. For instance, the term `dog` as a text prompt broadly covers all dog varieties. In contrast, visual prompts, given the vast diversity in dog breeds, sizes, and colors, would necessitate a comprehensive image collection to visually convey the abstract notion of `dog` adequately.

Recognizing the complementary strengths and weaknesses of both text and visual prompts, we introduce *T-Rex2*, a generic open-set object detection model that integrates both modalities. *T-Rex2* is built upon the DETR [1] architecture which is an end-to-end object detection model. It incorporates two parallel encoders for both text and visual prompts. For text prompts, we utilize the text encoder of CLIP [34] to encode input text into text embedding. For visual prompts, we introduce a novel visual prompt encoder equipped with the deformable attention mechanism [56] that can transform the input visual prompts (points or boxes) on a single image or across multiple images into visual prompt embeddings. To facilitate the collaborative operation of these two prompt modalities, we propose a contrastive learning [9, 34] module that can explicitly align text prompts and visual prompts. During the alignment, visual prompts can benefit from the generality and abstraction capabilities inherent in text prompts. Conversely, text prompts can enhance their descriptive capabilities by looking at various visual prompts. This iterative interaction allows both visual and text prompts to evolve continuously, thereby improving their ability for generic understanding within one model.

T-Rex2 supports four unique workflows that can be applied to various scenarios: 1) *interactive visual prompt workflow*, allowing users to specify the object to be detected by given visual examples through boxes or points on the current image; 2) *generic visual prompt workflow*, permitting users to define specific objects across multiple images through visual prompts, thereby creating universal visual embeddings applicable to other images; 3) *text prompt workflow*, enabling users to employ descriptive text for open-vocabulary object detection; 4) *mix prompt workflow*, which combines both text and visual prompts for joint inference.

T-Rex2 demonstrates strong object detection capabilities and achieves remarkable results on COCO [22], LVIS [8], ODinW [15] and Roboflow100 [2], all under zero-shot setting. Through our analysis, we observe that text and visual prompts serve complementary roles, each excelling in scenarios where the other may not be as effective. Specifically, text prompts are particularly good at recognizing common objects, while visual prompts excel in rare objects or scenarios that may not be easily described through language. This complementary relationship enables the model to perform effectively across a wide range of scenarios. To summarize, our contributions are threefold:

- We propose an open-set object detection model *T-Rex2* that unifies text and visual prompts within one framework, which demonstrates strong zero-shot capabilities across various scenarios.
- We propose a contrastive learning module to explicitly align text and visual prompts, which leads to mutual enhancement of these two modalities.
- Extensive experiments demonstrate the benefits of unifying text and visual prompts. We also reveal that text prompts can cover common object scenarios and visual prompts can cover rare or novel object scenarios, which collectively show promise in advancing towards generic object detection.

2 Related Work

2.1 Text-prompted Object Detection

Remarkable progress has been achieved in text-prompted object detection [7,11, 20,25,30,49,51,53], which demonstrate impressive zero-shot and few-shot recognition capability under diverse scenarios. These models are typically built upon a pre-trained text encoder like CLIP [34] and BERT [3]. GLIP [20] proposes to formulate object detection as grounding problems, which unifies different data formats to align different modalities and expand detection vocabulary. Following GLIP, Grounding DINO [25] improves the vision-language alignment by fusing different modalities in the early phase. DetCLIP [48] and RegionCLIP [53] leverages image-text pairs with pseudo boxes to expand region knowledge for more generalized object detection.

2.2 Visual-prompted Object Detection

Beyond text-prompted models, developing models incorporating visual prompts is a trending research area due to its flexibility and context-awareness. Mainstream visual-prompted models [30, 45, 49] adopt raw images as visual prompts and leverage image-text-aligned representation to transfer knowledge from text to visual prompts. However, it is restricted to image-level prompts and highly relies on aligned image-text foundation models. Another emergent approach for visual prompts is to use visual instructions like box, point, and referred region of another image. DINOv [16] proposes to use visual prompts as in-context examples for open-set detection and segmentation tasks. When detecting a novel category, it takes in several visual examples (marked with scribbles) of this category to understand this category in an in-context manner. In this paper, we focus on visual prompts in the form of visual instructions.

2.3 Interactive Object Detection

Interactive models have shown significant promise in aligning human intentions in the field of computer vision. It has been wildly applied for interactive segmentation [12, 18, 57], where the user provides a visual prompt (box, point, and mask, *etc.*) and the model outputs a mask corresponding to the prompt. This process typically follows a one-to-one interaction model, i.e., one prompt for one output mask. However, object detection requires a one-to-many approach, where a single visual prompt can lead to multiple detected boxes. Several works [14,46] have incorporated interactive object detection for the purpose of automating annotations. T-Rex [10] leverages interactive visual prompts for the task of object counting through object detection, however, its capabilities in generic object detection have not been extensively explored.

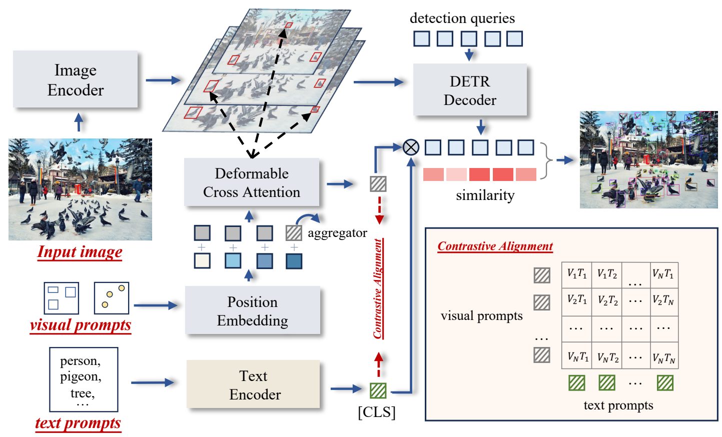


Fig. 3: Overview of the *T-Rex2* model. *T-Rex2* mainly follows the design principles of DETR [1] which is an end-to-end object detection model. Visual prompt and text prompt are introduced through deformable cross attention [56] and CLIP [34] text encoder, respectively, and are aligned through contrastive learning.

3 *T-Rex2* Model

T-Rex2 integrates four components, as illustrated in Fig. 3: i) Image Encoder, ii) Visual Prompt Encoder, iii) Text Prompt Encoder, and iv) Box Decoder. *T-Rex2* adheres to the design principles of DETR [1] which is an end-to-end object detection model. These four components collectively facilitate four distinct workflows that encompass a broad range of application scenarios.

3.1 Visual-Text Promptable Object Detection

Image Encoder. Mirroring the Deformable DETR [56] framework, the image encoder in *T-Rex2* consists of a vision backbone (e.g. Swin Transformer [26]) that extracts multi-scale feature maps from input image. This is followed by several transformer encoder layers [4] equipped with deformable self-attention [56], which are utilized to refine these extracted feature maps. The feature maps output from the image encoder is denoted as $\boldsymbol{f_i} \in \mathbb{R}^{C_i \times H_i \times W_i}, i \in \{1, 2, ..., L\}$, where L is the number of feature map layers.
Visual Prompt Encoder. Visual prompt has been widely used in interactive segmentation [12, 18, 57], yet to be fully explored within the domain of object detection. Our method incorporates visual prompts in both box and point formats. The design principle involves transforming user-specified visual prompts from their coordinate space to the image feature space. Given K user-specified 4D normalized boxes $b_j = (x_j, y_j, w_j, h_j), j \in \{1, 2, ..., K\}$, or 2D normalized points $p_j = (x_j, y_j), j \in \{1, 2, ..., K\}$ on a reference image.

$$, \tag{1}$$

we initially encode these coordinate inputs into position embeddings through a fixed sine-cosine embedding layer. Subsequently, two distinct linear layers are employed to project these embeddings into a uniform dimension:

$$B = \mathrm{Linear}(\mathrm{PE}(b_1, ...b_K); \theta_B) : \mathbb{R}^{K\times 4D} \to \mathbb{R}^{K\times D} \tag{2}$$

$$P = \mathrm{Linear}(\mathrm{PE}(p_1, ...p_K); \theta_P) : \mathbb{R}^{K\times 2D} \to \mathbb{R}^{K\times D} \tag{3}$$

where PE stands for position embedding and $\mathrm{Linear}(\cdot;\theta)$ indicate a linear project operation with parameter θ. Different from the previous method [18] that regards point as a box with minimal width and height, we model box and point as distinct prompt types. We then initiate a learnable content embedding that is broadcasted K times, denoted as $C \in \mathbb{R}^{K\times D}$. Additionally, a universal class token $C' \in \mathbb{R}^{1\times D}$ is utilized to aggregate features from other visual prompts, accommodating the scenario where users might supply multiple visual prompts within a single image. These content embeddings are concatenated with position embeddings along the channel dimension, and a linear layer is applied for projection, thereby constructing the input query embedding Q:

$$Q = \begin{cases} \mathrm{Linear}\left(\mathrm{CAT}\left(\left[C; C'\right], \left[B; B'\right]\right); \varphi_B\right) \in \mathbb{R}^{(K+1)\times D}, \text{for box input} \\ \mathrm{Linear}\left(\mathrm{CAT}\left(\left[C; C'\right], \left[P; P'\right]\right); \varphi_P\right) \in \mathbb{R}^{(K+1)\times D}, \text{for point input} \end{cases} \tag{4}$$

where notion CAT stands for concatenation at channel dimension. B' and P' represent global position embeddings, which are derived from global normalized coordinates $[0.5, 0.5, 1, 1]$ and $[0.5, 0.5]$. The global query serves the purpose of aggregating features from other queries. Subsequently, we employ a multi-scale deformable cross-attention [56] layer to extract visual prompt features from the multi-scale feature maps, conditioned on the visual prompts. For the j-th prompt, the query feature Q'_j after cross attention is computed as:

$$Q'_j = \begin{cases} \mathrm{MSDeformAttn}(Q_j, b_j, \{\boldsymbol{f}_i\}_{i=1}^{L}), \text{for box input} \\ \mathrm{MSDeformAttn}(Q_j, p_j, \{\boldsymbol{f}_i\}_{i=1}^{L}), \text{for point input} \end{cases} \tag{5}$$

Deformable attention [56] was initially employed to address the slow convergence problem encountered in DETR [1]. In our approach, we condition deformable attention on the coordinates of visual prompts, meaning that each query will selectively attend to a limited set of multi-scale image features encompassing the regions surrounding the visual prompts. This ensures the capture of visual prompt embeddings representing the objects of interest. Following the extraction process, we use a self-attention layer to regulate the relationships among different queries and a feed-forward layer for projection. The output of the global content query will be used as the final visual prompt embedding V.

$$V = \mathrm{FFN}(\mathrm{SelfAttn}(Q'))[-1] \tag{6}$$

Text Prompt Encoder. We employ the text encoder of CLIP [34] to encode category names or short phrases and use the `[CLS]` token output as the text prompt embedding, denoted as T.

Box Decoder. We employ a DETR-like decoder for box prediction. Following DINO [50], each query is formulated as a 4D anchor coordinate and undergoes iterative refinement across decoder layers. We employ the query selection layer proposed in Grounding DINO [25] to initialize the anchor coordinates. Specifically, We compute the similarity between the encoder feature and the prompt embeddings and select indices with similarity of top 900 to initialize the position embeddings. Subsequently, the detection queries utilize deformable cross-attention [56] to focus on the encoded multi-scale image features and are used to predict anchor offsets at each decoder layer. The final predicted boxes are obtained by summing the anchors and offsets. Instead of using a learnable linear layer to predict class labels, following previous methods [20, 25], we utilize the prompt embeddings as the weights for the classification layer.

3.2 Region-Level Contrastive Alignment

To integrate both visual prompts and text prompts within one model, we employ region-level contrastive learning to align these two modalities. Specifically, given an input image and K visual prompt embeddings $V = (v_1, ..., v_K)$ extracted from the visual prompt encoder, along with the text prompt embeddings $T = (t_1, ..., t_K)$ for each prompt region, we calculate the InfoNCE loss [32] between the two types of embeddings:

$$\mathcal{L}_{align} = -\frac{1}{K}\sum_{i=1}^{K} \log \frac{\exp(v_i \cdot t_i)}{\sum_{j=1}^{K} \exp(v_i \cdot t_j)} \tag{7}$$

The contrastive alignment can be regarded as a mutual distillation process, whereby each modality contributes to and benefits from the exchange of knowledge. Specifically, text prompts can be seen as a conceptual anchor, around which diverse visual prompts can converge so that the visual prompt can gain general knowledge. Conversely, the visual prompts act as a continuous source of refinement for text prompts. Through exposure to a wide array of visual instances, the text prompt is dynamically updated and enhanced, gaining depth and nuance.

3.3 Training Strategy and Objective

Visual prompt training strategy. For visual prompt training, we adopt the strategy of "current image prompt, current image detect". Specifically, for each category in a training set image, we randomly choose between one to all available GT boxes to use as visual prompts. We convert these GT boxes into their center point with a 50% chance for point prompt training. While using visual prompts from different images for cross-image detection training might seem more effective, creating such image pairs poses challenges in an open-set scenario due to inconsistent label spaces across datasets. Despite its simplicity, our straightforward training strategy still leads to strong generalization capability.
Text prompt training strategy. *T-Rex2* uses both detection data and grounding data for text prompt training. For detection data, we use the category names

in the current image as the positive text prompt and randomly sample negative text prompts in the remaining categories. For grounding data, we extract positive phrases corresponding to the bounding boxes and exclude other words in the caption for text input. Following the methodology of DetCLIP [47, 48], we maintain a global dictionary to sample negative text prompts for grounding data, which are concatenated with the positive text prompts. This global dictionary is constructed by selecting the category names and phrase names that occur more than 100 times in the text prompt training data.

Training objective. We employ the L1 loss and GIOU [38] loss for box regression. For classification loss, following Grounding DINO [25], we apply a contrastive loss that measures the difference between the predicted objects and the prompt embeddings. Specifically, we calculate the similarity between each detection query and the visual prompt or text prompt embeddings through a dot product to predict logits, followed by the computation of a sigmoid focal loss [21] for each logit. The box regression and classification loss are initially employed for bipartite matching [1] between predictions and ground truths. Subsequently, we calculate the final losses between ground truths and matched predictions, incorporating the same loss components. We use auxiliary loss after each decoder layer and after the encoder outputs. Following DINO [50], we also use denoising training to accelerate convergence. The final loss takes the following form:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{L1}} + \mathcal{L}_{\text{GIoU}} + \mathcal{L}_{\text{DN}} + \mathcal{L}_{\text{align}} \tag{8}$$

We adopt a cyclical training strategy that alternates between training for text prompts and visual prompts in successive iterations.

3.4 Four Inference Workflows

T-Rex2 offers four different workflows by combining text prompts and visual prompts in different ways.

Text prompt workflow. This workflow exclusively employs text prompts for object detection, which is the same as open-vocabulary object detection. This workflow is suitable for the detection of common objects, where the text prompt can provide clear descriptions.

Interactive visual prompt workflow. This workflow is designed around a core principle of user-driven interactivity with *T-Rex2*. Given the initial output of *T-Rex2* from the user-provided prompts, users can refine the detection results by adding additional prompts on missed or falsely-detected objects, based on the visualization result. This iterative cycle allows users to fine-tune *T-Rex2*'s performance interactively, ensuring precise detection. Notably, this interactive process remains fast and resource-efficient, as *T-Rex2* is a late fusion model that only requires the image encoder to forward once.

Generic visual prompt workflow. In this workflow, users have the flexibility to customize visual embeddings for specific objects by prompting *T-Rex2* with an arbitrary number of example images. This capability is crucial for generic object detection since a class of object may have very diverse instances, thus

we need a certain amount of visual examples to represent it. Let $V_1, V_2, ..., V_n$, represent the visual embeddings obtained from n different images, the generic visual embeddings V are computed as the mean of these embeddings.
Mixed prompt workflow. Benefit from the contrastive alignment, text prompt and visual prompt can also be used for inference at the same time. This is achieved simply by averaging their respective embeddings.

4 Experiments

4.1 Data Engines

For each modality, specialized data engines are designed to curate data.
Data engine for text prompt. By integrating CLIP [34] for text encoding, *T-Rex2* supports the integration of both detection and grounding data for training. Following [20,25], we utilize detection datasets Objects365 [40], OpenImages [13], along with the grounding dataset GoldG [11] for training. To enhance the text prompt capabilities of *T-Rex2*, we also make extensive use of pseudo-labeled data from image caption datasets and image classification datasets. Specifically, for image caption data in Conceptual Captions [42] and LAION400M [39] datasets, we use spaCy to extract noun chunks from image captions and use these noun chunks to prompt Grounding DINO [25] to get boxes. For image classification data in the Bamboo [52] dataset, we simply use the category of the current image to prompt Grounding DINO [25]. In total, we use 3.15M labeled images and 3.39M pseudo-labeled images for text prompt training.
Data engine for visual prompt. The training process for visual prompts is to use a portion of the GT box or its center point in the current image as the input. Thus we can leverage established detection datasets including Objects365 [40], OpenImages [13], HierText [27], CrowdHuman [41] for the initial training. Meanwhile, to make the data for visual prompt sufficiently diversified, we constructed a data engine to harvest data from SA-1B [12]. This data engine operates through a self-training loop, comprising two primary phases: **1) Initial training stage**: In this stage, we first train an initial version of *T-Rex2* with only visual prompt modality on the aforementioned datasets, endowing it with preliminary capabilities for interactive object detection. **2) Annotation stage**: With the initial model, we then utilize it to annotate the data in SA-1B. SA-1B has tremendous boxes for objects at all granularity. However, the box has no semantic labels, which is not suitable for object detection training. Thus, we employ TAP [33] to annotate each box with a category name from a dictionary of 2560 classes. We then adopt the following filtering strategy: if an image has at least one category with a number of instances greater than a certain threshold (e.g. 10), it is reserved. However in SA-1B, not all objects have boxes, so we use the original GT box as the interactive visual prompt input and use the initial *T-Rex2* to annotate the missing labeled boxes. In total, we use 2.4M labeled images and 0.65M pseudo-labeled images for visual prompt training.

https://spacy.io/

4.2 Model Details

T-Rex2 is built upon DINO [50], a DETR-based end-to-end detection model. We utilize Swin Transformer [26] as the vision backbone, followed by six layers of transformer encoder layers. We use CLIP-B [34] as the text encoder and fine-tune it. For the visual prompt encoder, we stack three layers of deformable cross-attention layer and set the hidden dimension of the feed-forward layer to 1024. We use AdamW [28] as the optimizer and set the learning rate to 1e-5 for backbone and text encoder, and 1e-4 for all other modules.

4.3 Settings and Metrics

For the object detection task, we evaluate in zero-shot setting, i.e. *T-Rex2* will not be trained on evaluation benchmarks. We report the AP metric on COCO [22], LVIS [8], ODinW [15] and Roboflow100 [2]. The COCO dataset encompasses 80 common categories. In contrast, the LVIS dataset is characterized by a long-tailed category distribution with 1203 categories. These categories are further segmented into three distinct groups: frequent, common, and rare, with a ratio of `405:461:337` for the val split, and `389:345:70` for the minival split [11]. The ODinW and Roboflow100 datasets contain 35 and 100 datasets collected from Roboflow, respectively, covering a variety of scenarios including aerial, video games, underwater, documents, real world, *etc*., with long-tailed categories.

We compare three evaluation protocols for *T-Rex2* under different workflows.

Text: In this protocol, we use all the category names of an evaluation benchmark as text prompt inputs, consistent with the previous open-vocabulary object detection setting.

Visual-G (Generic): In this protocol, *T-Rex2* works on the generic visual prompt workflow. We extract visual prompt embeddings from the training set images of each benchmark for each category. Taking COCO as an example, we first randomly sample N images for each category that has at least one instance of that category. Next, we extract N visual embeddings for each category using the GT box of each image as input for visual prompting. Subsequently, we compute the average of these N embeddings for each category. These averaged visual embeddings (a total of 80 embeddings) will be used for evaluation. By default, N is set to 16. For each test image, we will repeat this process.

Visual-I (Interactive): In this protocol, *T-Rex2* works on the interactive visual prompt workflow. Given a test image, suppose it has M categories, then for each category, we randomly select one GT box (or convert it to its center point) in the current image as the visual prompt input for this category. This protocol is relatively easier than Visual-G as we know the category of the test set images in advance, as well as being provided with a GT box. However, despite its simplicity, interactive object detection boasts a wide range of application scenarios, including automatic annotation, object counting, *etc*.

---
https://universe.roboflow.com/

Method	Prompt Type	Backbone	COCO-Val Zero-Shot	LVIS Zero-Shot								ODinW Zero-Shot		Roboflow100 Zero-Shot
			val-80	minival-804				val-1203				35val		100val
			AP	AP	AP_f	AP_c	AP_r	AP	AP_f	AP_c	AP_r	AP_{avg}	AP_{med}	AP_{avg}
GLIP-T [20]	Text	Swin-T	46.7	26.0	31.0	21.4	20.8	17.2	25.5	12.5	10.1	19.6	5.1	-
GLIP-L [20]	Text	Swin-L	49.8	37.3	41.5	34.3	28.2	26.9	35.4	23.3	17.1	23.4	11.0	8.6
Grounding DINO [25]	Text	Swin-T	48.4	27.4	32.7	23.3	18.1	-	-	-	-	22.3	11.9	-
Grounding DINO [25]	Text	Swin-L	**52.5**	33.9	38.8	30.7	22.2	-	-	-	-	26.1	18.4	-
DetCLIPv2 [47]	Text	Swin-T	-	40.4	40.0	41.7	36.0	-	-	-	-	-	-	-
DetCLIPv2 [47]	Text	Swin-L	-	44.7	43.7	46.3	43.1	-	-	-	-	-	-	-
DINOv [16]	Visual-G	Swin-T	-	-	-	-	-	-	-	-	-	14.9	5.4	-
DINOv [16]	Visual-G	Swin-L	-	-	-	-	-	-	-	-	-	15.7	4.8	-
T-Rex2	Text	Swin-T	45.8	42.8	46.5	39.7	37.4	34.8	41.2	31.5	29.0	18.0	4.7	8.2
T-Rex2	Visual-G	Swin-T	38.8	37.4	41.8	33.9	29.9	<u>34.9</u>	41.1	30.3	<u>32.4</u>	<u>23.6</u>	<u>17.5</u>	<u>17.4</u>
T-Rex2	Text	Swin-L	<u>52.2</u>	**54.9**	**56.1**	**54.8**	**49.2**	**45.8**	**50.2**	**43.2**	42.7	22.0	7.3	10.5
T-Rex2	Visual-G	Swin-L	46.5	47.6	49.5	46.0	45.4	45.3	49.5	42.0	**43.8**	**27.8**	**20.5**	**18.5**

Table 1: One suite of weights for zero-shot object detection. Red denotes regions where text prompts excel over visual prompts, while green signifies regions favoring visual prompts.

4.4 Zero-Shot Generic Object Detection

In this study, we explore the zero-shot object detection capabilities of *T-Rex2* across four distinct benchmarks: COCO, LVIS, ODinW, and Roboflow100. The term *zero-shot* refers to the methodological approach where the evaluation benchmarks were not exposed to the model during its training phase, possibly encompassing novel categories and image distributions. As shown in Tab. 1, we observe that text prompt and visual prompt can cover different scenarios respectively. Text prompt demonstrates superior performance in scenarios with relatively common categories. For instance, under the generic visual prompt and Swin-T backbone setting, text prompts surpass visual prompts by a margin of 7 AP points on COCO (80 categories). Similarly, in LVIS-minival (804 categories), text prompts achieve a 5.4 AP point advantage over visual prompts. Conversely, in scenarios characterized by long-tailed distributions, visual prompts exhibit a more robust performance compared to text prompts. Specifically, on LVIS-val (1203 categories), visual prompt leads by 3.4 AP points in the rare group, and by 5.6 AP points on ODinW as well as 9.2 AP points on Roboflow100, underscoring its efficacy in handling less common objects. This experiment result indicates that text prompts are suited for common concepts, while visual prompts are more effective for rare and novel categories.

In Fig. 4, we further show the per-category AP difference between visual prompt and text prompt on the LVIS benchmark. We rank the categories of the LVIS dataset in descending order of their frequency of occurrence in the training set. Our analysis shows that text prompts perform better in recognizing common categories with higher occurrence frequencies. In contrast, visual prompts excel in identifying rarer categories as the frequency decreases.

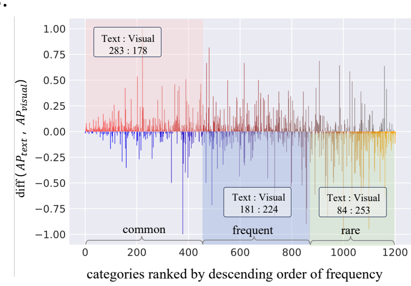


Fig. 4: Performance difference between text and visual prompt on LVIS-val. `Text:Visual` refers to the number of times each has won in the current interval.

Method	Prompt Type	Backbone	COCO-Val Zero-Shot	LVIS Zero-Shot								ODinW Zero-Shot		Roboflow100 Zero-Shot
			val-80	minival-804				val-1203				35val		100val
			AP	AP	AP_f	AP_c	AP_r	AP	AP_f	AP_c	AP_r	AP_{avg}	AP_{med}	AP_{avg}
T-Rex2	Visual-I (Box)	Swin-T	56.6	59.3	54.6	63.5	64.4	62.6	57.3	63.7	71.9	37.7	39.3	30.6
T-Rex2	Visual-I (Box)	Swin-L	58.5	62.5	57.9	66.1	70.1	65.8	61.2	67.3	72.6	39.7	38.1	30.2
T-Rex2	Visual-I (Point)	Swin-T	54.3	57.4	52.1	62.3	63.2	60.0	54.5	60.9	68.8	34.8	34.9	27.7
T-Rex2	Visual-I (Point)	Swin-L	56.8	60.6	56.4	64.2	65.3	63.8	59.1	65.1	71.1	37.5	35.7	27.8

Table 2: One suite of weights for interactive object detection.

4.5 Zero-Shot Interactive Object Detection

T-Rex2 also showcases strong interactive object detection capabilities. As shown in Tab. 2, the interactive visual prompt significantly outperforms both text prompt and generic visual prompt strategies. However, this comparison may not be entirely equitable, as under the Visual-I setting we have the prior about the categories present in the test image. To provide more insight, we evaluate *T-Rex2* on the few-shot object counting task. In this task [10,23,31,43], each test image will be provided with three visual exemplar boxes of the target object and requires to output the number of the target object. We evaluate on FSC147 [35] and FSCD-LVIS [31] datasets. Both datasets comprise scenes with densely populated small objects. Specifically, FSC147 typically features single-target scenes, where generally only one type of object is present per image, whereas FSCD-LVIS mainly includes multi-target images. We report the Mean Average Error (MAE) metric for FSC147 and the AP metric for FSCD-LVIS. Following previous work [10], we use the visual exemplar boxes as the interactive visual prompt. As shown in Tab. 3, *T-Rex2* achieves competitive results compared with the previous SOTA algorithm T-Rex. While not matching T-Rex in terms of MAE, *T-Rex2* performs better than T-Rex in terms of AP, which measures the overall detection accuracy. This result suggests that *T-Rex2*'s interactive capabilities are highly capable in dense and small object scenarios.

Method	FSC147 test	FSCD-LVIS test
	MAE↓	AP
FamNet [35]	22.08	-
Counting-DETR [31]		22.66
BMNet+ [43]	14.62	-
CountTR [23]	11.95	-
T-Rex [10]	**8.72**	40.32
T-Rex2	10.94	**43.35**

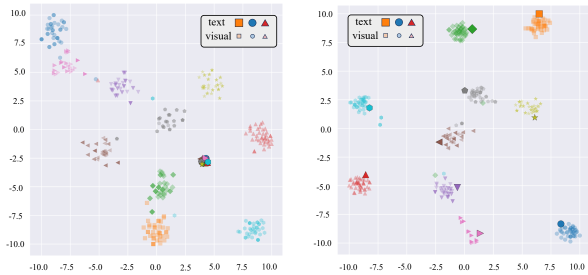
Table 3: Few-shot object counting results on FSC147 [35] and FSCD-LVIS [31] datasets.

Training Strategy	Prompt Type	COCO-Val Zero-Shot	LVIS-Val Zero-Shot			
		AP	AP	AP_r	AP_c	AP_f
Text Prompt Only	Text	46.4	32.8	32.1	32.0	34.0
Visual Prompt Only	Visual-G	14.0	15.3	8.6	11.3	22.8
W/O Contrastive Alignment	Text	44.4	32.2	28.2	28.9	37.6
	Visual-G	38.7	30.2	29.4	26.9	38.7
W/ Contrastive Alignment	Text	45.8(+1.4)	34.8(+2.6)	29.0(+0.8)	31.5(+2.6)	41.2(+3.6)
	Visual-G	38.8(+0.1)	34.9(+4.7)	32.4(+3.0)	30.3(+3.4)	41.1(+2.4)

Table 4: Ablation on the proposed text-visual prompt synergy.

4.6 Ablation Experiments

Ablation of naive joint training. As demonstrated in Tab. 4 (first two rows), the general detection capability of the visual prompt is notably poor (14.0 AP on COCO and 15.3 AP on LVIS-val) when the two prompt modalities are trained separately. The core of the issue lies in the diversity and variance of visual data. For example, when the model is trying to understand what makes a `chair` when every example the model sees is drastically different from the last. Without a consistent context, it is challenging for the model to form a general concept solely with visual prompts. Upon joint training (second two rows in Tab. 4), the efficacy of visual prompts significantly improves. This improvement suggests that the combination of textual context with visual data helps the model form more stable and generalizable representations. However, the naive joint training without explicit alignment between the two prompts somewhat reduce the effectiveness of text prompts, as both AP on COCO and LVIS dropped.



(a) w/o contrastive align **(b)** w/ contrastive align

Fig. 5: t-SNE [29] visualization of text prompt and visual prompt embeddings. We pick the first 10 categories in COCO training set and randomly sample 30 images for each category to get visual prompts.

The observed decline in text prompt capability could be due to the added complexity of multitask learning. We use t-SNE [29] to visualize the distribution of text prompt and visual prompt embeddings in Fig. 5a. We find that the corresponding text prompt and visual prompt embeddings are separated in the feature space, instead of gathered. Therefore the region feature cannot be simultaneously aligned to both the text prompt and the visual prompt, thus making the learning process more challenging.

Ablations of contrastive alignment. As presented in Tab. 4 (last two rows), employing contrastive alignment can lead to improved performance for both text and visual prompts. With contrastive alignment, the distribution between text prompt and visual prompt is more structured as shown in Fig. 5b: text prompts act as anchors and visual prompts cluster around them. This distribution means that visual prompts can learn or derive general knowledge from the closely associated text prompt, making the learning process more efficient. Furthermore, the text prompts are more separated in the feature space compared to Fig. 5a, this indicates that it allows for refinement of text prompts by exposing them to a vast array of visual prompts, thus making them more unique.

# Prompts	Prompt Type	COCO-Val Zero-Shot	LVIS-Val Zero-Shot			
		AP	AP	AP_r	AP_c	AP_f
1	Visual-G	29.2	26.2	27.6	21.3	30.9
4	Visual-G	32.9	32.9	32.0	28.2	38.7
16	Visual-G	38.8	34.9	32.4	30.3	41.1
32	Visual-G	41.3	35.1	32.2	30.3	41.7
64	Visual-G	41.4	35.2	32.4	30.4	41.8

Table 5: Ablation on the number of visual prompts and their generic object detection capabilities.

prompt combination	COCO-Val Zero-Shot	LVIS-Val Zero-Shot			
	AP	AP	AP_r	AP_c	AP_f
Text	**45.8**	34.8	29.0	31.5	41.2
Visual-G	38.8	34.9	32.4	30.3	41.1
Text + Visual-G	42.5	**37.0**	**34.3**	**33.8**	**41.7**

Table 6: Zero-shot object detection results on mixed prompt mode.

Model	Prompt Type	Training Data	Data Size	COCO-Val Zero-Shot	LVIS-Minival Zero-Shot			
				AP	AP	AP-R	AP-C	AP-F
Grounding DINO-T	Text	O365, GoldG	1.4M	48.1	25.6	14.14	19.6	32.2
Grounding DINO-T	Text	O365, GoldG, Cap4M	5.4M	48.4	27.4	18.1	23.3	32.7
T-Rex2-T	Text	O365, GoldG	1.4M	46.1	34.9	32.7	32.9	37.1
T-Rex2-T	Text	O365, GoldG, Bamboo	2.5M	45.7	38.7	35.3	39.4	38.8
T-Rex2-T	Text	O365, GoldG, OpenImages, Bamboo, CC3M, LAION	6.5M	46.4	39.3	35.4	40.5	39.0
T-Rex2-T	Visual-G	O365, OpenImages, HierText, CrowdHuman	2.4M	41.1	38.1	25.8	34.4	43.7
T-Rex2-T	Visual-G	O365, OpenImages, HierText, CrowdHuman, SA-1B	3.1M	38.8	37.4	29.9	33.9	41.8
T-Rex2-T	Visual-I (Box)	O365, OpenImages, HierText, CrowdHuman	2.4M	41.1	40.6	40.3	43.5	38.1
T-Rex2-T	Visual-I (Box)	O365, OpenImages, HierText, CrowdHuman, SA-1B	3.1M	56.6	59.3	64.4	63.5	54.6

Table 7: Ablation of the proposed data engines.

Ablation of generic visual prompt. In Tab. 5, we show that by using more visual prompts, the generic detection capability can be gradually increased. The reason is that visual prompts are not as versatile as text prompts, so we need a large number of visual examples to characterize a generic concept.

Ablation of mixed prompt. We further show the results of mixed prompts for generic object detection. This hybrid method aims to leverage the strengths of both modalities to improve detection performance. In Tab. 6, the mixed prompt on COCO achieves a result that is balanced between text prompt and visual prompt, while on LVIS there is a further performance improvement. We believe that this hybrid inference workflow is more suitable for the case of long-tailed distributions, where text prompt and visual prompt can promote each other.

Ablation of data engines. In Tab. 7, we ablate the effectiveness of the two data engines. For text prompts, introducing the Bamboo dataset improves the performance on the LVIS dataset (+3.8AP), owing to its diverse categories, but slightly declined performance on the COCO dataset (-0.4AP), indicating that the model is less fitted to COCO categories. Adding image caption data further improves the performance on both benchmarks. For visual prompts, the introduction of the SA-1B data significantly improves the interactive capability of the model, but slightly weakens its generic capability. We speculate that the observed performance degradation may stem from the inadequacy of simply employing TAP [33] for object classification within SA-1B, which results in incorrect semantic learning by the model on the SA-1B data. Future work will entail further optimization of this data engine.

5 Conclusion

T-Rex2 is a promising attempt towards generic object detection. We reveal the complementary advantages between text prompts and visual prompts, and successfully align the two prompt modalities into a single model, making it both generic and interactive for open-set object detection. We show that these two prompt modalities can benefit from each other and gain performance through contrastive learning. By switching between different prompt modalities in different scenarios, *T-Rex2* demonstrates impressive zero-shot object detection capabilities and can be used in a variety of applications. We hope that this work will bring new insights into the field of open-set object detection and contribute to further development.

Acknowledgments

We would like to thank everyone involved in the T-Rex2 project, including project lead Lei Zhang, application lead Wei Liu, product manager Qin Liu and Xiaohui Wang, front-end developers Yuanhao Zhu, Ce Feng, and Jiongrong Fan, back-end developers Weiqiang Hu and Zhiqiang Li, UX designer Xinyi Ruan, and tester Yinuo Chen.

References

1. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision. pp. 213–229. Springer (2020)
2. Ciaglia, F., Zuppichini, F.S., Guerrie, P., McQuade, M., Solawetz, J.: Roboflow 100: A rich, multi-domain object detection benchmark. arXiv preprint arXiv:2211.13523 (2022)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
4. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
5. Ghiasi, G., Gu, X., Cui, Y., Lin, T.Y.: Scaling open-vocabulary image segmentation with image-level labels. In: European Conference on Computer Vision. pp. 540–557. Springer (2022)
6. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 1440–1448 (2015)
7. Gu, X., Lin, T.Y., Kuo, W., Cui, Y.: Open-vocabulary object detection via vision and language knowledge distillation. arXiv preprint arXiv:2104.13921 (2021)
8. Gupta, A., Dollar, P., Girshick, R.: Lvis: A dataset for large vocabulary instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5356–5364 (2019)
9. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: 2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06). vol. 2, pp. 1735–1742. IEEE (2006)
10. Jiang, Q., Li, F., Ren, T., Liu, S., Zeng, Z., Yu, K., Zhang, L.: T-rex: Counting by visual prompting. arXiv preprint arXiv:2311.13596 (2023)
11. Kamath, A., Singh, M., LeCun, Y., Synnaeve, G., Misra, I., Carion, N.: Mdetr-modulated detection for end-to-end multi-modal understanding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1780–1790 (2021)
12. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R.: Segment anything (2023)
13. Krasin, I., Duerig, T., Alldrin, N., Ferrari, V., Abu-El-Haija, S., Kuznetsova, A., Rom, H., Uijlings, J., Popov, S., Veit, A., et al.: Openimages: A public dataset for large-scale multi-label and multi-class image classification. Dataset available from https://github. com/openimages **2**(3), 18 (2017)

14. Lee, C., Park, S., Song, H., Ryu, J., Kim, S., Kim, H., Pereira, S., Yoo, D.: Interactive multi-class tiny-object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14136–14145 (2022)
15. Li, C., Liu, H., Li, L.H., Zhang, P., Aneja, J., Yang, J., Jin, P., Lee, Y.J., Hu, H., Liu, Z., et al.: Elevater: A benchmark and toolkit for evaluating language-augmented visual models. arXiv preprint arXiv:2204.08790 (2022)
16. Li, F., Jiang, Q., Zhang, H., Ren, T., Liu, S., Zou, X., Xu, H., Li, H., Li, C., Yang, J., et al.: Visual in-context prompting. arXiv preprint arXiv:2311.13601 (2023)
17. Li, F., Zhang, H., Liu, S., Guo, J., Ni, L.M., Zhang, L.: Dn-detr: Accelerate detr training by introducing query denoising. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13619–13627 (2022)
18. Li, F., Zhang, H., Sun, P., Zou, X., Liu, S., Yang, J., Li, C., Zhang, L., Gao, J.: Semantic-sam: Segment and recognize anything at any granularity. arXiv preprint arXiv:2307.04767 (2023)
19. Li, H., Zhang, H., Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, L.: Taptr: Tracking any point with transformers as detection (2024), https://arxiv.org/abs/2403.13042
20. Li, L.H., Zhang, P., Zhang, H., Yang, J., Li, C., Zhong, Y., Wang, L., Yuan, L., Zhang, L., Hwang, J.N., et al.: Grounded language-image pre-training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10965–10975 (2022)
21. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
22. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. pp. 740–755. Springer (2014)
23. Liu, C., Zhong, Y., Zisserman, A., Xie, W.: Countr: Transformer-based generalised visual counting. arXiv preprint arXiv:2208.13721 (2022)
24. Liu, S., Li, F., Zhang, H., Yang, X., Qi, X., Su, H., Zhu, J., Zhang, L.: DAB-DETR: Dynamic anchor boxes are better queries for DETR. In: International Conference on Learning Representations (2022)
25. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., et al.: Grounding dino: Marrying dino with grounded pre-training for open-set object detection. arXiv preprint arXiv:2303.05499 (2023)
26. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10012–10022 (2021)
27. Long, S., Qin, S., Panteleev, D., Bissacco, A., Fujii, Y., Raptis, M.: Towards end-to-end unified scene text detection and layout analysis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1049–1059 (2022)
28. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
29. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)
30. Minderer, M., Gritsenko, A., Stone, A., Neumann, M., Weissenborn, D., Dosovitskiy, A., Mahendran, A., Arnab, A., Dehghani, M., Shen, Z., et al.: Simple open-vocabulary object detection. In: European Conference on Computer Vision. pp. 728–755. Springer (2022)

31. Nguyen, T., Pham, C., Nguyen, K., Hoai, M.: Few-shot object counting and detection. In: European Conference on Computer Vision. pp. 348–365. Springer (2022)
32. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
33. Pan, T., Tang, L., Wang, X., Shan, S.: Tokenize anything via prompting. arXiv preprint arXiv:2312.09128 (2023)
34. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
35. Ranjan, V., Sharma, U., Nguyen, T., Hoai, M.: Learning to count everything. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3394–3403 (2021)
36. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
37. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28**, 91–99
38. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized intersection over union: A metric and a loss for bounding box regression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 658–666 (2019)
39. Schuhmann, C., Vencu, R., Beaumont, R., Kaczmarczyk, R., Mullis, C., Katta, A., Coombes, T., Jitsev, J., Komatsuzaki, A.: Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. arXiv preprint arXiv:2111.02114 (2021)
40. Shao, S., Li, Z., Zhang, T., Peng, C., Yu, G., Zhang, X., Li, J., Sun, J.: Objects365: A large-scale, high-quality dataset for object detection. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 8430–8439 (2019)
41. Shao, S., Zhao, Z., Li, B., Xiao, T., Yu, G., Zhang, X., Sun, J.: Crowdhuman: A benchmark for detecting human in a crowd. arXiv preprint arXiv:1805.00123 (2018)
42. Sharma, P., Ding, N., Goodman, S., Soricut, R.: Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 2556–2565 (2018)
43. Shi, M., Lu, H., Feng, C., Liu, C., Cao, Z.: Represent, compare, and learn: A similarity-aware framework for class-agnostic counting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9529–9538 (2022)
44. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: Fully convolutional one-stage object detection. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9627–9636 (2019)
45. Xu, Y., Zhang, M., Fu, C., Chen, P., Yang, X., Li, K., Xu, C.: Multi-modal queried object detection in the wild. arXiv preprint arXiv:2305.18980 (2023)
46. Yao, A., Gall, J., Leistner, C., Van Gool, L.: Interactive object detection. In: 2012 IEEE conference on computer vision and pattern recognition. pp. 3242–3249. IEEE (2012)
47. Yao, L., Han, J., Liang, X., Xu, D., Zhang, W., Li, Z., Xu, H.: Detclipv2: Scalable open-vocabulary object detection pre-training via word-region alignment. In:

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23497–23506 (2023)
48. Yao, L., Han, J., Wen, Y., Liang, X., Xu, D., Zhang, W., Li, Z., Xu, C., Xu, H.: Detclip: Dictionary-enriched visual-concept paralleled pre-training for open-world detection. arXiv preprint arXiv:2209.09407 (2022)
49. Zang, Y., Li, W., Zhou, K., Huang, C., Loy, C.C.: Open-vocabulary detr with conditional matching. In: European Conference on Computer Vision. pp. 106–122. Springer (2022)
50. Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L.M., Shum, H.Y.: Dino: Detr with improved denoising anchor boxes for end-to-end object detection (2022)
51. Zhang, H., Li, F., Zou, X., Liu, S., Li, C., Yang, J., Zhang, L.: A simple framework for open-vocabulary segmentation and detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1020–1031 (2023)
52. Zhang, Y., Sun, Q., Zhou, Y., He, Z., Yin, Z., Wang, K., Sheng, L., Qiao, Y., Shao, J., Liu, Z.: Bamboo: Building mega-scale vision dataset continually with human-machine synergy. arXiv preprint arXiv:2203.07845 (2022)
53. Zhong, Y., Yang, J., Zhang, P., Li, C., Codella, N., Li, L.H., Zhou, L., Dai, X., Yuan, L., Li, Y., et al.: Regionclip: Region-based language-image pretraining. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16793–16803 (2022)
54. Zhou, X., Girdhar, R., Joulin, A., Krähenbühl, P., Misra, I.: Detecting twenty-thousand classes using image-level supervision. In: European Conference on Computer Vision. pp. 350–368. Springer (2022)
55. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. arXiv preprint arXiv:1904.07850 (2019)
56. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)
57. Zou, X., Yang, J., Zhang, H., Li, F., Li, L., Gao, J., Lee, Y.J.: Segment everything everywhere all at once. arXiv preprint arXiv:2304.06718 (2023)