WeDetect: Fast Open-Vocabulary Object Detection as Retrieval

Shenghao Fu[1†], Yukun Su[1†], Fengyun Rao[1*], Jing LYU[1], Xiaohua Xie[2*], Wei-Shi Zheng[2,3]
[1]WeChat Vision, Tencent Inc. [2]Pazhou Laboratory (Huangpu), China [3]Shenzhen Loop Area Institute
1185514120@qq.com, {yukunsu,fengyunrao}@tencent.com, wszheng@ieee.org

Abstract

Open-vocabulary object detection aims to detect arbitrary classes via text prompts. Methods without cross-modal fusion layers (non-fusion) offer faster inference by treating recognition as a retrieval problem, i.e.*, matching regions to text queries in a shared embedding space. In this work, we fully explore this retrieval philosophy and demonstrate its unique advantages in efficiency and versatility through a model family named WeDetect: (1)* ***State-of-the-art performance. WeDetect*** *is a real-time detector with a dual-tower architecture. We show that, with well-curated data and full training, the non-fusion WeDetect surpasses other fusion models and establishes a strong open-vocabulary foundation. (2)* ***Fast backtrack of historical data. WeDetect-Uni*** *is a universal proposal generator based on WeDetect. We freeze the entire detector and only finetune an objectness prompt to retrieve generic object proposals across categories. Importantly, the proposal embeddings are class-specific and enable a new application,* ***object retrieval****, supporting retrieval objects in historical data. (3)* ***Integration with LMMs for referring expression comprehension (REC).*** *We further propose* ***WeDetect-Ref****, an LMM-based object classifier to handle complex referring expressions, which retrieves target objects from the proposal list extracted by WeDetect-Uni. It discards next-token prediction and classifies objects in a single forward pass. Together, the WeDetect family unifies detection, proposal generation, object retrieval, and REC under a coherent retrieval framework, achieving state-of-the-art performance across 15 benchmarks with high inference efficiency. Code is available at* https://github.com/WeChatCV/WeDetect.

1. Introduction

Recognition is a central problem in computer vision. At the image level, the field has progressed from closed-set image classification [10, 20, 25] to open-vocabulary image retrieval powered by large-scale image–text contrastive learning [48]. In parallel, open-vocabulary object detection [12, 15, 16, 35, 39, 41] extends beyond the fixed label spaces of closed-set detectors [3, 13, 14, 50], allowing recognition and localization of arbitrary categories specified by textual prompts. By aligning region features with text embeddings, open-vocabulary detectors can achieve zero-shot region recognition without task-specific training.

To improve vision–language alignment, recent open-vocabulary object detectors [16, 35, 41] employ various deep cross-modal fusion mechanisms. While achieving high accuracy, their computationally intensive fusion layers substantially degrade inference efficiency. Moreover, the fusion makes visual features query-specific, preventing feature sharing across different textual prompts. For example, evaluating Grounding-DINO [41] on LVIS [19] containing 1,203 categories with a chunk size of 40 requires 31 separate forward passes, resulting in several seconds of latency per image and limiting practical deployment.

In contrast, non-fusion methods adopt a dual-tower architecture and enjoy a fast inference speed. We notice a key characteristic of the non-fusion paradigm, i.e., its recognition is similar to the retrieval problem, which matches image regions against text queries in a shared embedding space. This formulation enjoys unique advantages in efficiency and versatility. Motivated by this insight, we fully explore the retrieval-inspired philosophy through a model family named WeDetect: (1) **WeDetect**, a strong detection foundation with real-time latency and superior open-vocabulary object detection performance; (2) **WeDetect-Uni**, a universal proposal generator supporting fast backtrack of historical data; and (3) **WeDetect-Ref**, an LLM-based REC model for complex expression detection. They are demonstrated as follows:

WeDetect is finetuned from a pretrained CLIP model, comprising a text encoder for encoding class names and a visual encoder for extracting multi-scale visual features. Classification is performed via dot products between class embeddings and image grid features. Rather than relying on deep fusion, we employ three key techniques to achieve su-

*: Corresponding authors are Xiaohua Xie and Fengyun Rao.
†: Equal Contribution.
Work was done when Shenghao Fu was an intern at Tencent.

perior open-vocabulary object detection performance with efficient inference: (1) Model pretraining: WeDetect is fine-tuned from a strong, well-pretrained CLIP model to inherit robust open-vocabulary capabilities; (2) Model architecture: As mainstream CLIP variants typically adopt a ViT encoder [10], whose plain design is suboptimal for detection, we pretrain a CLIP variant with a ConvNeXt [42] backbone that naturally provides multi-scale features; and (3) Training data: We develop a data engine to curate a high-quality dataset characterized by balanced concepts, exhaustive annotations, and multi-granularity labels, comprising 15M images and 330M bounding boxes. Together, these design choices enable WeDetect to deliver strong open-vocabulary object detection performance and fast inference without resorting to deep fusion.

Building on WeDetect, we introduce a universal proposal generator, **WeDetect-Uni**. We freeze the entire detector and train only an objectness embedding for classification. As the detector is frozen, the box embeddings corresponding to the top-scoring proposals remain class-specific and can therefore be used for class-specific classification. By caching proposals together with their box embeddings, we enable fast object retrieval via a CLIP-style dot product. On this basis, we propose a new application, **object retrieval**, which aims to retrieve images containing user-specified objects even when they are small (*e.g.* cigarette butts). This fine-grained, local retrieval task complements CLIP's conventional image-level retrieval.

To further handle complex expressions in Referring Expression Comprehension (REC), we employ an LMM-based classification model, **WeDetect-Ref**. Given the top-scoring proposals produced by WeDetect-Uni together with the query expression, WeDetect-Ref retrieves the target objects by applying a newly designed binary classification head over the candidate proposals. The model processes all objects in parallel and the prediction is conducted in a single forward pass, eliminating the time-consuming next-token prediction which decodes objects sequentially. This retrieval-based paradigm avoids the bounding box regression drawback derived from language modeling and the slow inference speed derived from next-token prediction, while fully leveraging LLM's language understanding and open-vocabulary capabilities to achieve fast and accurate classification.

Leveraging the retrieval-based methodology, the WeDetect family demonstrates strong open-vocabulary capability with exceptionally high inference throughput. Specifically, WeDetect-Tiny attains 37.4 AP on LVIS minival and 31.4 AP on LVIS at 62.5 fps, surpassing YOLO-World-L [6] by 2.0 and 4.6 AP, respectively, while YOLO-World-L runs at 54.6 fps. By scaling up the model, WeDetect-Large achieves 49.4 AP on LVIS, outperforming LLMDet [16] by 7.4 AP. In the object retrieval task, WeDetect-Uni outperforms CLIP [48] by 37.2 F1 scores, showing its unique advantage in fine-grained perception. Moreover, WeDetect-Ref 4B gets an average score of 93.2 on refcoco/+/g [26], exceeding Qwen3-VL [1] 4B by 6.5 points with a 13× speedup. We hope that this retrieval paradigm can be broadly adopted by the research community.

2. Related Work

2.1. Open-Vocabulary Object Detection

Open-vocabulary object detection aims to detect objects with text prompts, requiring fine-grained vision-language alignment. To construct a unified vision-language space, previous works mainly focus on four aspects: (1) Training data constructions: GLIP [35] first unifies object detection and phrase grounding through region-word contrastive pre-training, which can leverage massive image-text pairs for training. The vast vocabulary existing in the web-scale image-text pairs builds a robust vision-language space. Further, constructing hard negative samples [34, 66, 67, 75] can provide richer supervision and achieve fine-grained alignment. By scaling up data and computation [47, 67], models can achieve impressive zero-shot performance. (2) Training objective: In addition to region-word contrastive learning, unifying other language tasks, including mask language modeling [72], dense captioning [43, 68], and co-training with a large language model [16], enriches visual representations with language knowledge, thus creating a stronger open-vocabulary detector. (3) Vision-language fusion layers: Deep vision-language fusion layers [11, 18, 35, 41, 58] that jointly integrate visual and textual features can further improve vision-language alignment. However, these computationally intensive fusion layers greatly reduce inference efficiency. And the extracted vision features can not be shared across different queries. (4) Model distillation: Other methods [14, 15, 17, 60] aim to distill the open-vocabulary knowledge from other foundation models. In this work, we revisit the deep fusion architecture and propose a family of models with plain architecture following the retrieval methodology.

2.2. Large Vision-Language Model

Large Vision-Language Models (LVLMs) [1, 2, 5, 62], pretrained on massive corpus, show not only professional world knowledge and reasoning ability but also superior visual perception and understanding ability. Therefore, LVLMs excel at open-vocabulary perception. To extend LVLMs with region perception ability [36, 70, 71, 73], each object will be encoded into special tokens separately. However, the language modeling mechanism constrains them for precise object localization, as digits are encoded into separate discrete tokens and are optimized via cross-entropy loss. To eliminate the regression drawback, some meth-

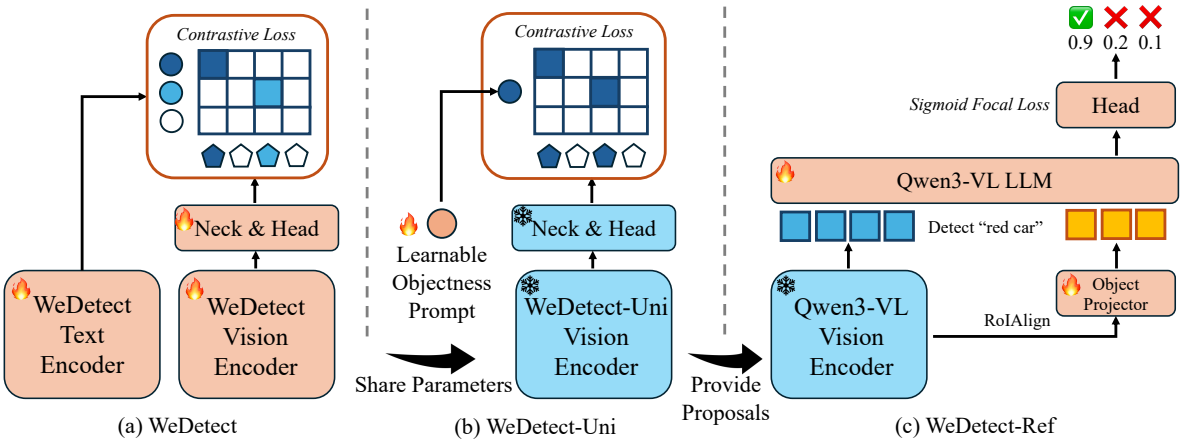

Figure 1. The WeDetect model family: (a) WeDetect is an open-vocabulary object detector with a dual-tower architecture without any multi-modal fusion layers. (b) WeDetect-Uni is a universal proposal generator whose parameters are shared with WeDetect except for a learnable objectness prompt for classification. (c) WeDetect-Ref is an LLM-Based REC model, which can retrieve target objects from proposals provided by WeDetect-Uni corresponding to user-provided expressions.

ods [30, 55, 56] utilize an extra decoder to decode object tokens, while others [22, 24, 40, 74] pre-extract some proposals for LLMs to refer to. However, these methods still follow the next-token prediction mechanism, in which objects are decoded sequentially. Therefore, the inference speed is greatly constrained. In this work, we follow the retrieval methodology and utilize the LLM as a classifier to process objects in parallel.

3. WeDetect: A Strong Detection Foundation

In this work, we aim to develop a simple and fast open-vocabulary object detector with diverse usages following the retrieval methodology. Based on the goal, we discard the time-consuming fusion layers. In contrast, we adopt the dual-tower architecture from CLIP [48] and extend it to region-wise perception. To achieve fine-grained vision-language alignment, we make great efforts in dataset construction and model training, which are detailed as follows.

3.1. Model Architecture

WeDetect is a dual-tower architecture model, as shown in Figure 1(a). The language encoder is initialized from XLM-RoBERTa [7] while the vision encoder follows a YOLO-like architecture containing a ConvNeXt [42] backbone to produce multi-scale features, a CSPRepBiFPAN neck [32], and a YOLO-World [6] contrastive head. The loss functions and label assignment strategy are all the same as YOLO-World, which is a region-text contrastive loss for classification along with a box regression loss. Different from YOLO-World, we do not use any fusion layers within the neck. And classification is simply conducted via the dot product between image grid features and class text embeddings. This simple and elegant architecture ensures a high inference speed.

3.2. Dataset Construction

A high-quality dataset should be rich in diversity and accurate in annotations. Although open-sourced grounding datasets (*e.g.* GoldG [35]) contain diverse text expressions, they are limited in dataset size, image diversity, annotation integrity, and annotation diversity. Therefore, we collect a large-scale grounding dataset with balanced-sampled images and well-annotated labels.

Source image sampling. We first sample source images from various datasets, including SAM-1B [28], LAION [52], CC12M [4], Zero [64], and self-crawled images from licensed websites. The raw captions paired with images (if they exist) are used for selecting some rare nouns to balance the concepts. Totally, source images comprise 15M samples with a wide span of concepts from various domains, ensuring high image diversity.

Box annotation pipeline. We further propose an automatic data engine to annotate images with high-quality and multi-granularity labels, as shown in Figure 2. To ensure high text diversity, we resort to generative methods to generate instance-specific annotations. Specifically, we first train an objectness detector with all available object detection datasets. The objectness detector recalls all objects within the image, which ensures the integrity of annotations. Then, a modern MLLM Qwen2.5-VL 7B [2] is used to generate instance-specific hierarchical labels. For example, as shown in Figure 2, a dog will be annotated as "animal, dog, a yellow dog". In our experiments, these multi-granularity labels greatly enrich the text diversity and improve the performance. To ensure label quality, we fine-tune the Qwen2.5-VL model with a human-annotated in-

Figure 2. The proposed data engine. We first use an objectness detector to detect all regions of interest along with masks produced by SAM. Then, an LMM is used to generate multi-granularity and instance-specific labels for each object. The LMM is finetuned by us to ensure high-quality labeling and structural output.

struction dataset to enhance two crucial abilities: structural output and rejecting recognition. Structural output requires the model to output the label with a fixed template, first outputting coarse-grained labels and then fine-grained labels. And the model with rejecting recognition will not generate labels for erroneous boxes, which also serves as a validation for the previous proposal generation. These boxes will be discarded. Further, to enhance Qwen2.5-VL's local awareness, we highlight the object boundaries in the original image with the mask produced by SAM [28] along with the textual box coordinates as the model inputs. Once the annotator is trained, it can annotate remaining images without human supervision.

In summary, our self-annotated dataset contains 15M samples and 330M bounding boxes. We also include other open-sourced object detection and grounding datasets for training, including OpenImagesV6 [29], Objects365 V2 [53], V3Det [59], ImageNetBox [9], and GoldG [35]. Details are shown in the Appendix.

3.3. Model Training

Staged-wise training method. To equip the model with the basic open-vocabulary ability, we first pretrain the model with a CLIP-like image-level contrastive objectiveness on a large-scale image-text dataset. The resulting checkpoints are used for initializing the visual backbone and the language encoder of WeDetect. As the neck and the head are still randomly initialized, in the second stage, we freeze the visual backbone and the language encoder, and only train the remaining components. In the last stage, all parameters are trained in an end-to-end manner. This staged-wise training method can fully leverage the pretraining knowledge while adopting it from image-level to region-level.

Multi-granularity label sampling. In our self-collected dataset, each object is annotated with multi-granularity labels. We propose a multi-granularity label sampling method as a kind of data augmentation, which independently samples a label from the candidate list for each object during each training iteration. The fine-grained and diverse text labels will not only provide rich supervision for the single object but also construct a diverse and training-time-specific vocabulary for the whole batch, providing diverse negative samples for learning. This data augmentation greatly boosts the open-vocabulary performance.

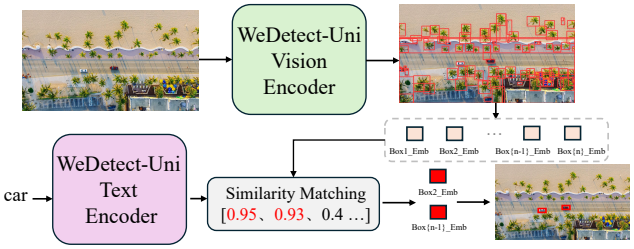

Figure 3. Illustration of applying WeDetect-Uni to the object retrieval task. We first use WeDetect-Uni to extract some regions of interest. And the box embeddings corresponding to the top-scoring proposals are cached to represent the image. Once a query comes, only a simple dot production is needed for fast retrieval.

4. WeDetect-Uni: A Universal Proposal Generator

Extracting arbitrary objects via a universal objectness prompt. In this section, we extend WeDetect to a universal proposal generator, WeDetect-Uni, without user-provided text prompts. Specifically, as shown in Figure 1(b), we freeze the entire detector and train a universal objectness prompt for classification, which is kind of linear probing finetuning. Based on WeDetect's discriminative features, only a single learnable prompt is needed for high recall rates. Importantly, different from other class-agnostic proposal networks [50], the box embeddings corresponding to the top-scoring proposals are still class-specific, which can be used for classification and serve as the basis for the following new application.

A new local object retrieval task. In this work, we propose a new task, object retrieval, in which models should retrieve all images containing a user-specified object category from a database. Different from the image-text retrieval task, where the query focuses on the global image semantics, the object retrieval task pays attention to the local semantics, such as small objects like "cigarette butts", which comple-

ments the image-level retrieval task. This task is valuable for keyword image retrieval, image content examination and verification, and other real-world applications. For evaluation, the new task can be conducted on common object detection datasets, where class names are user queries, the whole validation set is the database, and images containing class-specific annotations constitute the ground truth image set. The evaluation is conducted between the ground truth image set and the predicted image set, and the metrics are precision, recall, and F1 score.

Applying WeDetect-Uni to the object retrieval task. Different from CLIP, where each image is encoded as a single embedding, we use a set of object embeddings to represent an image. As shown in Figure 3, we use WeDetect-Uni to detect all regions of interest. The box embeddings corresponding to the top-scoring proposals are pre-extracted and cached to represent the image. Once a new query arrives, a simple dot production is needed for fast retrieval.

5. WeDetect-Ref: An LLM-Based REC Model

5.1. Formulating REC as Retrieval

In real-world applications, user queries can be much more complex by specifying appearances, materials, locations, and even requiring some reasoning ability based on common sense, which needs deep language and semantic understanding and poses great challenges to traditional detectors. Inspired by recent frontier large vision-language models, we aim to use them to handle the complex referring expression comprehension (REC) task. However, two crucial challenges should be tackled: First, as large language models (LLMs) are trained with the language modeling objective rather than the regression objective, in which digits are represented as discrete tokens and optimized by cross-entropy loss, making the model less sensitive to the precise bounding boxes. Second, LLMs work in a next-token prediction manner, in which tokens should be generated sequentially with multiple model forward passes, resulting in extremely long model latency. Motivated by VideoITG [61], we formulate the REC task as a retrieval task and simply use the large language models as a classifier to retrieve target objects from a pre-extracted candidate list.

Specifically, we first use WeDetect-Uni to extract some objects of interest as the object candidate list $\{B_i\}_{i=1}^n$. As shown in Figure 1(c), for each object, we extract its multiscale RoI features from the MLLM's visual encoder and then compress them to a single token $\{o_i\}_{i=1}^n$ via a linear object projector. The full image tokens I, the user query q, and object tokens $\{o_i\}_{i=1}^n$ are concatenated and sent to the LLM for classification. The classification is conducted by applying a newly introduced binary classification head over the hidden embeddings of object tokens to decide whether the object belongs to the query:

$$\{h_i\}_{i=1}^n = \text{LLM}(I, q, \{o_i\}_{i=1}^n), \quad (1)$$

$$\{s_i\}_{i=1}^n = \text{Sigmoid}(\text{Classifier}(\{h_i\}_{i=1}^n)) \in [0, 1], \quad (2)$$

where $\{s_i\}_{i=1}^n$ are classification scores for each object.

In this paradigm, the LLM only acts as a classifier to retrieve target objects corresponding to the query from a class agnostic candidate list, which enjoys two advantages: First, this retrieval-based paradigm avoids the bounding box regression drawback derived from language modeling while fully leveraging LLM's language understanding and open-vocabulary capabilities to achieve fast and accurate classification. Second, this retrieval-based paradigm discards the next-token prediction mechanism so that the prediction can be conducted in a single model forward pass, achieving a superior inference speed. Although the retrieval-based paradigm is briefly explored by previous works [22, 74], they still follow the next-token prediction mechanism thus limiting the inference speed.

5.2. A Three-Stage Training Recipe

In this work, we use Qwen3-VL [1] as our base MLLM and extend it with fine-grained region perception ability by a three-stage training recipe.

Stage 1: Region projector training. In our model, we introduce a new special token "⟨object⟩" as a placeholder for each object. For each object, we use RoIAlign to extract its multi-scale features from the last visual feature map and other deep stack visual feature maps. We use a linear layer as the region projector to compress RoI features into a single token. The bounding boxes will be encoded as position embeddings and added to object tokens. Then, the placeholder is replaced by the object token before sending to the LLM. In this stage, we only finetune the newly introduced region projector with a 700K image-level and region-level caption dataset. The data format is shown as follows:

> user: ⟨image⟩ Describe the object ⟨object⟩ briefly.
> assistant: far right guy.

Stage 2: Region perception finetuning. In this stage, we further finetune the LLM and projector to better align with the object tokens, while the vision encoder is still frozen. In addition to caption data, we include many other image-level and region-level instruction tuning data (around 1.7M data) with the same format as above. After finetuning, the LLM can perceive specific objects accurately.

Stage 3: Region classification finetuning. In the last stage, we finetune the LLM into a classifier model. We discard the original language modeling head, and instead train a binary classification head. The classification head is only applied over the hidden embeddings of object tokens. To process a list of objects at once, we formulate the data template as follows:

Table 1. Zero-shot detection performance. WeDetect achieves state-of-the-art performance across various model scales. Gray numbers indicate including COCO data in training. FPS is tested on COCO dataset.

Method	Backbone	Resolution	#Params	FPS	LVIS^{minival}				LVIS				COCO	COCO-O	ODinW13	ODinW35
					AP	AP_r	AP_c	AP_f	AP	AP_r	AP_c	AP_f	AP	AP	AP	AP
YOLO-World-L [6]	YOLOv8-L	640*640	48M	54.6	35.4	27.6	34.1	38.0	26.8	19.8	23.6	33.4	**44.9**	32.5	38.4	17.1
YOLOE-8-L [57]	YOLOv8-L	640*640	45M	-	35.9	33.2	34.8	37.3	-	-	-	-	-	-	-	-
WeDetect-Tiny	ConvNext-T	640*640	33M	62.5	**37.4**	**33.3**	**36.8**	**38.8**	**31.4**	**24.7**	**29.2**	**36.8**	**44.9**	**38.6**	**46.4**	**21.1**
GLIP [35]	Swin-T	800*1333	232M	5.4	26.0	20.8	21.4	31.0	17.2	10.1	12.5	25.2	46.1	29.0	46.5	19.6
Grounding-DINO [41]	Swin-T	800*1333	172M	6.0	27.4	18.1	23.3	32.7	20.1	10.1	15.3	29.9	48.4	37.6	51.4	22.7
DetCLIP [66]	Swin-T	800*1333	-	-	35.9	33.2	35.7	36.4	28.4	25.0	27.0	28.4	-	-	43.3	-
DetCLIPv2 [67]	Swin-T	800*1333	-	-	40.4	36.0	41.7	40.4	32.8	31.0	31.7	34.8	-	-	-	-
DetCLIPv3 [68]	Swin-T	800*1333	-	-	47.0	**45.1**	**47.7**	46.7	38.9	37.2	37.5	41.2	47.2	38.5	-	-
T-Rex2 [21]	Swin-T	800*1333	-	-	42.8	37.4	39.7	46.5	34.8	29.0	31.5	41.2	45.8	-	-	18.0
OV-DINO [58]	Swin-T	800*1333	-	-	40.1	34.5	39.5	41.5	32.9	29.1	30.4	37.4	50.2	-	-	-
MM-GDINO [76]	Swin-T	800*1333	172M	6.0	41.4	34.2	37.4	46.2	31.9	23.6	27.6	40.5	50.4	34.0	52.5	23.1
LLMDet [16]	Swin-T	800*1333	172M	6.0	44.7	37.3	39.5	**50.7**	34.9	26.0	30.1	44.3	55.6	36.1	52.1	23.8
DINO-X Edge [51]	EfficientViT-L2	640*640	-	19.8	44.5	41.4	47.3	42.6	38.4	**38.9**	38.3	38.2	48.7	-	-	-
WeDetect-Base	ConvNext-B	640*640	176M	35.1	**47.3**	43.5	45.9	49.3	**41.4**	35.2	**39.5**	**46.2**	**52.1**	**44.1**	**53.1**	**24.6**
GLIP [35]	Swin-L	800*1333	430M	3.1	37.3	28.2	34.3	41.5	26.9	17.1	23.3	36.4	49.8	-	-	-
Grounding-DINO [41]	Swin-L	800*1333	343M	2.1	33.9	22.2	30.7	38.8	–	–	–	–	52.5	-	-	**26.1**
DetCLIP [66]	Swin-L	800*1333	-	-	38.6	36.0	38.3	39.3	28.4	25.0	27.0	31.6	-	-	50.0	24.9
DetCLIPv2 [67]	Swin-L	800*1333	-	-	44.7	43.1	46.3	43.7	36.6	33.3	36.2	38.5	-	-	-	-
DetCLIPv3 [68]	Swin-L	800*1333	-	-	48.8	49.9	49.7	47.8	41.4	41.4	40.5	42.3	48.5	48.8	-	-
LLMDet [16]	Swin-L	1000*1560	343M	2.1	50.6	41.7	46.2	**56.1**	42.0	31.6	38.8	50.2	59.2	**53.2**	53.3	24.6
T-Rex2 [21]	Swin-L	800*1333	-	-	54.9	49.2	**54.8**	**56.1**	45.8	42.7	43.2	50.2	52.2	-	50.3	22.0
WeDetect-Large	ConvNext-L	1280*1280	490M	6.0	**55.0**	**51.1**	54.5	**56.1**	**49.4**	**43.3**	**48.2**	**53.5**	**54.5**	47.0	**53.4**	25.8

user: ⟨image⟩ Please detect the "CLASSNAME" in the image.
assistant: ⟨object⟩⟨object⟩⟨object⟩⟨object⟩⟨object⟩

where the "CLASSNAME" will be replaced by the user-provided categories or expressions and the number of ⟨object⟩ equals the number of proposals. We collect some open-sourced object detection datasets and referring expression comprehension datasets containing 4M samples for training. Details are summarized in the Appendix. For the loss function, we use sigmoid focal loss [38] with IoU as soft labels. All proposals with an IoU greater than 0.5 with any ground truth bounding box are treated as positive samples. In this stage, the trainable parameters are also the LLM and the projector. Other implementation details will be provided in the Appendix.

6. Experiment

6.1. Main Result

WeDetect achieves superior open-vocabulary object detection performance with a faster inference speed. To demonstrate the open-vocabulary capacity, we evaluate WeDetect on various object detection benchmarks in a zero-shot manner, including LVIS [19], COCO [37], COCO-O [46], and ODinW [31]. LVIS is a large vocabulary dataset with 1203 classes and a long-tail distribution, requiring recognition of a wide span of objects. COCO contains 80 common object categories in everyday scenes, while COCO-O retains the same categories but extends them to six distinct domains, posing a challenge for cross-domain generalization. ODinW includes 35 diverse object detection datasets with different vocabularies, offering a comprehensive test of zero-shot transferability. As shown in Table 1, WeDetect achieves state-of-the-art performance across different model scales. Specifically, WeDetect-Tiny outperforms YOLO-World-L [6] by 2.0 AP on LVIS minival, 4.6 AP on LVIS, 6.1 AP on COCO-O, 8.0 AP on ODinW13, and 4.0 AP on ODinW35, while running at a faster speed. When scaling up model size, WeDetect-Large outperforms the previous SOTA model T-Rex2 [21] by 3.6 AP on the challenging LVIS benchmark. These results highlight WeDetect's superior open-vocabulary recognition ability. More importantly, without cross-modal fusion layers, WeDetect runs at an extremely fast speed. WeDetect-Tiny runs at a 62.5 fps, surpassing YOLO-World, despite the latter is optimized for efficient inference. Further, WeDetect-Base and WeDetect-Large exceed GroundingDINO [41] by 6 times and 3 times in speed but with higher performance. These results demonstrate the superior advantages of the dual-tower architecture.

WeDetect-Uni gets high recall rates with only a learnable prompt. Based on the strong detection foundation WeDetect, WeDetect-Uni only trains a learnable prompt for universal proposal generation. We evaluate the recall rates on three benchmarks: COCO [37], LVIS [19], and PACO-LVIS [49]. Note that the three benchmarks share the same images but with different annotation granularities. COCO has only 80 classes, while LVIS extends the vocabulary to 1203 classes, and PACO further annotates object parts. The multi-granularity annotations construct an ideal benchmark for universal proposal generation. As shown in Table 3, WeDetect-Large-Uni achieves the highest recall rates on all datasets with a frozen detector, which demonstrates the highly discriminative features of WeDetect.

WeDetect-Uni enjoys unique advantages in the region-wise object retrieval task. In this work, we propose a new application, denoted as object retrieval, which aims to retrieve images with user-specified objects. We use common

Table 2. Evaluation results on common referring expression comprehension datasets. The evaluation metric for RefCOCO, RefCOCO+, and RefCOCOg is the Top-1 accuracy. FPS is tested on the RefCOCO dataset.

Method	FPS	RefCOCO			RefCOCO+			RefCOCOg			HumanRef			
		val	testA	testB	val	testA	testB	val	test	Avg.	P	R	DF1	Rej.
Grounding-DINO-L [41]	3.1	90.6	93.2	88.2	82.8	89.0	75.9	86.1	87.0	86.6	33.1	75.2	23.3	-
Qwen2.5-VL 3B [2]	-	89.1	91.7	84.0	82.4	88.0	74.1	85.2	85.7	85.0	-	-	-	-
Qwen2.5-VL 7B [2]	-	90.0	92.5	85.4	84.2	89.1	76.9	87.2	87.2	86.6	68.5	52.5	56.2	7.1
InternVL2.5-8B [5]	-	90.3	94.5	85.9	85.2	91.5	78.8	86.7	87.6	87.6	37.9	29.8	31.9	54.9
InternVL3.5-8B [62]	-	92.4	94.7	88.7	87.9	92.4	82.4	89.6	89.4	89.7	-	-	-	-
InternVL3.5-38B [62]	-	90.3	91.8	89.0	87.5	90.0	84.7	89.7	89.9	89.1	-	-	-	-
InternVL3.5-241B-A28B [62]	-	94.1	96.3	91.5	**91.6**	94.6	**86.9**	92.0	92.1	92.4	-	-	-	-
Qwen3-VL-235B-A22B Thinking [1]	-	-	-	-	-	-	-	-	-	92.4	-	-	-	-
Octopus 7B [74]	-	89.0	92.6	83.4	83.6	89.4	76.0	84.3	86.3	85.6	-	-	-	-
VLM-R1 3B [54]	-	90.1	92.3	85.2	84.2	89.4	76.8	85.6	86.8	86.3	-	-	-	-
Rex-Omni 3B [23]	-	86.6	89.5	82.8	79.6	84.8	71.4	85.3	86.2	83.3	79.3	80.1	75.6	-
ChatRex 7B [22]	-	91.0	94.1	87.0	89.8	91.9	79.3	89.8	90.0	89.1	72.2	50.4	55.6	0.0
VLM-FO1 3B [40]	-	91.1	93.7	87.6	86.4	91.9	80.6	88.9	88.3	88.6	**87.1**	83.3	**82.6**	-
RexSeek 7B [24]	-	-	-	-	-	-	-	84.0	84.4	-	85.8	85.9	82.4	54.1
Qwen3-VL 2B [1]	0.6	88.2	91.0	83.1	78.6	85.2	70.4	84.7	85.0	83.3	69.7	58.2	60.2	20.6
Qwen3-VL 4B [1]	0.4	90.7	92.2	86.7	82.9	89.4	75.6	87.3	87.7	86.6	76.7	65.9	67.8	39.1
WeDetect-Ref 2B	6.6	94.3	95.6	92.6	88.1	92.6	83.1	92.0	92.2	91.3	84.7	85.1	79.8	61.0
WeDetect-Ref 4B	5.3	**95.6**	**96.7**	**93.6**	90.5	**94.8**	86.8	**93.8**	**93.9**	**93.2**	86.3	**87.1**	81.8	**64.1**

Table 3. Zero-shot recall rates on object detection datasets. Gray numbers indicate including the target data in training.

Method	COCO		LVIS		PACO-LVIS	
	AR_{50}	AR	AR_{50}	AR	AR_{50}	AR
100 proposals						
MAVL [44]	67.3	40.4	40.7	22.3	24.5	12.5
OLN [27]	71.9	47.5	35.8	21.4	26.7	14.8
RPN-R50 [50]	75.7	46.1	39.3	22.9	30.0	15.9
UPN (fine-grained) [22]	89.6	69.2	65.0	49.0	38.3	27.2
UPN (coarse-grained) [22]	90.6	69.7	62.0	46.6	37.8	26.6
WeDetect-Base-Uni	87.9	66.7	67.4	50.8	37.1	25.7
WeDetect-Large-Uni	**89.7**	**69.3**	**70.9**	**56.3**	**38.4**	**27.9**
300 proposals						
MAVL [44]	69.7	41.2	44.1	23.5	27.9	13.4
OLN [27]	79.5	52.4	44.3	26.0	37.0	19.6
RPN-R50 [50]	86.2	53.5	53.4	31.4	44.7	23.7
UPN (fine-grained) [22]	95.0	72.9	78.9	57.2	**54.4**	35.5
UPN (coarse-grained) [22]	95.4	73.2	76.6	55.6	53.3	34.8
WeDetect-Base-Uni	92.3	69.6	77.9	56.9	50.2	32.1
WeDetect-Large-Uni	**95.3**	**73.2**	**84.2**	**65.1**	53.3	**35.8**

Table 4. Zero-shot object retrieval results on common object detection datasets. As LVIS [19] is a federated dataset, we only compute the recall rates on it. To prevent setting a low threshold to get a high recall rate, we directly use the threshold used in COCO to evaluate LVIS.

Method	thre.	COCO			LVIS
		P	R	F1	R
OpenAI CLIP [48]	0.550	60.0	46.4	46.4	30.4
HQ-CLIP [63]	0.550	59.9	59.2	52.2	41.3
FG-CLIP2 [65]	0.001	67.9	62.4	57.7	43.1
WeDetect-Base-Uni	0.200	82.5	83.9	82.5	51.1
WeDetect-Large-Uni	0.200	**82.6**	**85.6**	**83.6**	**57.5**

object detection datasets COCO val [37] and LVIS val [19] as the benchmark datasets and use the category names as queries. The precision, recall, and F1 scores are first computed within each class and then averaged across different classes. As LVIS is a federated dataset where not all classes within the image are annotated, we only compute the recall rates on it. We select OpenAI CLIP ViT-large-patch14-336 [48], HQ-CLIP-base [63] which includes hard negative samples for training, and FG-CLIP2-so400M [65] which is optimized for fine-grained perception as the image-level perception model for comparisons. As shown in Table 4, our WeDetect-Large-Uni with 300 proposals significantly outperforms the image-level baselines, showing that the object retrieval task is complementary to the image-text retrieval task and WeDetect-Uni enjoys unique advantages on fine-grained perception.

WeDetect-Ref excels in REC tasks with many fewer parameters and a much faster inference speed. We select RefCOCO [26], RefCOCO+ [69], RefCOCOg [45], and HumanRef [24] as the REC benchmarks. As shown in Table 2, our WeDetect-Ref 4B achieves the highest 93.2 average scores on refcoco/+/g with only 4B parameters and top 100 proposals from WeDetect-Base-Uni, outperforming our baseline Qwen3-VL 4B [1] by 6.6 points and other much larger models with thinking ability. Importantly, since we formulate the REC task as a retrieval task and discard the next-token prediction mechanism, WeDetect-Ref 4B runs at an extremely fast speed, exceeding Qwen3-VL 4B by 13 times and even faster than Grounding-DINO-L [41]. In our scenarios, each image will be represented as 900-1600 tokens and 100 proposals are used, containing around 1000 tokens per image. The plain and simple architecture (containing only self-attention layers) can be easily accelerated by modern GPUs and Flash Attention [8]. Further, the inference time of WeDetect-Ref is consistent with the number of target objects, while the time for methods with the next-token prediction will increase linearly. The advantages in both accuracy and inference speed demonstrate the effec-

Table 5. Evaluating WeDetect-Ref on common object detection datasets. * indicates evaluation under a simplified setting where only ground-truth categories are queried.

Method	COCO				ODinW13
	AP	AP_s	AP_m	AP_l	AP
Grounding-DINO-T [41]	48.4	-	-	-	51.4
Qwen2.5-VL 7B [2]	17.7	-	-	-	37.3*
ChatRex 7B [22]	48.2*	-	-	-	-
PaDT Pro 7B [55]	39.0	-	-	-	-
VLM-FO1 3B [40]	44.4	-	-	-	44.0
LMM-Det 7B [33]	47.5	34.7	51.8	60.3	-
Qwen3-VL 8B [1]	-	-	-	-	44.7
WeDetect-Ref 4B	50.0	34.7	57.6	69.2	47.3

Table 6. Ablation studies on WeDetect-Base. Experiments are performed on LVIS minival.

Exp	Model	AP	AP_r	AP_c	AP_f
	WeDetect-Base	47.3	43.5	45.9	49.3
(1)	w/o coarse-grained labels	46.4	41.9	44.7	48.6
(2)	w/o fine-grained labels	45.1	41.1	43.2	47.8
(3)	w/o staged-wise training	45.5	41.2	43.7	47.9

tiveness of the new paradigm.

WeDetect-Ref also performs well in multi-class multi-instance object detection tasks. Although object detection is a relatively easy task for traditional detectors, it poses great challenges to LMMs for two reasons: First, as images may contain multiple instances, LMMs with next-token prediction tend to predict a small part of objects, resulting in low recall rates. Second, recent LMMs tend to predict objects for every query, failing to reject negative queries with no objects existing in the image. Therefore, Qwen2.5-VL 7B [2] gets only 17.7 AP on the COCO dataset. In contrast, WeDetect-Ref retrieves objects from a candidate list and each object is decoded independently, which ensures high recall rates and it is the first LMM exceeding 50 AP on the COCO dataset, for the first time matching the performance of traditional object detectors. WeDetect-Ref also achieves a high 47.3 mAP on ODinW13. We provide more implementation details in the Appendix.

6.2. Ablation Study

Effect of different training strategies for WeDetect. In this work, we initialize WeDetect with a pretrained CLIP model and finetune it with our grounding dataset with multi-granularity labels. For multi-granularity labels, we randomly sample one of the last two labels from the list, the finest one and the second finest one. Without multi-granularity labels, the text diversity is greatly reduced and can not achieve fine-grained vision-language alignment. In Table 6 (1) and (2), it reduces 1.6 AP_r with only the fine-grained labels and reduces 2.2 AP with only the coarse-grained labels. Using multi-granularity labels simultaneously achieves the highest performance. We also propose a staged-wise training recipe by first training the random-initialized head and neck and then training the model as a whole. In Table 6 (3), without initializing the head and neck, the pretrained CLIP features will be disturbed, decreasing 1.8 AP.

Table 7. Ablation studies on WeDetect-Ref. Models are trained with a part of data.

Exp	Model	RefCOCO	COCO			
		Avg.	AP	AP_s	AP_m	AP_l
	WeDetect-Ref 4B	93.0	47.5	29.6	54.7	67.9
(1)	BCE loss	92.8	43.1	29.0	52.5	60.7
(2)	w/o negative det data	93.1	42.1	22.6	48.0	63.7
(3)	25 tokens per object	93.3	47.6	30.8	55.3	67.5
(4)	shuffle proposals	93.1	47.6	30.2	54.9	67.7

Effect of different design choices for WeDetect-Ref. As shown in Table 7: (1) Sigmoid focal loss is a standard loss function in object detection and it is also suitable for WeDetect-Ref. Replacing it with binary cross-entropy (BCE) loss leads to a notable drop of 4.4 AP on COCO. (2) Negative supervision is crucial for learning a robust model. For object detection datasets, the classes that do not exist in the image serve as the negative queries. Adding negative detection data can significantly increase 5.4 AP and 7.0 AP_s. (3) In this work, we represent each object with a single token to maintain efficiency. We find that increasing the token count per object to 25 brings marginal improvement while inflating the context length by a factor of 25. We hypothesize that the full-image context already preserves sufficient object details, making a single token sufficient to establish an effective correspondence between the object and the global image representation. (4) As LLM adopts causal masks in attention layers, we study whether the proposal order will affect the performance. We observe that positive objects can appear at arbitrary positions within the token sequence during training. As a result, the model develops robustness to proposal ordering, and evaluation performance remains stable even when proposals are randomly shuffled.

7. Conclusion

In this work, we propose a family of open-vocabulary detection models following the retrieval methodology, in which targets are simply picked up from a candidate list, rather than generating a query-specific temporary candidate list. This design principle does not use computationally intensive cross-modal fusion layers and ensures a high inference speed. Following the methodology, we propose (1) WeDetect, which is an open-vocabulary object detection foundation, (2) WeDetect-Uni, which is a universal proposal generator and can be applied to a new object retrieval application, and (3) WeDetect-Ref, an LLM-based REC model discarding next-token prediction. The WeDetect model family attains state-of-the-art performance across 15 diverse benchmarks, demonstrating strong generalization ability, and still runs at an extremely fast speed.

References

[1] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025. 2, 5, 7, 8

[2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 2, 3, 7, 8

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 1

[4] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021. 3

[5] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024. 2, 7

[6] Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-world: Real-time open-vocabulary object detection. In *CVPR*, 2024. 2, 3, 6

[7] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *ACL*, 2020. 3

[8] Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *ICLR*, 2024. 7

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 4

[10] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 2

[11] Zi-Yi Dou, Aishwarya Kamath, Zhe Gan, Pengchuan Zhang, Jianfeng Wang, Linjie Li, Zicheng Liu, Ce Liu, Yann LeCun, Nanyun Peng, et al. Coarse-to-fine vision-language pre-training with fusion in the backbone. In *NeurIPS*, 2022. 2

[12] Penghui Du, Yu Wang, Yifan Sun, Luting Wang, Yue Liao, Gang Zhang, Errui Ding, Yan Wang, Jingdong Wang, and Si Liu. Lami-detr: Open-vocabulary detection with language model instruction. In *ECCV*, 2024. 1

[13] Shenghao Fu, Junkai Yan, Yipeng Gao, Xiaohua Xie, and Wei-Shi Zheng. Asag: Building strong one-decoder-layer sparse detectors via adaptive sparse anchor generation. In *ICCV*, 2023. 1

[14] Shenghao Fu, Junkai Yan, Qize Yang, Xihan Wei, Xiaohua Xie, and Wei-Shi Zheng. Frozen-detr: Enhancing detr with image understanding from frozen foundation models. In *NeurIPS*, 2024. 1, 2

[15] Shenghao Fu, Junkai Yan, Qize Yang, Xihan Wei, Xiaohua Xie, and Wei-Shi Zheng. A hierarchical semantic distillation framework for open-vocabulary object detection. *TMM*, 2025. 1, 2

[16] Shenghao Fu, Qize Yang, Qijie Mo, Junkai Yan, Xihan Wei, Jingke Meng, Xiaohua Xie, and Wei-Shi Zheng. Llmdet: Learning strong open-vocabulary object detectors under the supervision of large language models. In *CVPR*, 2025. 1, 2, 6

[17] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. In *ICLR*, 2021. 2

[18] Yuchen Guan, Chong Sun, Canmiao Fu, Zhipeng Huang, Chun Yuan, and Chen Li. Text-guided visual prompt dino for generic segmentation. In *ICCV*, 2025. 2

[19] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019. 1, 6, 7

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[21] Qing Jiang, Feng Li, Zhaoyang Zeng, Tianhe Ren, Shilong Liu, and Lei Zhang. T-rex2: Towards generic object detection via text-visual prompt synergy. In *ECCV*, 2024. 6

[22] Qing Jiang, Gen Luo, Yuqin Yang, Yuda Xiong, Yihao Chen, Zhaoyang Zeng, Tianhe Ren, and Lei Zhang. Chatrex: Taming multimodal llm for joint perception and understanding. *arXiv preprint arXiv:2411.18363*, 2024. 3, 5, 7, 8

[23] Qing Jiang, Junan Huo, Xingyu Chen, Yuda Xiong, Zhaoyang Zeng, Yihao Chen, Tianhe Ren, Junzhi Yu, and Lei Zhang. Detect anything via next point prediction. *arXiv preprint arXiv:2510.12798*, 2025. 7

[24] Qing Jiang, Lin Wu, Zhaoyang Zeng, Tianhe Ren, Yuda Xiong, Yihao Chen, Liu Qin, and Lei Zhang. Referring to any person. In *ICCV*, 2025. 3, 7

[25] Jiayu Jiao, Yu-Ming Tang, Kun-Yu Lin, Yipeng Gao, Andy J Ma, Yaowei Wang, and Wei-Shi Zheng. Dilateformer: Multi-scale dilated transformer for visual recognition. *TMM*, 2023. 1

[26] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014. 2, 7

[27] Dahun Kim, Tsung-Yi Lin, Anelia Angelova, In So Kweon, and Weicheng Kuo. Learning open-world object proposals without learning to classify. *IEEE Robotics and Automation Letters*, 2022. 7

[28] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, 2023. 3, 4

[29] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020. 4

[30] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *CVPR*, 2024. 3

[31] Chunyuan Li, Haotian Liu, Liunian Li, Pengchuan Zhang, Jyoti Aneja, Jianwei Yang, Ping Jin, Houdong Hu, Zicheng Liu, Yong Jae Lee, et al. Elevater: A benchmark and toolkit for evaluating language-augmented visual models. In *NeurIPS*, 2022. 6

[32] Chuyi Li, Lulu Li, Yifei Geng, Hongliang Jiang, Meng Cheng, Bo Zhang, Zaidan Ke, Xiaoming Xu, and Xiangxiang Chu. Yolov6 v3. 0: A full-scale reloading. *arXiv preprint arXiv:2301.05586*, 2023. 3

[33] Jincheng Li, Chunyu Xie, Ji Ao, Dawei Leng, and Yuhui Yin. Lmm-det: Make large multimodal models excel in object detection. In *ICCV*, 2025. 8

[34] Liunian Li, Zi-Yi Dou, Nanyun Peng, and Kai-Wei Chang. Desco: Learning object recognition with rich language descriptions. In *NeurIPS*, 2023. 2

[35] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *CVPR*, 2022. 1, 2, 3, 4, 6

[36] Long Lian, Yifan Ding, Yunhao Ge, Sifei Liu, Hanzi Mao, Boyi Li, Marco Pavone, Ming-Yu Liu, Trevor Darrell, Adam Yala, et al. Describe anything: Detailed localized image and video captioning. In *ICCV*, 2025. 2

[37] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 6, 7

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

[39] Chang Liu, Xudong Jiang, and Henghui Ding. Primitivenet: decomposing the global constraints for referring segmentation. *Visual Intelligence*, 2(1):16, 2024. 1

[40] Peng Liu, Haozhan Shen, Chunxin Fang, Zhicheng Sun, Jiajia Liao, and Tiancheng Zhao. Vlm-fo1: Bridging the gap between high-level reasoning and fine-grained perception in vlms. *arXiv preprint arXiv:2509.25916*, 2025. 3, 7, 8

[41] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *ECCV*, 2024. 1, 2, 6, 7, 8

[42] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022. 2, 3

[43] Yanxin Long, Youpeng Wen, Jianhua Han, Hang Xu, Pengzhen Ren, Wei Zhang, Shen Zhao, and Xiaodan Liang. Capdet: Unifying dense captioning and open-world detection pretraining. In *CVPR*, 2023. 2

[44] Muhammad Maaz, Hanoona Rasheed, Salman Khan, Fahad Shahbaz Khan, Rao Muhammad Anwer, and Ming-Hsuan Yang. Class-agnostic object detection with multi-modal transformer. In *ECCV*, 2022. 7

[45] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, 2016. 7

[46] Xiaofeng Mao, Yuefeng Chen, Yao Zhu, Da Chen, Hang Su, Rong Zhang, and Hui Xue. Coco-o: A benchmark for object detectors under natural distribution shifts. In *ICCV*, 2023. 6

[47] Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. In *NeurIPS*, 2023. 2

[48] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1, 2, 3, 7

[49] 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. In *CVPR*, 2023. 6

[50] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *TPAMI*, 2016. 1, 4, 7

[51] Tianhe Ren, Yihao Chen, Qing Jiang, Zhaoyang Zeng, Yuda Xiong, Wenlong Liu, Zhengyu Ma, Junyi Shen, Yuan Gao, Xiaoke Jiang, et al. Dino-x: A unified vision model for open-world object detection and understanding. *arXiv preprint arXiv:2411.14347*, 2024. 6

[52] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021. 3

[53] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *ICCV*, 2019. 4

[54] Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, et al. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*, 2025. 7

[55] Yongyi Su, Haojie Zhang, Shijie Li, Nanqing Liu, Jingyi Liao, Junyi Pan, Yuan Liu, Xiaofen Xing, Chong Sun, Chen Li, et al. Patch-as-decodable-token: Towards unified multi-modal vision tasks in mllms. *arXiv preprint arXiv:2510.01954*, 2025. 3, 8

[56] Hao Tang, Chenwei Xie, Haiyang Wang, Xiaoyi Bao, Tingyu Weng, Pandeng Li, Yun Zheng, and Liwei Wang. Ufo: A unified approach to fine-grained visual perception via open-ended language interface. In *NeurIPS*, 2025. 3

[57] Ao Wang, Lihao Liu, Hui Chen, Zijia Lin, Jungong Han, and Guiguang Ding. Yoloe: Real-time seeing anything. In *ICCV*, 2025. 6

[58] Hao Wang, Pengzhen Ren, Zequn Jie, Xiao Dong, Chengjian Feng, Yinlong Qian, Lin Ma, Dongmei Jiang, Yaowei Wang, Xiangyuan Lan, et al. Ov-dino: Unified open-vocabulary detection with language-aware selective fusion. *arXiv preprint arXiv:2407.07844*, 2024. 2, 6

[59] 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. In *ICCV*, 2023. 4

[60] Luting Wang, Yi Liu, Penghui Du, Zihan Ding, Yue Liao, Qiaosong Qi, Biaolong Chen, and Si Liu. Object-aware distillation pyramid for open-vocabulary object detection. In *CVPR*, 2023. 2

[61] Shihao Wang, Guo Chen, De-an Huang, Zhiqi Li, Minghan Li, Guilin Li, Jose M Alvarez, Lei Zhang, and Zhiding Yu. Videoitg: Multimodal video understanding with instructed temporal grounding. *arXiv preprint arXiv:2507.13353*, 2025. 5

[62] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025. 2, 7

[63] Zhixiang Wei, Guangting Wang, Xiaoxiao Ma, Ke Mei, Huaian Chen, Yi Jin, and Fengyun Rao. Hq-clip: Leveraging large vision-language models to create high-quality image-text datasets and clip models. In *ICCV*, 2025. 7

[64] Chunyu Xie, Heng Cai, Jincheng Li, Fanjing Kong, Xiaoyu Wu, Jianfei Song, Henrique Morimitsu, Lin Yao, Dexin Wang, Xiangzheng Zhang, et al. Ccmb: A large-scale chinese cross-modal benchmark. In *ACM MM*, 2023. 3

[65] Chunyu Xie, Bin Wang, Fanjing Kong, Jincheng Li, Dawei Liang, Ji Ao, Dawei Leng, and Yuhui Yin. Fg-clip 2: A bilingual fine-grained vision-language alignment model. *arXiv preprint arXiv:2510.10921*, 2025. 7

[66] 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. In *NeurIPS*, 2022. 2, 6

[67] Lewei Yao, Jianhua Han, Xiaodan Liang, Dan Xu, Wei Zhang, Zhenguo Li, and Hang Xu. Detclipv2: Scalable open-vocabulary object detection pre-training via word-region alignment. In *CVPR*, 2023. 2, 6

[68] 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. In *CVPR*, 2024. 2, 6

[69] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *ECCV*, 2016. 7

[70] Yuqian Yuan, Wentong Li, Jian Liu, Dongqi Tang, Xinjie Luo, Chi Qin, Lei Zhang, and Jianke Zhu. Osprey: Pixel understanding with visual instruction tuning. In *CVPR*, 2024. 2

[71] Yuqian Yuan, Hang Zhang, Wentong Li, Zesen Cheng, Boqiang Zhang, Long Li, Xin Li, Deli Zhao, Wenqiao Zhang, Yueting Zhuang, et al. Videorefer suite: Advancing spatial-temporal object understanding with video llm. In *CVPR*, 2025. 2

[72] Haotian Zhang, Pengchuan Zhang, Xiaowei Hu, Yen-Chun Chen, Liunian Li, Xiyang Dai, Lijuan Wang, Lu Yuan, Jenq-Neng Hwang, and Jianfeng Gao. Glipv2: Unifying localization and vision-language understanding. In *NeurIPS*, 2022. 2

[73] Shilong Zhang, Peize Sun, Shoufa Chen, Min Xiao, Wenqi Shao, Wenwei Zhang, Yu Liu, Kai Chen, and Ping Luo. Gpt4roi: Instruction tuning large language model on region-of-interest. In *ECCV*, 2024. 2

[74] Chuyang Zhao, YuXin Song, Junru Chen, Kang Rong, Haocheng Feng, Gang Zhang, Shufan Ji, Jingdong Wang, Errui Ding, and Yifan Sun. Octopus: A multi-modal llm with parallel recognition and sequential understanding. In *NeurIPS*, 2024. 3, 5, 7

[75] Shiyu Zhao, Long Zhao, Yumin Suh, Dimitris N Metaxas, Manmohan Chandraker, Samuel Schulter, et al. Generating enhanced negatives for training language-based object detectors. In *CVPR*, 2024. 2

[76] Xiangyu Zhao, Yicheng Chen, Shilin Xu, Xiangtai Li, Xinjiang Wang, Yining Li, and Haian Huang. An open and comprehensive pipeline for unified object grounding and detection. *arXiv preprint arXiv:2401.02361*, 2024. 6