Detect Anything via Next Point Prediction

Qing Jiang[‡], **Junan Huo, Xingyu Chen, Yuda Xiong, Zhaoyang Zeng,**
Yihao Chen, Tianhe Ren, Junzhi Yu, Lei Zhang[†]

International Digital Economy Academy (IDEA)

Rex-Omni.github.io
IDEA-Research/Rex-Omni

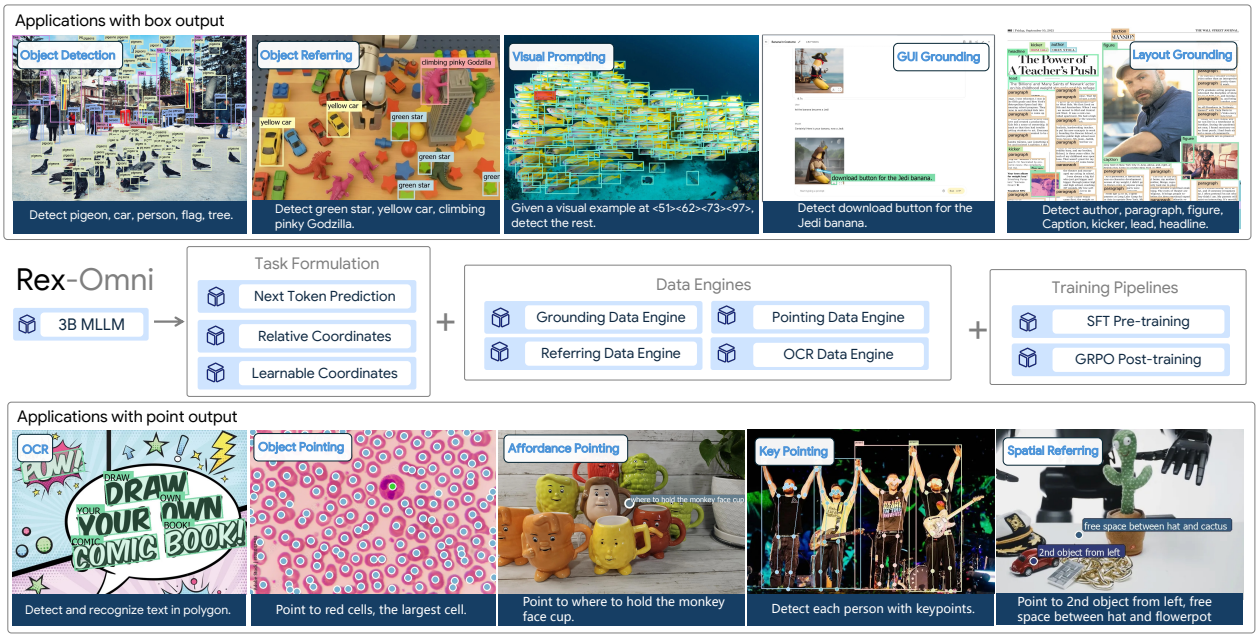


Figure 1: We introduce Rex-Omni, a 3B-parameter MLLM with strong visual perception capabilities.

Object detection has long been dominated by traditional coordinate regression-based models, such as YOLO, DETR, and Grounding DINO. Although recent efforts have attempted to leverage MLLMs to tackle this task, they face challenges like low recall rate, duplicate predictions, coordinate misalignment, etc. In this work, we bridge this gap and propose **Rex-Omni**, a 3B-scale MLLM that achieves state-of-the-art object perception performance. On benchmarks like COCO and LVIS, Rex-Omni attains performance comparable to or exceeding regression-based models (e.g., DINO, Grounding DINO) in a zero-shot setting. This is enabled by three key designs: **1) Task Formulation**: we use special tokens to represent quantized coordinates from 0 to 999, reducing the model's learning difficulty and improving token efficiency for coordinate prediction; **2) Data Engines**: we construct multiple data engines to generate high-quality grounding, referring, and pointing data, providing semantically rich supervision for training; **3) Training Pipelines**: we employ a two-stage training process, combining supervised fine-tuning on 22 million data with GRPO-based reinforcement post-training. This RL post-training leverages geometry-aware rewards to effectively bridge the discrete-to-continuous coordinate prediction gap, improve box accuracy, and mitigate undesirable behaviors like duplicate predictions that stem from the teacher-guided nature of the initial SFT stage. Beyond conventional detection, Rex-Omni's inherent language understanding enables versatile capabilities such as object referring, pointing, visual prompting, GUI grounding, spatial referring, OCR and key-pointing, all systematically evaluated on dedicated benchmarks. We believe that Rex-Omni paves the way for more versatile and language-aware visual perception systems.

[†]*Corresponding author: leizhang@idea.edu.cn;* [‡]*Project Lead: jiangqing@idea.edu.cn. This work was done when Qing and Junan were interns at IDEA, and Xingyu was on an academic visit at IDEA.*

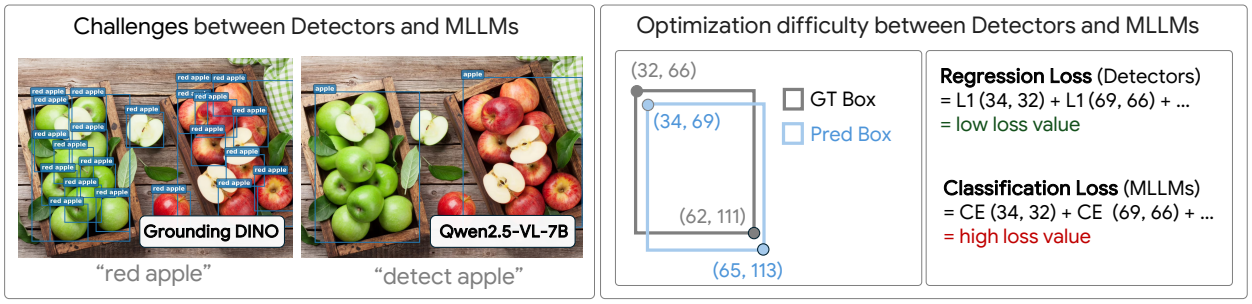


Figure 2: 1) Detectors excel in localization but lack language understanding. MLLMs understand language well but struggle with localization. 2) Differences in optimization difficulty between detectors and MLLMs.

1. Introduction

Object detection [23, 87, 86, 85, 8, 122, 102, 60, 58, 127, 38, 99, 20] has long been a foundational task in computer vision due to its broad applications. The field has progressed from early CNN-based architectures, such as YOLO [86] and Faster R-CNN [87], to Transformer-based models like DETR [8] and DINO [122], while the task itself has evolved from traditional closed-set detection to open-set detection [59, 49, 29, 13, 72, 88, 35, 71, 72] to better handle emerging real-world challenges.

A paramount objective in object detection is to develop models capable of identifying arbitrary objects and concepts. A common approach to this problem is open vocabulary object detection, where models such as Grounding DINO [59] and DINO-X [88] leverage text encoders (e.g., BERT [37] or CLIP [81]) to represent object categories and perform category-level open-set detection. Despite their effectiveness, these methods are fundamentally constrained by their relatively shallow language understanding, which restricts their ability to handle complex semantic descriptions (In Figure 2, Grounding DINO detected all apples, despite the input prompt being *red apple*.). Consequently, these methods are inherently limited in fully addressing this objective.

In contrast, multimodal large language models (MLLMs) [74, 101, 65, 107, 11, 1, 44, 18, 105] benefit from the strong language understanding capabilities of their underlying LLMs, presenting a promising avenue for integrating advanced language comprehension into object detection. A common MLLM-based approach [9, 116, 106, 120, 123, 30, 69, 33, 133, 24, 4] is to represent coordinates as discrete tokens [10] and predict bounding boxes through next-token prediction. While conceptually elegant, existing MLLM-based methods have rarely matched the performance of traditional regression-based detectors on benchmarks such as COCO. As exemplified in Figure 2, even advanced MLLMs like Qwen2.5-VL [4] struggles with precise object localization, in addition to facing limitations such as low recall rates, coordinate drift, and duplicate predictions.

We argue that the performance disparity in MLLM-based object detection primarily arises from two fundamental challenges inherent in their current formulation and training. First, MLLMs typically treat coordinate prediction as a discrete classification task, directly generating absolute coordinate values and relying on cross-entropy loss for supervision. While traditional regression-based models benefit from continuous, geometry-aware losses (e.g., L1, GIoU) that are directly sensitive to small geometric offsets, MLLMs face a significant learning difficulty in accurately mapping a fixed set of discrete tokens to the continuous pixel space. As illustrated in Figure 2, even minor pixel misalignments in discrete coordinate predictions can result in disproportionately large cross-entropy losses, hindering precise localization. This inherent challenge underscores the need for effective strategies to reduce coordinate learning complexity and provide extensive data for this mapping.

Secondly, MLLMs commonly employ Supervised Fine-tuning (SFT) for teacher-guided next-token

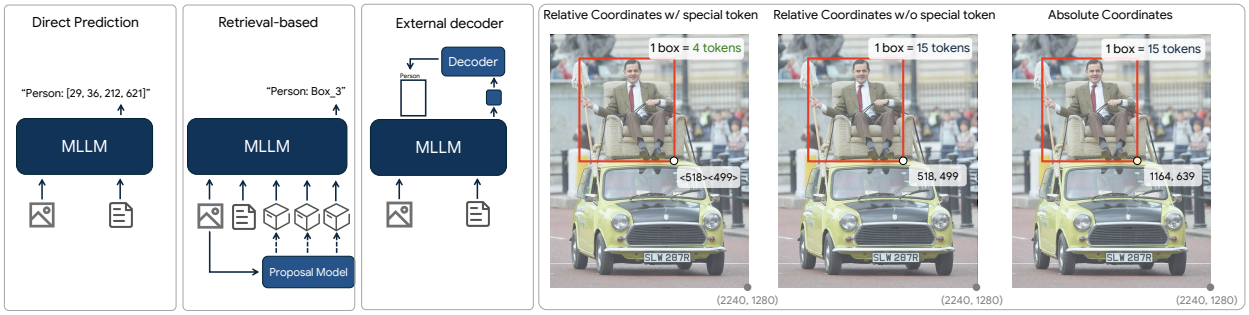
prediction training [79]. While efficient, this paradigm creates a fundamental mismatch between training and inference. During SFT, the model is always conditioned on a ground-truth prefix, namely teacher forcing, meaning it is never exposed to its own, potentially imperfect, predictions. This training setup fails to capture the model's true performance in an autonomous generation setting. This inherently prevents the model from developing robust behavioral awareness. Consequently, during free-form inference without this direct guidance, the model often struggles to regulate its own output structure. This leads to anomalous coordinate sequence generation, such as spurious duplicate predictions or object omissions, which undermine its overall performance. Addressing these two intertwined challenges is crucial for advancing MLLM-based object detection.

To overcome these inherent limitations and unlock the full potential of MLLMs for precise and versatile object perception, we propose **Rex-Omni**, a 3B-scale MLLM that achieves competitive performance with traditional detectors while distinctly excelling in language understanding capabilities. We address the aforementioned challenges through three core design principles:

- **Task Formulation**: We unify visual perception tasks under a coordinate prediction framework, wherein each task is formulated as generating a sequence of coordinates. Specifically, pointing predicts one point, detection employs two points to form a bounding box, polygons use four or more points to represent object contours, and keypoint tasks output multiple semantic points. We adopt a quantized coordinate representation, where each coordinate value is mapped to one of 1,000 discrete tokens corresponding to values from 0 to 999. This approach significantly reduces the coordinate learning complexity and eases optimization, concurrently enhancing the efficiency of spatial representation.
- **Data Engines**: To facilitate the model's learning of the mapping between 1,000 discrete coordinate tokens and pixel-level positions, and to foster robust comprehension of complex natural language expressions, we design multiple specialized data engines for grounding, referring, and pointing tasks. These engines generate high-quality, semantically rich visual supervision signals for coordinate prediction.
- **Training Pipelines**: We adopt a two-stage training paradigm. In the first stage, we perform supervised fine-tuning on 22 million data to teach the model basic coordinate prediction skills. In the second stage, we apply GRPO-based [92] reinforcement post-training with three geometry-aware reward functions. This reinforcement phase serves two purposes: it enhances the precision of coordinate predictions through continuous geometric supervision, and crucially, it mitigates undesired behaviors (such as duplicate predictions) that arise from the teacher-guided nature of the initial SFT stage.

After this two-stage training, Rex-Omni achieves superior performance across a diverse range of perception tasks, as shown in Figure 1., including object detection, object referring, visual prompting, GUI grounding, layout grounding, OCR, pointing, keypointing, and spatial referring. All of these tasks are achieved through direct prediction of coordinate points. To quantitatively assess its performance, Rex-Omni is first evaluated on COCO [53], a core benchmark for object detection. In a zero-shot setting (without training on COCO data), Rex-Omni demonstrates superior F1-score performance compared to traditional coordinate regression-based models (e.g., DINO-ResNet50, Grounding DINO) and other MLLMs (e.g., SEED1.5-VL [24]). Beyond COCO, Rex-Omni's performance is further benchmarked across diverse tasks, such as long-tailed detection, referring expression comprehension, dense object detection, GUI grounding, and OCR. Rex-Omni consistently outperforms both traditional detectors and MLLMs, thereby establishing a unified framework that combines precise localization with robust language understanding.

In summary, Rex-Omni represents a significant step towards unifying robust language understanding with precise visual perception. By carefully integrating principled task formulations, advanced



(a) (b)

Figure 3: Design philosophy of Rex-Omni for coordinate prediction. It illustrates our chosen approach: (a) adopting a direct coordinate prediction strategy, and (b) employing a quantized relative coordinate format represented by special tokens for efficient and robust spatial encoding.

data engines, and a sophisticated two-stage training pipeline, we demonstrate that MLLMs have the profound potential to define the next generation of object detection models, offering unprecedented versatility and a truly language-aware approach to visual perception systems.

2. Task Formulation

In this section, we present Rex-Omni's task formulation design, covering its coordinate representation, the specific output formats for different tasks, and the details of its model architecture.

2.1. Coordinate Formulation

We begin by defining the output formulation for coordinate prediction. Existing approaches to utilizing MLLMs for this task can be broadly categorized into three paradigms, as illustrated in Figure 3a: **1) Direct Coordinate Prediction:** Inspired by the Pix2Seq [10] paradigm, these method [9, 116, 106, 120, 123] treats coordinate values as discrete tokens within the language model's vocabulary, enabling the model to directly generate coordinate outputs; **2) Retrieval-based Methods:** This approach [30, 69, 31, 33] incorporates an additional proposal module. The LLM is trained to predict the index of a candidate region or bounding box, thereby representing the output as a retrieval task over predefined proposals; **3) External decoder:** In this strategy [121, 108, 42, 61], the LLM predicts special tokens, whose corresponding embeddings are then passed to an external decoder responsible for producing the final coordinates. We adopt the direct coordinate prediction strategy for Rex-Omni, motivated by its simplicity, flexibility, and the advantage of not relying on external modules or additional supervision.

Within the direct coordinate prediction paradigm, several variations exist, as illustrated in Figure 3b: **1) Relative coordinates with special tokens:** Coordinates are quantized to values between 0 and 999, with each coordinate represented by a special token in the LLM's vocabulary. The model is thus trained to predict these 1,000 tokens as representations for coordinates. A representative model is Pix2Seq [10]. **2) Relative coordinates without special tokens:** Coordinates are similarly quantized to 1,000 bins; however, they are represented by multiple atomic tokens rather than a single special token. A representative model is SEED1.5-VL [24]. **3) Absolute coordinates:** This method uses absolute coordinates, where a coordinate value such as 1921 is tokenized into individual digits (1, 9, 2, 1). A representative model is Qwen2.5-VL [4]. We choose relative coordinates with special tokens modeling approach for two primary reasons: First, selecting relative coordinates over absolute coordinates inherently reduces learning complexity by confining the classification task to a bounded

range of 1,000 categories. Second, the use of dedicated special tokens for coordinates significantly reduces the required token length per coordinate. For instance, a bounding box is represented by only four special tokens, in contrast to 15 atomic tokens (including separators) without such a scheme. This significantly improves token efficiency and inference speed, especially in dense object scenes.

2.2. Input Format

Rex-Omni adopts a unified text-based interface for all visual perception tasks. Each task is expressed as a natural language query that specifies the target objects or relationships to be identified in the image. This design allows the model to seamlessly integrate diverse vision-language tasks under a single instruction-driven framework.

Text Prompts. For most tasks, the model receives an image paired with a text prompt formulated in natural language. The text prompt can describe one or multiple. When multiple targets are specified, their corresponding categories or referring expressions are concatenated using commas. For example:

Example of a text prompt for multi-object detection

Please detect pigeon, person, truck, snow in this image. Return the output in box format.

For different tasks, we design distinct query styles to guide the model for generation.

Visual Prompts. While text prompts offer strong generalization and interpretability, they face limitations when dealing with objects that lack clear linguistic descriptions—particularly rare or visually complex categories. As shown in prior work such as T-Rex2 [32], certain objects are inherently difficult to express through text alone. To address this, Rex-Omni supports visual prompting, allowing users to provide bounding boxes as an additional and intuitive form of input.

Unlike existing methods [32, 88, 28] that treat visual prompts as feature-matching problems by extracting embeddings from the indicated region and comparing them to detection queries, Rex-Omni adopts a unified text-based interface. Given a visual prompt in box format, the corresponding region is first converted into quantized coordinate tokens. The model is then guided through natural language instructions to identify all objects that share the same category as the indicated region. This design seamlessly integrates visual prompting into the generative text framework, enabling the model to reason about visual correspondence through language.

An example of visual prompting in Rex-Omni

Here are some example boxes specifying the location of several objects in the image: "object1": ["<12><412><339><568>", "<92><55><179><378>"]. Please detect all objects with the same category and return their bounding boxes in [x0, y0, x1, y1] format.

2.3. Output Format for Each Task

The output for each visual task is uniformly represented as a structured token sequence that includes descriptive phrases, coordinate tokens, and special tokens for demarcation, organized as follows::

Basic output format of Rex-Omni

<|object_ref_start|>**PHRASE**<|object_ref_end|><|box_start|> **COORDS**<|box_end|>

Here, **PHRASE** denotes the category or description of the object represented by the coordinate sequence, and **COORDS** refers to the sequence of coordinates. Rex-Omni is bulid upon Qwen2.5-VL-3B and we retain Qwen2.5-VL's original special tokens for task formatting, including the phrase start token (<object_ref_start>), phrase end token (<object_ref_end>), coordinate start token (<box_start>), and coordinate end token (<box_end>).

For tasks involving outputting boxes, such as object detection, COORDS consists of a sequence of coordinates in the format of [x0, y0, x1, y1], sorted by x0 in ascending order. For example:

An example of a task for outputting bounding boxes

<|object_ref_start|>person<|object_ref_end|><|box_start|><12><42><512><612>, <24><66><172><623>, ...<|box_end|>, ... (more phrases)

For tasks involving outputting points, such as object pointing, COORDS is composed of a sequence of [x0, y0] pairs. For example:

An example of a task for outputting points

<|object_ref_start|>button<|object_ref_end|><|box_start|><100><150>,<200><250>, ...<|box_end|>, ... (more phrases)

For tasks involving outputting polygons, such as OCR, COORDS consists of a sequence of coordinates in the format [x0, y0, x1, y1, x2, y2, ...]. For example:

An example of a task for outputting polygons

<|object_ref_start|>text<|object_ref_end|><|box_start|><10><20>...<|box_end|>, ... (more phrases)

For the keypointing task, we output a structured JSON format that includes both the bounding box of the object and its associated keypoints.

An example of keypoint detection task

{"person1": {"box": <0><123><42><256>, "keypoints": {"left eye": <32><43>, "right eye": <66><55>, ...}}, {"person2": {"box": <51><116><72><522>, "keypoints": {"left eye": <342><23>, "right eye": <16><571>, ...}}}

For the simultaneous detection of multiple phrases, the predicted outputs corresponding to different phrases are concatenated using commas. If a particular phrase refers to an object that is not present in the image, the corresponding COORDS field is replaced with **None**.

2.4. Model Architecture

As shown in Figure 4, Rex-Omni is built upon the Qwen2.5-VL-3B-Instruct model with minimal architectural modifications. While the original Qwen2.5-VL employs an absolute coordinate encoding scheme, we adapt the model to support relative coordinate representations without introducing additional parameters. Specifically, we repurpose the final 1,000 tokens of the model's vocabulary to serve as special tokens, each corresponding to a quantized coordinate ranging from 0 to 999.

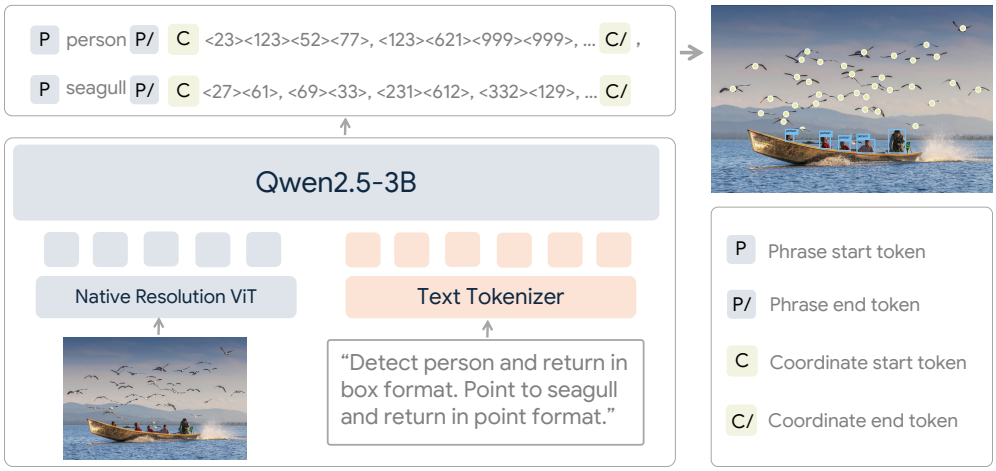


Figure 4: Overview of the Rex-Omni Model Architecture. Rex-Omni is constructed upon the Qwen2.5-VL-3B backbone with minimal architectural modifications. Notably, the last 1,000 tokens of the original vocabulary are repurposed to serve as dedicated special tokens, representing quantized coordinate values from 0 to 999.

3. Training Data

To equip Rex-Omni with both precise coordinate prediction capabilities and strong language understanding, we utilize two sources of training data: publicly available datasets and automatically annotated data generated by our custom-designed data engines.

3.1. Public Datasets

In Table 1, we enumerate the publicly available datasets leveraged for Rex-Omni's training across various subtasks, including Object Detection, Object Referring, Visual Prompting, OCR, Layout Grounding, GUI Grounding, Pointing, Affordance Grounding, Spatial Referring, and Keypointing. For each of these tasks, a set of question templates was defined to construct corresponding question-answer (QA) pairs. In total, approximately 8.9 million public data samples were utilized.

3.2. Data Engines

Effective training of Rex-Omni necessitates learning a fine-grained mapping between its 1,000 quantized coordinate tokens and the continuous pixel space of images. This capability demands a substantially larger volume of high-quality training data than what is conventionally available in existing public datasets. Moreover, while many public datasets offer category-level annotations, those providing richer, instance-level semantic grounding (e.g., referring expressions) remain scarce in both scale and diversity. To address these limitations, we developed a dedicated suite of data engines engineered to generate high-quality, large-scale training data specifically tailored for fine-grained spatial reasoning and complex language grounding tasks.

3.2.1. Grounding Data Engine

A common strategy for constructing large-scale detection datasets is to develop a grounding data engine [29, 88, 89, 13, 77], which typically involves generating image captions, extracting candidate phrases, and using a grounding model (e.g., Grounding DINO) to assign bounding boxes to those phrases. In contrast to prior approaches, we introduce a phrase filtering stage into the pipeline to

Task	Output Format	Question Template Example	Datasets
Object Detection	Box	Detect [PHRASE] in this image	APTv2 [114], BDD100K [117], DeepFashion [62] DOTAv2 [111], EgoObjects [132], FAIR-1M [97] HumanParts [51], ImageNet-Part [26] NuImages [7] PACO [82], OpenImages [40], O365 [91] V3Det [104], VisDrone [19]
Object Referring	Box	Detect [PHRASE] in this image	HumanRef, RefCOCOg
Visual Prompting	Box	Given reference boxes [BOX] indicating one or more objects, find all objects with the same category	O365 [91], OpenImages [40], HierText [63] CrowdHuman [90], SA-1B [39], VisDrone [19] FSCD147 [73]
OCR	Box / Polygon	Detect all the text in box/polygon format and recognize them	Art [14], HierText [63], ICDAR2013 [66] ICDAR2015 [36], LSVT [98], RCTW [93] ReCTS [125] SROIE [27], TextOCR [95] IDLOCR [5], WildReceipt [96]
Layout Groudning	Box	Detect [PHRASE] in this image	DocLayNet [78], PubLayNet [129], TableBank [50] M6Doc [12], CDLA [46], TabRecSet [112]
GUI Grounding	Box / Point	Detect/Point to element [PHRASE]	Os-Atlas [109], UI-RefExp [3], ShowUI [52]
Pointing	Point	Point to [PHRASE]	Pixmo-point [18]
Affordance	Point	Point to [PHRASE]	AGD20K [67]
Spatial Referring	Point	Point to [PHRASE]	RefSpatial [131]
KeyPointing	Box & Point	Can you detect each [PHRASE] in the image in box format, and then provide the coordinates of its [KEYPOINT] as [x0, y0]? Output the answer in JSON format.	AP10K [118], APT36K [113], COCO-Keypoint [54] MacaquePose [41], HumanArt [34], MPII [2] OCHuman [126], CrowdPose [47]

Table 1: Publicly available training datasets used by Rex-Omni, covering tasks such as object detection, referring, prompting, OCR, grounding, pointing, affordance, and keypointing, with outputs including boxes, points, polygons, and JSON-formatted keypoints.

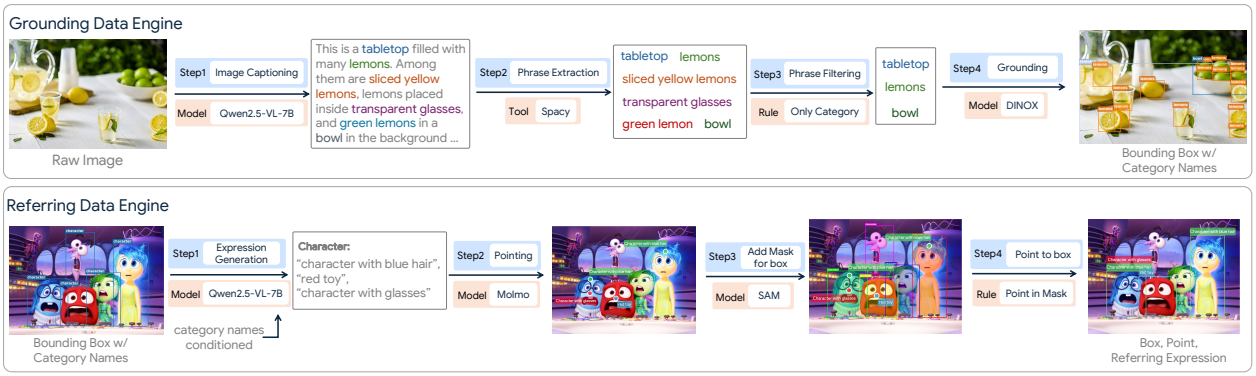


Figure 5: Pipelines of our two primary data engines. The figure illustrates the processes of the Grounding Data Engine (top) and the Referring Data Engine (bottom), which are custom-designed to produce extensive, high-quality grounding and referring data for Rex-Omni's training.

improve annotation quality. Specifically, our annotation process consists of the following four stages:

- **Image Captioning:** We begin by generating descriptive captions for each image using Qwen2.5-VL-7B-Instruct. These captions provide natural language descriptions of the visual content, typically covering multiple objects within the scene.
- **Phrase Extraction:** We then apply the SpaCy[1] NLP toolkit to extract noun phrases from the generated captions. These phrases may include basic class names (e.g., tabletop, lemon) as well as more specific descriptions (e.g., sliced yellow lemons, green lemons).
- **Phrase Filtering:** This step marks a key departure from prior approaches. To minimize data ambiguity, we remove noun phrases containing descriptive attributes such as adjectives (e.g., green lemon is discarded, while lemon is retained). The rationale is that current grounding models struggle to accurately interpret such descriptive expressions, often detecting all instances of a category regardless of the modifier. For instance, the phrase green lemon may incorrectly

[1] https://spacy.io/

trigger detections of all lemons, thereby introducing significant labeling errors.
- **Phrase Grounding:** Finally, we use DINO-X [88], an open-vocabulary object detector, to produce bounding boxes corresponding to the filtered phrases.

For this data engine, images are primarily sourced from the COYO [6] and SA-1B [39] datasets. We apply rigorous preprocessing, including discarding low-resolution images and filtering content labeled as NSFW. This process yields a curated dataset of approximately 3 million images, each annotated with high-quality grounding labels.

3.2.2. Referring Data Engine

Unlike detection or grounding data, which primarily emphasize object category names, referring data necessitate semantically richer natural language descriptions, exemplified by phrases like "a man in a yellow shirt". The RexSeek [33] study underscores that high-quality referring annotations should accommodate a single referring expression mapping to multiple instances, thereby fostering the model's ability to learn flexible and context-aware reference grounding. However, RexSeek's reliance on manual annotation renders it labor-intensive and inherently unscalable. To address this limitation, we design a fully automated referring data engine capable of generating large-scale referring data without human supervision.

- **Expression Generation:** Given an image annotated with bounding boxes and corresponding category labels, we prompt Qwen2.5-VL-7B with the image and category information to generate a set of referring expressions. Each expression is designed to naturally describe an object category present in the image, mimicking human-like descriptions.
- **Pointing:** For each generated referring expression, we employ Molmo [18], a state-of-the-art referring model, to produce the corresponding spatial point. Although Molmo outputs only point-level predictions, it exhibits strong performance in understanding and grounding referring expressions.
- **Mask Generation:** We apply SAM [39] to generate a mask for each ground-truth bounding box in the image.
- **Point-to-Box Association:** Each point produced by Molmo is aligned with a SAM-generated mask. When a point lies within a mask, the corresponding bounding box is linked to the referring expression, thereby grounding the language in the object region.

For this data engine, we use images from O365 [91], OpenImages [40], and additional data generated by our Grounding Data Engine. Through this pipeline, we obtain approximately 3 million images with automatically generated referring annotations.

3.2.3. Other Data Engines

In addition to grounding and referring data, we develop two relatively lightweight data engines to generate datasets for pointing and OCR tasks.

- **Pointing Data Engine:** Point-level supervision offers an efficient alternative to bounding boxes, particularly when object boundaries are ambiguous or difficult to delineate (e.g., edges, whitespace, or fine structures). To derive point annotations from box-level supervision, we adopt a geometry-aware strategy. Given a bounding box, SAM is first used to obtain the corresponding segmentation mask. We then compute the minimum-area enclosing rotated rectangle of the

mask and take the intersection of its diagonals as the candidate point. If this point lies within the mask, it is designated as the point annotation for the box. Through this conversion, we obtain approximately 5 million point-level samples from existing detection datasets as well as from the outputs of our grounding and referring data engines.

- **OCR Data Engine:** PaddleOCR[2] is utilized to annotate images containing textual content, extracting both polygonal boundaries of text regions and their corresponding transcriptions. For each extracted polygon, the minimum enclosing axis-aligned rectangle is subsequently computed to serve as its bounding box representation. Images are sourced from the COYO dataset, yielding approximately 2 million OCR-annotated samples.

In total, combining publicly available datasets and data generated by our annotation pipelines, we obtain 22 million high-quality annotated images for training.

4. Training Pipelines

We employ a two-stage training strategy, depicted in Figure 6. In the first stage, supervised fine-tuning (SFT) is performed on 22 million annotated samples using a teacher-guided approach, enabling the model to acquire fundamental coordinate prediction capabilities. In the second stage, we apply reinforcement learning based on the GRPO framework, which further refines the model's performance by combining geometry-aware rewards with behavior-aware optimization, thus addressing the limitations of the SFT stage and enhancing overall prediction quality.

4.1. Stage1: Supervised Fine-Tuning

Since the model predicts coordinates in the form of quantized tokens ranging from 0 to 999, it must first learn how to accurately map these discrete values back to continuous pixel locations within the image. This corresponds to a 1,000 way classification problem over spatial positions, which requires substantial supervision to achieve reliable performance. Therefore, we begin training with a teacher-guided supervised fine-tuning stage on large-scale annotated data, enabling the model to acquire the fundamental ability to interpret and predict spatial coordinates.

We adopt the following online strategy to construct SFT conversation data:

- **Conversation Templates:** For each training task, we construct multiple question templates with GPT-4o to mimic real user scenarios. These templates include placeholders for **PHRASE** keywords, which are replaced with actual phrases from the data during training.
- **Multi-Phrase Queries:** In practical settings, users may wish to detect multiple object categories within a single image. To reflect this, if an image contains N annotated phrases, we randomly sample between 1 and N phrases to form the training query.
- **Visual Prompting Training:** Following T-Rex2 [32], for each training sample consisting of an image and its annotated category-specific bounding boxes, we simulate visual prompting scenarios. Specifically, for each category present in the image, we randomly sample between 1 and N bounding boxes, where N denotes the maximum number of annotated instances for that category. These sampled boxes are treated as visual prompts and converted into quantized coordinate tokens consistent with our coordinate formulation. The model is then instructed, through natural language queries, to detect all objects of the same category as indicated by the given visual prompts.

[2] https://github.com/PaddlePaddle/PaddleOCR

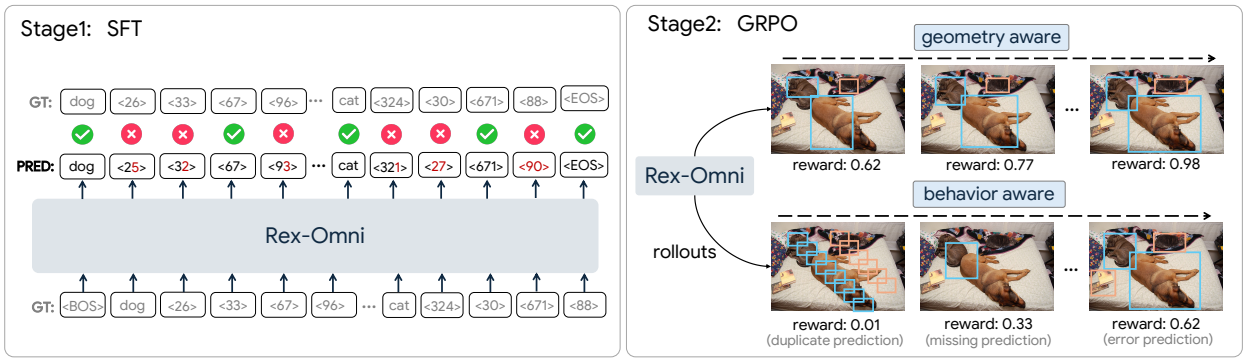


Figure 6: Overview of Rex-Omni's two-stage training pipeline. The first stage involves supervised fine-tuning (SFT) on 22 million samples to establish fundamental coordinate prediction capabilities. This is followed by GRPO-based reinforcement post-training, which leverages geometry-aware rewards and behavior-aware optimization to refine precision and correct SFT-induced behavioral deficiencies.

We adopt the standard cross-entropy loss for training. The model is trained on 8 nodes, each equipped with 8 A100 GPUs, and the total training time is approximately 8 days. All model parameters are updated during training. We use separate learning rates for different components: 2e-6 for the vision encoder, and 2e-5 for both the projection layer and the LLM. Optimization is performed using the AdamW [64] optimizer with a learning rate warm-up of 3% and a weight decay of 0.01. Following the architecture of Qwen2.5-VL, Rex-Omni also employs a native resolution Vision Transformer as its vision encoder. We constrain the number of input pixels to range from a minimum of $16 \times 28 \times 28$ to a maximum of $2560 \times 28 \times 28$. Given a ViT patch size of 28, this limits the number of image tokens between 16 and 2560.

4.2. Stage2: Reinforcement Post-Training

4.2.1. Limitations of SFT

While SFT allows the model to quickly acquire basic coordinate prediction capabilities by leveraging massive amounts of labeled data, it presents two key limitations:

Geometric Discretization Issue. Using cross-entropy loss for coordinate prediction inherently introduces a discretization problem. Coordinates are represented as categorical tokens (from <0> to <999>), and the model is trained to classify each token exactly. However, this formulation is misaligned with the continuous nature of geometry in spatial tasks. For example, if the ground-truth token is <33> but the model predicts <32>, the difference in pixel space may be negligible, yet the CE loss penalizes it as a completely incorrect prediction. Conversely, if the ground truth is <0><0><100><100> but the model predicts <0><0><100><1000>, only one token is misclassified. In this case, the CE loss remains relatively small, even though the resulting bounding box is severely misaligned and the geometric error is substantial.

Behavioral Regulation Deficiency. In the SFT stage, teacher-forced training relies on full ground-truth sequences for efficient parallel learning. This setup fixes the number of predicted boxes to the ground-truth count, preventing the model from autonomously learning how many objects to predict. Consequently, during inference the model often fails to regulate output quantity, leading to two typical errors: (1) predicting fewer boxes than required (missed detections), or (2) predicting more boxes than necessary (repetitive detections with identical or slightly shifted coordinates). These behaviors reflect the model's lack of effective output regulation.

4.2.2. GRPO-based Post-Training

To address the geometry and behavior-related limitations of SFT, we adopt a reinforcement post-training strategy based on GRPO [92]. GRPO enables the model to explore its own output space and improve through reward-guided optimization. Given an image and a question (I, x), the model samples a group of G complete responses $\{o_1, o_2, \ldots, o_G\}$ from the current policy π_θ. Each response consists of a full reasoning trace and a final set of predicted coordinates or boxes, depending on the task. For each output o_i, we compute a scalar reward r_i, which is normalized across the group to obtain the *relative advantage*:

$$A_i = \frac{r_i - \text{mean}(r_1, \ldots, r_G)}{\text{std}(r_1, \ldots, r_G)}. \tag{1}$$

These group-based advantages provide fine-grained credit assignment among diverse outputs, encouraging the model to prefer more accurate and non-redundant predictions. The GRPO objective is formulated as a clipped policy gradient with KL regularization:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\left[\min\left(\rho_{i,t}\hat{A}_{i,t}, \text{clip}(\rho_{i,t}, 1-\epsilon, 1+\epsilon)\hat{A}_{i,t}\right) - \beta\mathbb{D}_{\text{KL}}\left[\pi_\theta \| \pi_{\text{ref}}\right]\right], \tag{2}$$

where $\rho_{i,t}$ is the importance sampling ratio, and π_{ref} is the model frozen after the SFT stage. The KL penalty ensures training stability by preventing excessive divergence from the reference model.

This framework naturally mitigates both geometry and behavior limitations: 1) rewards can be made *geometry-aware*, such as IoU or L1 distance metrics, directly encouraging accurate spatial alignment beyond token-level correctness; and 2) by allowing *variable-length outputs*, the model can learn to avoid repetition or over-generation. Repetitive or redundant predictions receive lower rewards, leading to more concise and behaviorally aligned responses.

4.2.3. Geometry-aware Rewards

To provide informative feedback on the spatial quality of predictions, we design three geometry-aware reward functions tailored to different tasks: box IoU reward, point-in-mask reward, and point-in-box reward. These reward types reflect the structural correctness of the predicted outputs with respect to ground-truth annotations.

Box IoU Reward. This reward is applied to tasks requiring bounding box predictions, including object detection, grounding, referring, and OCR. The reward encourages both accurate localization and correct object-category alignment.

Given a set of predicted boxes $\hat{B} = \{\hat{b}_1, \ldots, \hat{b}_m\}$ and the ground-truth boxes $B^* = \{b_1^*, \ldots, b_n^*\}$, we perform a ground-truth-guided matching. For each ground-truth box b_j^*, we find the predicted box $\hat{b}_i$ that maximizes the IoU with b_j^*:

$$\text{IoU}(b_j^*, \hat{b}_i) = \max_{\hat{b}_i \in \hat{B}} \text{IoU}(b_j^*, \hat{b}_i). \tag{3}$$

If the category label of $\hat{b}_i$ matches that of b_j^*, we assign the IoU value as the reward r_j for that ground-truth box. Otherwise, $r_j = 0$. Let $R = \{r_1, \ldots, r_n\}$ be the reward set for all GT boxes. We then compute recall and precision as follows:

$$\text{Recall} = \frac{\sum_{j=1}^{n} r_j}{n}, \quad \text{Precision} = \frac{\sum_{j=1}^{n} r_j}{m}, \quad r^{\text{IoU}} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall} + \epsilon}, \tag{4}$$

where ϵ is a small constant to prevent division by zero. This formulation rewards both spatial accuracy and label correctness. It penalizes unmatched or misclassified predictions and balances over- and under-prediction through the F1-style reward signal.

Point-in-Mask Reward. This reward is applied to tasks where the model localizes objects via point predictions, such as pointing-based detection, grounding, and referring. It evaluates whether a predicted point lies within the object mask.

Given a set of ground-truth bounding boxes $B^* = \{b_1^*, \ldots, b_n^*\}$, we apply SAM to extract a binary mask M_j for each ground-truth box b_j^*. Let $\hat{P} = \{\hat{p}_1, \ldots, \hat{p}_m\}$ denote the predicted points, each associated with a category label. For each ground-truth mask M_j, we determine whether there exists a predicted point $\hat{p}_i$ that lies inside M_j:

$$\exists \hat{p}_i \in \hat{P}, \quad \text{s.t.} \quad \hat{p}_i \in M_j. \tag{5}$$

If such a point exists and its associated category label matches that of M_j, we assign a reward of 1 to the corresponding ground-truth instance; otherwise, the reward is 0. Precision, recall, and F1 reward are then computed using the same formulation as in the Box IoU Reward.

Point-in-Box Reward. This reward is specifically designed for the GUI Grounding task, where the model is expected to predict a point indicating the clickable position (e.g., a button) on a graphical user interface. If the predicted point falls within the ground-truth bounding box of the target GUI element, a reward of 1 is assigned; otherwise, the reward is 0. This simple binary reward effectively encourages precise point-level interaction behavior required in GUI scenarios.

4.2.4. Implementation Details

We sample 66K data from the SFT dataset to serve as training data for the GRPO stage. We reuse the same dialogue templates from the SFT phase. The GRPO training is conducted on 8 A100 GPUs for approximately 24 hours. We set the rollout size to 8, the KL penalty coefficient β to 0.01, and use a batch size of 64. All model parameters are updated during this stage.

5. Benchmark Results

This section presents the evaluation of Rex-Omni across multiple visual perception tasks, such as common, long-tailed, and dense object detection, referring object detection, and object pointing. For each task, we outline the benchmark datasets, experimental settings, and evaluation metrics.

5.1. Common Object Detection

Common object detection refers to the task of detecting objects from a predefined set of categories that frequently appear in real-world scenarios. The goal of this task is to evaluate the model's baisc ability to accurately identify and localize these common objects.

Benchmark: We conduct our evaluation on the COCO [54] dataset, one of the most widely used benchmarks in the field of object detection. The dataset includes 5,000 test images and spans 80 distinct object categories, representing a broad range of common objects.

Type	Method	Zero-Shot	Score Thresh.	COCO								
				mAP	R@IoU 0.5	P@IoU 0.5	F1@IoU 0.5	R@IoU 0.95	P@IoU 0.95	F1@IoU 0.95	R@ mIoU	P@ mIoU	F1@ mIoU
Closed-set	Faster RCNN-R50	No	0.42	38.4	60.4	60.7	60.6	4.9	12.6	7.1	43.2	54.2	48.1
	DETR-R50	No	0.78	41.5	59.6	73.9	65.9	10.6	19.0	13.6	42.9	55.3	48.3
	DyHead-R50	No	0.24	45.9	58.1	76.6	66.1	11.9	20.6	15.0	44.8	60.1	51.3
	DAB-DETR-R50	No	0.31	44.4	59.2	77.4	67.1	10.5	18.5	13.4	43.8	58.8	50.2
	Deformable-DETR-R50	No	0.34	49.4	62.6	78.5	69.7	14.8	17.7	17.7	48.7	62.4	54.7
	DINO-R50	No	0.30	51.7	62.6	76.5	68.8	17.8	25.8	21.1	50.0	62.4	55.6
	DINO-Swin-L	No	0.32	59.6	**69.8**	**82.5**	**75.6**	**22.2**	**29.7**	**25.4**	**57.0**	**68.2**	**62.1**
Open-set	Grounding DINO-Swin-T	Yes	0.37	51.5	62.8	78.4	69.8	20.5	26.2	23.0	54.6	58.8	56.6
MLLM	DeepSeek-VL2-Tiny	UNK	-	-	29.4	55.0	38.2	5.9	13.0	8.1	20.0	38.6	26.3
	OVIS2.5-9B	UNK	-	-	60.0	45.3	51.6	8.4	8.0	8.2	39.3	31.0	34.6
	Mimo-VL-7B	UNK	-	-	56.2	56.9	56.5	6.1	7.4	6.7	35.3	36.4	35.9
	OVIS2.5-2B	UNK	-	-	63.1	50.7	56.2	10.4	10.1	10.3	42.6	35.4	38.7
	DeepSeek-VL2-Small	UNK	-	-	52.1	73.3	60.9	12.5	**18.5**	14.9	39.1	55.4	45.9
	Qwen2.5-VL-7B	UNK	-	-	55.4	75.8	64.0	10.6	15.5	12.6	39.5	54.2	45.7
	Qwen2.5-VL-3B	UNK	-	-	55.7	77.2	64.7	12.7	18.3	15.0	41.0	56.8	47.6
	SEED1.5-VL	Yes	-	-	65.3	**78.6**	71.3	12.7	16.4	14.3	46.8	**56.9**	51.4
	Rex-Omni-SFT	Yes	-	-	66.4	70.1	68.2	**14.8**	17.0	15.8	48.7	52.2	50.4
	Rex-Omni	Yes	-	-	**68.1**	76.3	**72.0**	14.5	17.5	**15.9**	**49.8**	56.5	**52.9**

Table 2: Evaluation results on the COCO benchmark for common object detection. Rex-Omni-SFT refers to the model after the first-stage supervised fine-tuning (SFT), while Rex-Omni is the final model after the second-stage GRPO-based reinforcement post-training. "UNK" signifies that the information was not reported in the respective papers.

Evaluation Settings: We evaluate two variants of our proposed model: **Rex-Omni-SFT**, which undergoes only the first stage of supervised fine-tuning, and the full **Rex-Omni** model, which undergoes both SFT and the subsequent GRPO-based reinforcement post-training. We compare these variants with three types of models: 1) Closed-set detection models trained on COCO, including Faster RCNN [87], DETR [8], DyHead [17], DAB-DETR [58], Deformable-DETR [134] and DINO [122]; 2) Open-set detection model Grounding DINO [59] that is not trained on COCO, and 3) Multimodal large language models (MLLMs) including DeepSeek-VL2 [110], Ovis2.5 [65], MiMo-VL [100], Qwen2.5-VL [4] and SEED1.5-VL [24]. For closed-set detection models, we input images and retain only the predicted bounding boxes whose categories match the ground-truth (GT) labels in each image. For open-set models, we provide all GT categories as text prompts and keep the corresponding results. For MLLMs, we adopt two prompting strategies: (1) querying one GT category at a time (e.g., "Detect dog in this image"), and (2) querying all GT categories simultaneously (e.g., "Detect dog, cat, person in this image"). Although the latter is more practical in real-world scenarios, most MLLMs exhibit a performance drop when handling multiple categories simultaneously. Therefore, except for SEED1.5-VL and Rex-Omni, we use the single-category strategy. All evaluations of Rex-Omni (both SFT and full versions) are conducted with a sampling temperature of 0 to minimize randomness.

Metric: In object detection, the standard metric is Average Precision (AP), which relies on confidence scores to compute precision and recall at varying thresholds. However, multimodal models often lack reliable confidence estimation, rendering AP unsuitable. We therefore adopt Recall, Precision, and F1 score as evaluation metrics. Given predicted and ground-truth boxes, Recall and Precision are computed per category and then averaged, while F1 is taken as their harmonic mean. Following COCO conventions, Intersection over Union (IoU) is evaluated at thresholds from 0.5 to 0.95 (step size 0.05), and results are reported at IoU=0.5, IoU=0.95, and the mean across thresholds. For fair comparison with MLLMs, we further compute F1 scores across confidence thresholds ranging from 0 to 1 (step size 0.01) for both closed- and open-set detection models, reporting the highest F1 as the final performance.

Results: The results are presented in Table 2. Firstly, among MLLMs, Rex-Omni surpasses existing approaches, including SEED1.5-VL, which previously held state-of-the-art detection performance.

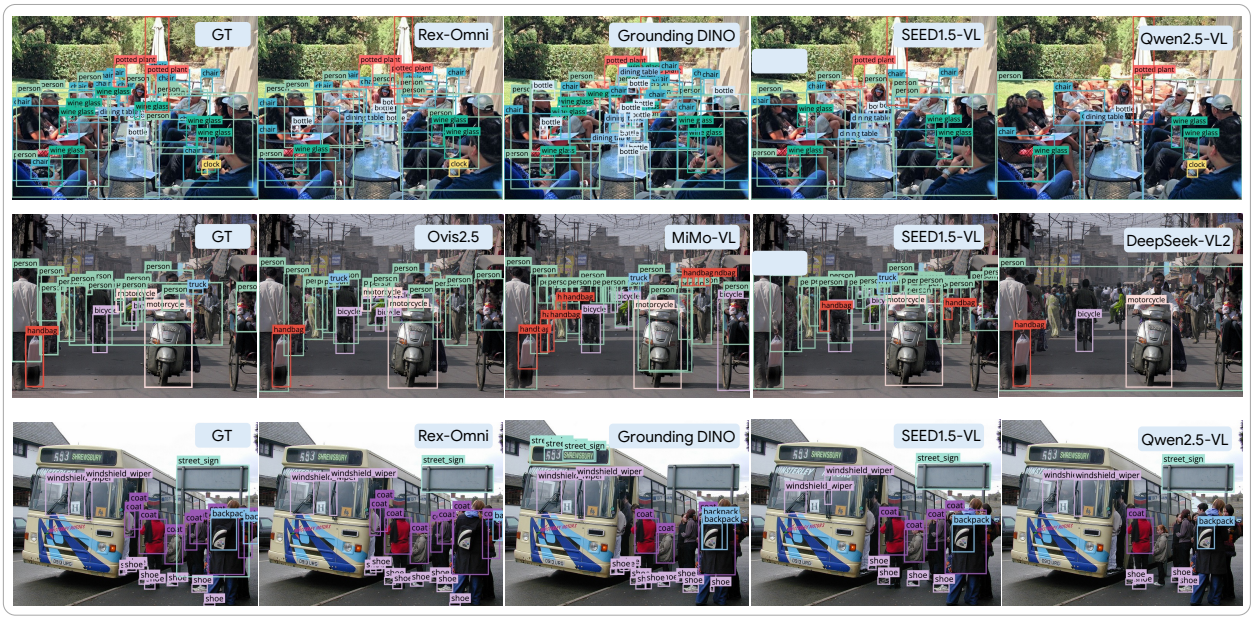


Figure 7: Visualization of detection predictions from different models on common and long-tailed object detection benchmarks, using COCO and LVIS, respectively.

At an IoU threshold of 0.5, Rex-Omni demonstrates superior performance, outperforming both the open-set detection model Grounding DINO-SwinT and the closed-set detection model DINO-R50. Crucially, Rex-Omni achieves this in a zero-shot setting (without training on COCO data), thereby indicating that MLLM-based detection methods can indeed surpass traditional regression-based models when highly precise bounding box localization is not the sole critical factor. However, at a stricter IoU threshold of 0.95, Rex-Omni's performance, while still strong, only marginally outperforms DAB-DETR, suggesting that MLLMs may still lag behind conventional regression-based models in scenarios demanding extremely precise bounding box tightness.

Nevertheless, despite this nuanced limitation, the achieved performance is generally sufficient for a wide range of practical applications. We show some visualization results in Figure 7. Furthermore, a significant improvement is observed with GRPO post-training, where the full Rex-Omni model substantially outperforms its SFT-only variant (Rex-Omni-SFT). This clearly highlights the effectiveness of our reinforcement learning strategy.

5.2. Long-tailed Object Detection

Long-tailed object detection tackles the challenge of recognizing categories with highly imbalanced instance distributions, where most categories are sparsely represented. This task requires models to generalize effectively and robustly detect rare objects in complex real-world scenarios.

Benchmark: We evaluate on the widely used LVIS [25] dataset. LVIS comprises 1,203 categories, significantly more than COCO's 80, and features 19,626 test images. Its categories are derived from WordNet synsets and are intentionally distributed to mimic real-world frequencies, resulting in a natural long-tail distribution where many categories have very few instances.

Evaluation Settings and Metrics: We assess the performance of open-set detection models and MLLMs, following the same evaluation settings and metrics as described in Section 5.1 for COCO.

Results: The results are presented in Table 3. On LVIS, MLLMs generally outperform traditional

Type	Method	Zero-Shot	Score Thresh.	LVIS								
				R@IoU 0.5	P@IoU 0.5	F1@IoU 0.5	R@IoU 0.95	P@IoU 0.95	F1@IoU 0.95	R@ mIoU	P@ mIoU	F1@ mIoU
Open-set	Grounding DINO-Swin-T	Yes	0.21	39.1	61.2	47.7	16.9	**34.8**	**22.7**	31.9	49.6	38.8
MLLM	DeepSeek-VL2-Tiny	UNK	-	22.4	55.7	32.0	7.2	25.7	11.2	14.2	41.9	21.2
	MiMo-VL-7B	UNK	-	43.3	57.8	49.5	6.5	13.7	8.8	25.4	41.1	31.4
	OVIS2.5-2B	UNK	-	53.2	55.6	54.4	13.3	19.4	15.8	34.0	41.6	37.4
	OVIS2.5-9B	UNK	-	52.2	54.1	53.1	11.2	19.9	14.4	32.2	40.1	35.8
	Qwen2.5-VL-7B	UNK	-	47.1	74.3	57.7	12.7	29.0	17.6	32.0	54.3	40.2
	Qwen2.5-VL-3B	UNK	-	44.8	73.9	55.8	14.3	29.8	19.3	31.9	54.6	40.3
	DeepSeek-VL2-Small	UNK	-	46.1	72.2	56.2	15.8	31.2	21.0	33.7	55.0	41.8
	SEED1.5-VL	Yes	-	54.7	**82.0**	**65.6**	15.0	28.1	19.5	38.5	**59.3**	46.7
	Rex-Omni-SFT	Yes	-	52.0	71.6	60.3	**16.6**	27.4	20.7	37.7	53.3	44.2
	Rex-Omni	Yes	-	**54.7**	78.1	64.3	16.5	27.6	20.7	**39.6**	57.5	**46.9**

Table 3: Performance evaluation on the LVIS dataset for long-tailed object detection. All reported metrics adhere to the same methodology described for COCO evaluation in Section 5.1.

open-set detectors such as Grounding DINO, owing to the stronger linguistic reasoning ability of their LLM components compared to conventional text encoders (e.g., CLIP or BERT). This advantage facilitates better generalization to low-frequency categories.

In the zero-shot setting, Rex-Omni achieves competitive performance, with an F1 score at IoU=0.5 second only to SEED1.5-VL, likely due to the latter's larger model size and stronger language understanding. Notably, Rex-Omni attains state-of-the-art results on the mIoU metric, reflecting its superior bounding box precision across thresholds. Moreover, the substantial improvement from Rex-Omni-SFT to the full Rex-Omni model underscores the effectiveness of GRPO-based reinforcement post-training in enhancing object localization. Qualitative results are shown in Figure 7 and Figure 19.

5.3. Dense and Tiny Object Detection

Dense and tiny object detection is crucial for applications such as remote sensing and object counting, requiring accurate localization of numerous small objects in crowded scenes. For MLLMs, this task is particularly challenging: it not only demands precise, extended coordinate predictions sensitive to subtle pixel variations, but also exposes the absence of multi-scale feature mechanisms (e.g., feature pyramids [55]) that traditional regression-based detectors exploit to handle scale diversity. As a result, MLLMs often suffer from issues such as duplicate predictions and coordinate offsets in dense and tiny object detection scenarios.

Benchmark, settings and metrics: We evaluate open-set detection models and MLLMs on two distinct datasets tailored for dense and tiny object detection. The first dataset, VisDrone [19], comprises 1,610 aerial traffic images spanning 10 categories, with individual boxes measuring approximately 30.7×32.4 pixels. Additionally, we introduce Dense200, a manually collected dataset consisting of 200 densely annotated images covering 109 categories. In Dense200, each image contains an average of 91.2 bounding boxes, with an average size of 66.8×64.5 pixels. Together, these datasets pose significant challenges due to the combination of small object sizes and high object density, demanding precise spatial reasoning and accurate localization. The evaluation settings and metrics are the same as those used in Section 5.1 for COCO evaluation.

Results: The results are reported in Table 4, with representative visualizations in Figure 8 and Figure 20. As anticipated in Section 4.2.1, MLLMs struggle with dense and tiny object detection, with most models showing poor performance. We identify two critical failure modes: (1) **Large-box prediction**, where a single oversized bounding box erroneously covers multiple adjacent objects, and (2) **Structured duplicate predictions**, where repeated coordinates with minimal offsets are

Type	Method	Score Threshhold	Dense200			VisDrone		
			F1@IoU 0.5	F1@IoU 0.95	F1@IoU mIoU	F1@IoU 0.5	F1@IoU 0.95	F1@IoU mIoU
Open-set	Grounding DINO-Swin-T	0.25	36.9	**19.7**	33.1	55.2	**3.9**	**38.5**
MLLM	DeepSeek-VL2-Tiny	-	2.2	0.3	1.5	4.3	0.1	1.8
	OVIS2.5-9B	-	14.0	0.0	5.1	15.8	0.1	6.5
	OVIS2.5-2B	-	17.9	0.0	6.7	21.0	0.1	9.2
	MiMo-VL-7B	-	29.7	0.4	15.9	27.7	0.3	14.3
	Qwen2.5-VL-3B	-	0.8	0.1	0.5	31.5	1.9	20.4
	Qwen2.5-VL-7B	-	1.1	0.1	0.6	34.5	1.6	21.7
	DeepSeek-VL2-Small	-	16.0	3.9	12.7	35.8	1.7	23.3
	SEED1.5-VL	-	76.9	5.3	53.2	55.9	0.6	27.4
	Rex-Omni-SFT	-	60.2	10.6	46.4	55.6	1.9	32.4
	Rex-Omni	-	**78.4**	10.3	**58.3**	**61.6**	1.5	35.8

Table 4: Evaluation Results on dense object detection benchmark VisDrone and Dense200. We report the same metric used in COCO evaluation at Section 5.1.

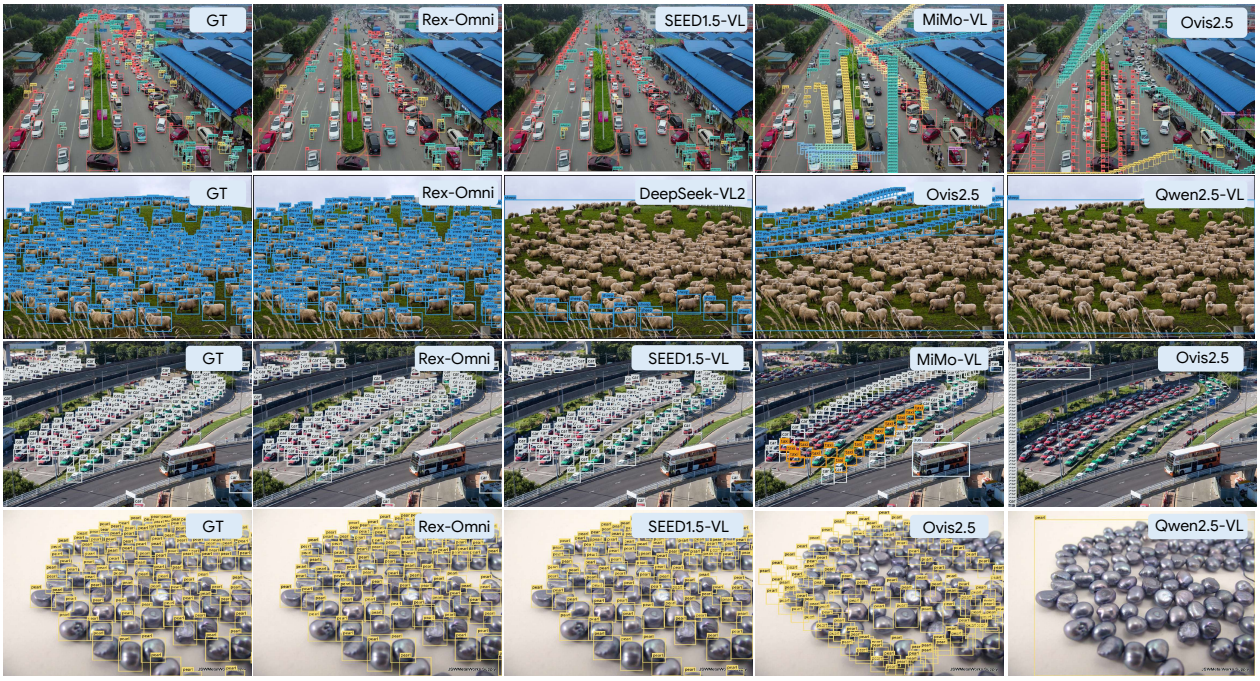


Figure 8: Visualization of dense and tiny object detection predictions. This figure presents a qualitative comparison of various models on the VisDrone and Dense200 datasets.

generated instead of distinct object boxes.

We attribute these issues to the SFT stage. Teacher-forced training on full ground-truth sequences restricts the model's ability to regulate its own output structure. Without such guidance at inference, the model fails to decide object counts or avoid redundant predictions. Notably, we also observed these problematic repetitive predictions in our SFT-only variant. Crucially, after GRPO-based reinforcement post-training, these duplication issues largely disappear, compellingly demonstrating the effectiveness of our two-stage pipeline in correcting SFT-induced deficiencies and enabling more coherent, accurate predictions in dense and tiny object scenarios.

Type	Method	Score Thresh.	HumanRef			RefCOCOg val			RefCOCOg test		
			F1@IoU 0.5	F1@IoU 0.95	F1@IoU mIoU	F1@IoU 0.5	F1@IoU 0.95	F1@IoU mIoU	F1@IoU 0.5	F1@IoU 0.95	F1@IoU mIoU
Open-set	Grounding DINO-Swin-T	0.25	28.0	16.5	25.2	52.9	20.9	45.9	53.8	22.9	46.8
MLLM	DeepSeek-VL2-Tiny	-	39.1	16.9	31.4	67.4	16.1	50.5	69.3	16.9	52.1
	OVIS2.5-2B	-	70.6	12.3	50.0	87.4	29.3	73.4	87.6	30.5	73.8
	OVIS2.5-9B	-	73.1	12.4	52.8	88.8	23.5	72.1	**88.7**	24.2	72.6
	Qwen2.5-VL-3B	-	66.7	46.8	60.5	83.5	30.7	69.2	83.8	31.8	70.1
	MiMo-VL-7B	-	77.6	26.4	63.4	84.9	14.4	65.3	84.6	14.9	65.5
	Qwen2.5-VL-7B	-	72.9	42.9	64.1	86.2	27.2	70.0	85.7	28.4	70.4
	DeepSeek-VL2-Small	-	72.0	46.5	64.7	**92.4**	**45.6**	**81.4**	**91.8**	**47.0**	**81.6**
	SEED1.5-VL	-	**88.2**	60.0	**81.6**	84.7	30.9	71.9	85.2	32.1	73.2
	Rex-Omni-SFT	-	83.3	64.3	77.9	84.9	34.2	71.7	85.2	35.2	72.4
	Rex-Omni	-	85.4	**65.4**	79.9	86.6	35.3	73.6	86.8	36.6	74.3

Table 5: Evaluation results on referring expression comprehension benchmarks, including HumanRef and RefCOCOg.

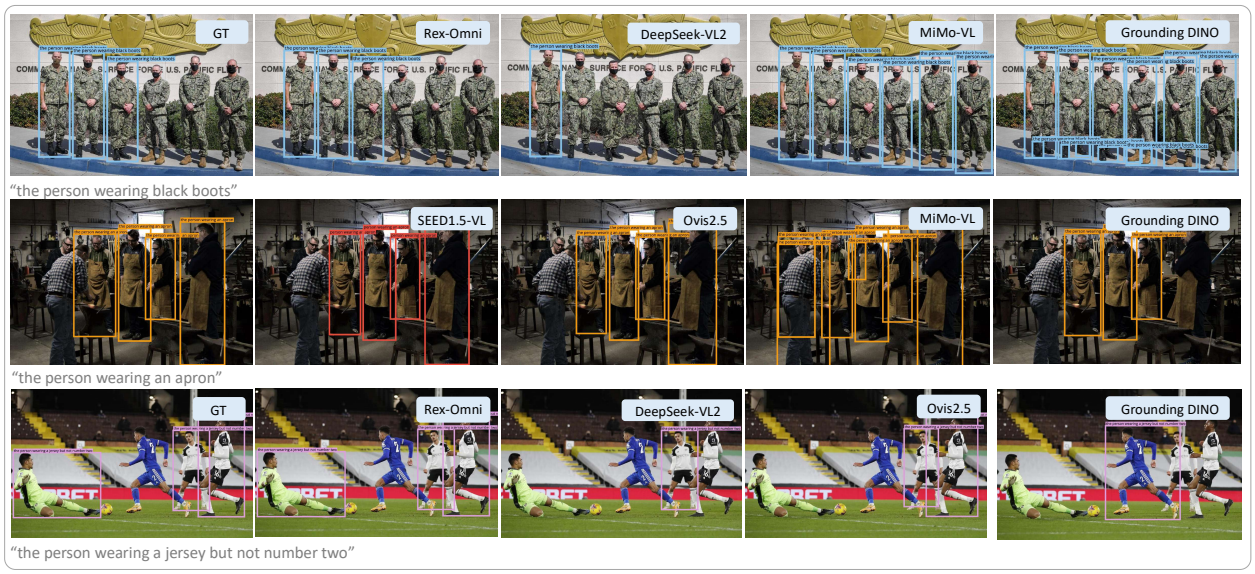


Figure 9: Visualization of model predictions on referring object detection benchmarks.

5.4. Referring Object Detection

Referring object detection requires a model to identify and localize objects described by a natural language expression. Unlike standard object detection, which focuses on category-level recognition, this task demands fine-grained language understanding and strong alignment between linguistic descriptions and visual content.

Benchmark: The evaluation is conducted on two established public benchmarks: **1) RefCOCOg (val/test):** RefCOCOg [70] is built on COCO images and includes 4,889 validation and 9,577 test referring expressions. Each expression maps to a single ground-truth bounding box, making this benchmark relatively straightforward for evaluation. **2) HumanRef:** HumanRef [33] is a human-annotated benchmark focused on people, comprising 6,000 test expressions organized into six subsets: attribute, position, interaction, reasoning, celebrity, and rejection. We use the first five subsets (5,000 images) for evaluation. Unlike RefCOCOg, a single expression in HumanRef may correspond to multiple ground-truth boxes, averaging two per expression. This design poses greater challenges, requiring both fine-grained language understanding and robust visual perception.

Evaluation Settings and Metrics: We evaluate open-set detection models and MLLMs using the

Type	Method	FSC147-test				Dense200				COCO				LVIS			
		F1@ 0.5	F1@ 0.95	F1@ mIoU	MAE	F1@ 0.5	F1@ 0.95	F1@ mIoU	MAE	F1@ 0.5	F1@ 0.95	F1@ mIoU	MAE	F1@ 0.5	F1@ 0.95	F1@ mIoU	MAE
Counting	BMNet+ [94]	-	-	-	14.6	-	-	-	-	-	-	-	-	-	-	-	-
	CountTR [57]	-	-	-	12.0	-	-	-	-	-	-	-	-	-	-	-	-
	DAVE [75]	-	-	-	8.7	-	-	-	-	-	-	-	-	-	-	-	-
Open-set	T-Rex2 [29]	**91.5**	**47.5**	**73.3**	10.9	**93.9**	**67.1**	**88.1**	**6.5**	72.3	**19.9**	57.8	**4.0**	71.1	**34.6**	**58.8**	**3.4**
MLLM	Rex-Omni-SFT	87.0	10.5	64.6	7.8	65.1	11.1	50.0	50.9	73.6	19.8	58.4	11.7	70.2	26.3	56.0	11.7
	Rex-Omni	86.0	10.5	62.8	**7.0**	79.5	10.8	59.2	20.1	**79.1**	19.4	**61.3**	4.5	**74.4**	24.8	57.0	5.2

Table 6: Evaluation results for the visual prompting task across COCO, LVIS, Dense200, and FSC147 datasets. Performance is assessed using F1-score for detection and Mean Absolute Error (MAE) for object counting.

same settings and metrics as in Section 5.1 for COCO, with one exception: during testing, models are queried with a single referring expression at a time. For the open-set detector Grounding DINO, we adopt its official demo confidence threshold of 0.25

Results: The results are presented in Table 5. Open-set detection models notably struggle with this task, as evidenced by Grounding DINO's consistent underperformance across benchmarks. In stark contrast, MLLMs, leveraging their inherent strong language understanding capabilities, consistently excel at this task. On HumanRef, Rex-Omni achieves competitive results, ranking second only to SEED1.5-VL. This indicates that, while Rex-Omni (3B parameters) possesses sufficient language understanding for effective REC, larger models like SEED1.5-VL benefit from greater capacity for more nuanced reasoning. Overall, Rex-Omni's strong performance across all datasets demonstrates its ability to align natural language with visual content, making it highly practical for real-world referring scenarios. Visualization examples are shown in Figure 9 and and Figure 21.

5.5. Visual Prompting

While text prompts are widely used in many tasks, they have inherent limitations, particularly when certain objects are difficult to describe using language alone. In such scenarios, visual prompting can offer a more effective approach for object detection. In this section, we define visual prompting as a task where, given an image alongside several example bounding boxes within it, the model is required to detect all other objects belonging to the same category as those indicated by the examples.

Benchmark and Evaluation Setting: We evaluate visual prompting on the object counting dataset FSC147 [83], as well as on object detection benchmarks COCO, LVIS, and Dense200. The FSC147 dataset consists of 1,190 images, each containing a dense set of objects from a single category along with three example bounding boxes, which are used as visual prompts for detection. For COCO, LVIS, and Dense200, we follow the T-Rex2 [29] methodology, where for each ground-truth category in an image, one bounding box is randomly sampled as the visual prompt for that category. To interface with Rex-Omni, the coordinates of the selected visual prompt box are converted into special tokens and embedded into the query, e.g., "Given reference boxes <12><52><212><337> indicating one or more objects, find all objects of the same category in the image."

Metric: We primarily adopt the F1-score as described in Section 5.1 for object detection. Additionally, we introduce the Mean Absolute Error (MAE) metric to evaluate the model's object counting ability. MAE is computed as the absolute difference between predicted and ground-truth object counts, averaged across the entire dataset, thereby providing an additional measure of the model's capability to accurately count objects in dense scenes.

Results: While Rex-Omni's overall performance still falls short of the traditional expert model T-Rex2, it demonstrates strong visual prompting capabilities. In particular, Rex-Omni performs well in

Method	COCO	LVIS	Dense200	VisDrone	HumanRef	RefCOCOg val	RefCOCOg test
	F1@Point	F1@Point	F1@Point	F1@Point	F1@Point	F1@Point	F1@Point
OVIS2.5-2B	73.4	52.8	36.4	23.8	72.5	83.1	83.1
Qwen2.5-VL-3B	65.9	48.3	4.3	13.9	64.1	77.4	77.8
Qwen2.5-VL-7B	61.1	56.5	2.0	14.2	65.1	78.9	79.4
OVIS2.5-9B	72.6	61.7	35.0	18.8	62.3	**85.0**	84.5
Molmo-7B-D	77.3	40.3	33.1	29.2	70.0	83.7	83.6
SEED1.5-VL	78.2	70.7	72.1	56.7	83.1	83.6	84.2
Rex-Omni-SFT	76.0	66.7	72.9	49.5	82.1	83.3	83.9
Rex-Omni	**80.5**	**70.8**	**82.5**	**58.9**	**83.8**	84.7	**85.1**

Table 7: Performance evaluation for the object pointing task across a diverse range of benchmarks (COCO, LVIS, Dense200, VisDrone, RefCOCOg, HumanRef). F1-scores are used as the primary metric.

both dense scenes and long-tailed scenarios, highlighting its effectiveness in addressing high object density and severe class imbalance. Representative visualization results are shown in Figure 10.

Figure 10: Qualitative comparison of visual prompting predictions between T-Rex2 and Rex-Omni.

5.6. Object Pointing

The object pointing task requires models to predict precise point coordinates for specified target objects. Unlike bounding boxes, point annotations offer greater flexibility in localization, as models can indicate an object's center or any representative position within its boundaries.

Benchmarks, Evaluation Settings and Metric: To evaluate object pointing, we integrate datasets previously used for box-based detection, including COCO, LVIS, Dense200, VisDrone, RefCOCOg, and HumanRef. This collection covers a broad range of visual scenarios, from common and long-tailed objects to dense small objects and complex referring expressions. The evaluation protocol follows that of the earlier detection tasks. For most models, except SEED1.5-VL and Rex-Omni, each test image is queried with a single ground-truth category at a time. We adopt the same F1-score–based evaluation metric as in object detection, with one modification in the matching criterion. For each ground-truth bounding box, we generate a segmentation mask using SAM, and a predicted point is considered correct if it falls within the corresponding mask. Recall, Precision, and F1 are then computed analogously to standard box-based evaluation.

Results: The performance of all evaluated models is reported in Table 7. While most MLLMs achieve reasonable pointing accuracy on common object categories, they struggle with dense or

Figure 11: Qualitative comparison of object pointing predictions from different models.

Method	ScreenSpot-V2							ScreenSpot-Pro												
	Mobile		Desktop		Web		Avg	Dev.		Creative		CAD		Sci.		Office		OS		Avg
	Text	Icon	Text	Icon	Text	Icon		Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	
UI-TARS-2B	95.2	79.1	90.7	68.6	87.2	78.3	84.7	47.4	4.1	42.9	6.3	17.8	4.7	56.9	17.3	50.3	17.0	21.5	5.6	27.7
Qwen2.5-VL-3B	93.4	73.5	88.1	58.6	88.0	71.4	80.9	38.3	3.4	40.9	4.9	22.3	6.3	44.4	10.0	48.0	17.0	33.6	4.5	25.9
UI-R1-3B	84.3	**96.2**	75.4	**89.2**	63.6	**92.3**	85.4	22.7	4.1	27.3	3.5	11.2	6.3	42.4	11.8	32.2	11.3	13.1	4.5	17.8
InfiGUI-R1-3B	-	-	-	-	-	-	-	51.3	12.4	44.9	7.0	33.0	**14.1**	58.3	20.0	**65.5**	28.3	**43.9**	12.4	35.7
SE-GUI-3B	-	-	-	-	-	-	-	55.8	7.6	47.0	4.9	**38.1**	12.5	**61.8**	16.4	59.9	24.5	40.2	12.4	35.9
JEDI-3B	**96.6**	81.5	96.9	78.6	**88.5**	83.7	**88.6**	61.0	**13.8**	**53.5**	8.4	27.4	9.4	54.2	18.2	64.4	**32.1**	38.3	9.0	36.1
Rex-Omni-SFT	95.5	80.6	**97.4**	77.1	85.5	76.4	86.4	54.6	10.3	46.5	10.5	22.3	7.8	55.6	19.1	55.9	20.8	37.4	11.2	32.6
Rex-Omin-GRPO	93.3	84.3	96.4	84.3	86.8	80.5	88.4	**61.7**	9.7	52.5	**12.6**	22.3	9.4	59.0	**26.4**	63.3	28.3	24.1	**15.7**	**36.8**

Table 8: Evaluation results for GUI Grounding task on the ScreenSpot-V2 and ScreenSpot Pro datasets.

small-scale instances, particularly on Dense200 and VisDrone. Rex-Omni attains the highest F1-scores across both general and challenging datasets, highlighting its strong spatial localization ability. Representative visualizations are shown in Figure 11 and and Figure 22.

5.7. GUI Grounding

Graphical User Interface (GUI) grounding evaluates a model's ability to localize specific UI elements based on natural language queries. This task is critical for applications such as intelligent agents, automated UI interaction, and software testing, as it requires seamless integration of visual perception and language understanding.

Benchmarks, Evaluation Setting and Metric: We evaluate models on two datasets: ScreenSpot-V2 [109] and ScreenSpot-Pro [48]. ScreenSpot-V2 encompasses mobile, desktop, and web scenarios, featuring a diverse array of UI layouts across 1,272 images. ScreenSpot-Pro, conversely, focuses on ultra-high-resolution interfaces, specifically designed to test the model's precision in localizing UI elements under highly challenging visual conditions, comprising 1,581 images. Rex-Omni is assessed using its point-based prediction capability, outputting a point within the target UI element for each query. Following standard protocols, we report accuracy, considering a prediction correct if the point

Type	Method	Score Thresh.	DocLayNet			M6Doc		
			F1@IoU 0.5	F1@IoU 0.95	F1@IoU mIoU	F1@IoU 0.5	F1@IoU 0.95	F1@IoU mIoU
Closed-Set	DocLayout-YOLO [128]	0.3	**91.2**	**52.1**	**81.1**	-	-	-
MLLM	Qwen2.5-VL-3B	-	17.5	2.9	9.1	13.3	2.5	8.4
	Qwen2.5-VL-7B	-	25.6	5.1	13.4	24.0	4.1	15.0
	SEED1.5-VL	-	54.9	4.3	28.7	48.0	3.4	28.0
	Rex-Omni-SFT	-	85.9	27.2	70.7	74.5	16.2	54.2
	Rex-Omni		89.5	28.4	70.7	**76.3**	**18.7**	**55.6**

Table 9: Performance comparison of different models on the DocLayNet and M6Doc datasets for layout grounding. F1-scores (at IoU=0.5, IoU=0.95, mIoU) are reported.

falls within the ground-truth bounding box.

Results: As shown in Table 8, Rex-Omni consistently demonstrates strong performance on GUI grounding tasks. Specifically, among 3B-parameter models, Rex-Omni achieves the highest accuracy across both ScreenSpot V2 and ScreenSpot Pro. This underscores its superior capability to seamlessly integrate robust language understanding with fine-grained visual localization, even in diverse and ultra-high-resolution UI scenarios.

5.8. Layout Grounding

Layout grounding requires models to localize and interpret the spatial relationships among elements in a document, such as titles, paragraphs, sections, and figures. This task is crucial for applications like document layout analysis and web page understanding, as it demands not only object detection but also reasoning over structural arrangement and semantic relationships.

Benchmark, Evaluation Setting, and Metrics: We evaluate our model on the DocLayNet [78] and M6Doc [12] datasets. DocLayNet is collected from PDF documents and includes 11 categories such as footnotes, pictures, tables, and titles, with a test set consisting of 6,480 images.The M6Doc dataset is considerably more complex, encompassing data from diverse domains (e.g., scientific articles, textbooks, test papers, magazines, newspapers, notes, books) with a total of 74 categories across 2,724 test images. For evaluation, we treat this task as an object detection problem, following the same evaluation protocol used for common object detection on COCO.

Results: The results are presented in Table 9. Rex-Omni outperforms other MLLMs by a large margin on layout grounding. While there remains a performance gap compared to closed-set models, Rex-Omni's ability to handle open-set layout grounding provides a unique advantage. Unlike closed-set models, which are limited to predefined categories, Rex-Omni demonstrates a unique capability to generalize to unseen domains and novel layout structures, establishing it as a more versatile and adaptable solution for real-world layout understanding tasks. Representative visualization results are provided in Figure 12 and Figure 23.

5.9. OCR

Optical Character Recognition (OCR) involves both text detection and recognition, where the model identifies and extracts text from images or documents. The task requires the model to detect text regions and then recognize the characters or words within those regions, enabling the conversion of scanned documents or images into machine-readable text.

Benchmark, Evaluation Setting: We evaluate the performance of PaddleOCR, SEED1.5-VL, and

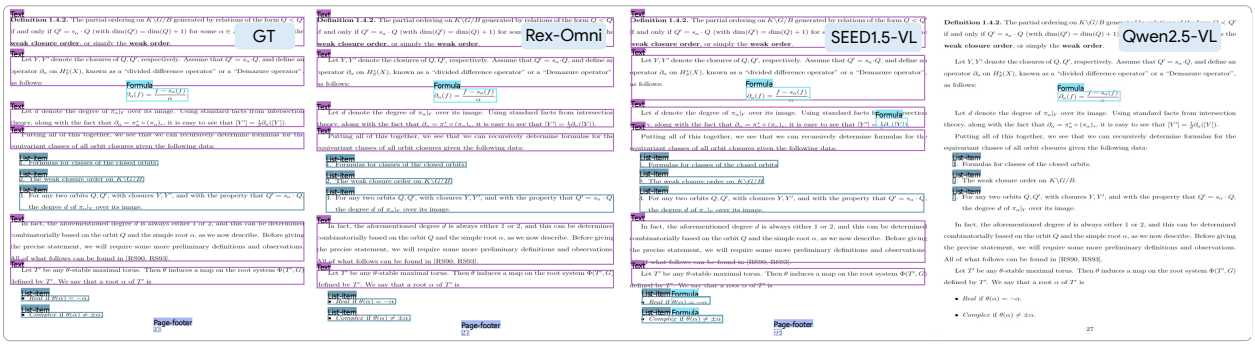


Figure 12: Qualitative comparison of layout grounding predictions from different models. The figure illustrates the models' ability to localize and interpret various layout elements.

Output Format	Method	HierText			ICDAR2015			TotalText			SROIE		
		F1@IoU 0.5	F1@IoU 0.95	F1@IoU mIoU	F1@IoU 0.5	F1@IoU 0.95	F1@IoU mIoU	F1@IoU 0.5	F1@IoU 0.95	F1@IoU mIoU	F1@IoU 0.5	F1@IoU 0.95	F1@IoU mIoU
BBOX	PaddleOCRv5 [16]	45.2	**3.4**	**30.5**	38.2	**1.2**	25.6	40.2	0.7	25.7	**77.7**	**5.6**	**58.6**
	SEED1.5-VL	27.1	0.2	12.0	38.6	0.0	18.7	35.0	0.3	19.5	51.9	0.8	28.1
	Rex-Omni-SFT	23.5	0.5	13.7	31.4	0.1	18.7	38.1	1.5	25.0	46.5	0.9	28.6
	Rex-Omni	**45.9**	1.4	28.0	**45.2**	0.3	**28.1**	**56.6**	**3.9**	**40.6**	72.0	1.5	44.8
POLY	PaddleOCRv5	41.5	1.1	**26.3**	36.4	0.2	23.3	34.1	0.0	18.4	70.5	**2.3**	**50.2**
	Rex-Omni-SFT	**43.2**	0.3	26.2	43.2	**0.3**	26.2	50.3	0.1	**25.7**	**73.8**	0.2	39.7
	Rex-Omni	40.2	0.1	20.2	**50.7**	0.0	**28.5**	**52.8**	**0.1**	25.6	60.3	0.0	19.2

Table 10: Performance comparison of various models on the OCR task, evaluated using F1 score for text detection and recognition accuracy.

Rex-Omni on four diverse datasets. The core evaluation method involves detecting and recognizing all text within the images. The datasets include HierText (3,446 instances, primarily dense text), TotalText (600 instances, scene text with predominantly curved text), ICDAR2015 (1,000 instances, scene text), and SROIE (720 instances, printed receipt data with mostly horizontal text). Together, these datasets cover a broad spectrum of OCR challenges, from dense and curved scene text to structured document text. For PaddleOCR and Rex-Omni, both bounding box (BBOX) and polygonal (POLY) text regions are predicted, and we report performance for both formats to provide a comprehensive assessment of text localization.

Metrics: We formulate OCR as an object detection task, following the COCO evaluation protocol with categories replaced by recognized text. A prediction is considered correct if (1) the predicted and ground-truth regions match, and (2) the recognized text exactly matches the ground truth. Performance is reported using the F1 score, balancing precision and recall.

Results: The evaluation results for the OCR task are presented in Table 10. For bounding box (BBOX) outputs, Rex-Omni demonstrates strong competitive performance. It significantly outperforms SEED1.5-VL across all metrics and datasets, and achieves comparable or superior results to the dedicated OCR expert model PaddleOCRv5 in several key aspects. This highlights Rex-Omni's robust capabilities in text detection and recognition using bounding boxes. In the polygonal (POLY) output format, Rex-Omni also shows competitive performance. The full Rex-Omni model, after GRPO post-training, notably achieves leading results on challenging datasets like ICDAR2015 for polygonal text region detection. This indicates the versatility of our approach in handling more complex text geometries. Consistent gains from Rex-Omni-SFT to Rex-Omni further validate the effectiveness of our two-stage training pipeline in enhancing OCR performance. Representative visualization results are provided in Figure 13 and and Figure 24.

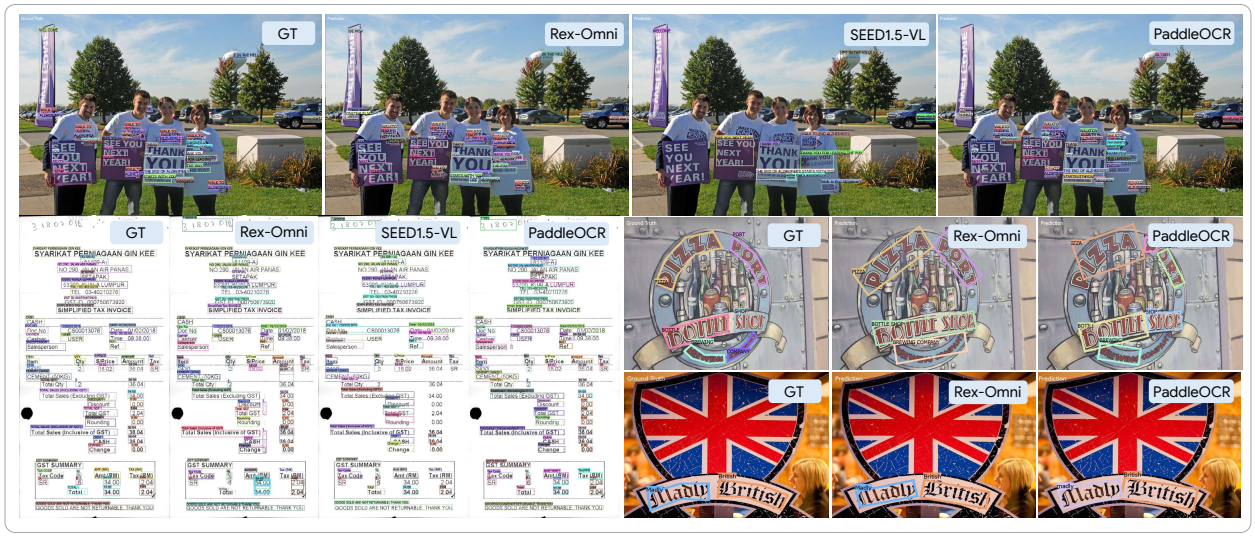


Figure 13: Visualization of OCR results across models.

Multi Subtask	*Proprietary Models*	*Referring Specialist Models*					*Our Models*	
	Gemini-2.5-Pro [15]	SpaceLLAVA [21]	RoboPoint [119]	Molmo-7B [18]	Molmo-72B [18]	ReboRefer-2B* [131]	Rex-Omini-SFT	Rex-Omini
Location	46.96	5.82	22.87	21.91	45.77	51.00	**55.00**	54.00
Placement	24.21	4.31	9.27	12.85	14.74	49.00	45.00	**50.00**
Useen	27.14	4.02	8.40	12.23	21.24	**38.96**	36.36	36.36

Table 11: Comparison of different models on the RefSpatial benchmark. The value is the percentage (%) of correct predictions. *: evaluated without depth prior.

5.10. Spatial Pointing

This task focuses on grounding natural language expressions that describe spatial relationships in complex scenes. Unlike standard referring object detection, which primarily matches objects to category names or simple attributes, spatial grounding requires models to interpret relational cues such as relative position, anchoring, and free-space placement.

Benchmark and Metrics: RefSpatial-Bench [131] evaluates spatial referring and reasoning in complex indoor scenes across two tasks: *location* and *placement*, each with 100 curated samples. Each sample includes an image, a natural language referring expression, and precise mask annotations. The location task requires models to predict a 2D point corresponding to a target object based on referring expressions that may involve attributes such as color, shape, spatial order, or anchor-based references. The placement task requires identifying a suitable 2D point within a free space described in the expression, often involving multiple anchors or hierarchical spatial relations. To assess generalization, the benchmark additionally provides 77 unseen samples containing novel combinations of spatial relations not present in training. Evaluation is performed using ground-truth masks, with accuracy defined as the percentage of predictions falling within the mask region.

Results: As shown in Table 11, Rex-Omni substantially outperforms prior proprietary and referring-specialist models. Its strong performance on both location and placement tasks suggests enhanced applicability to downstream scenarios such as robotic manipulation, where accurate grasping and placement are essential. Moreover, Rex-Omni demonstrates superior generalization to unseen cases, underscoring its robustness in handling novel spatial relations. Representative visualization results are provided in Figure 14.

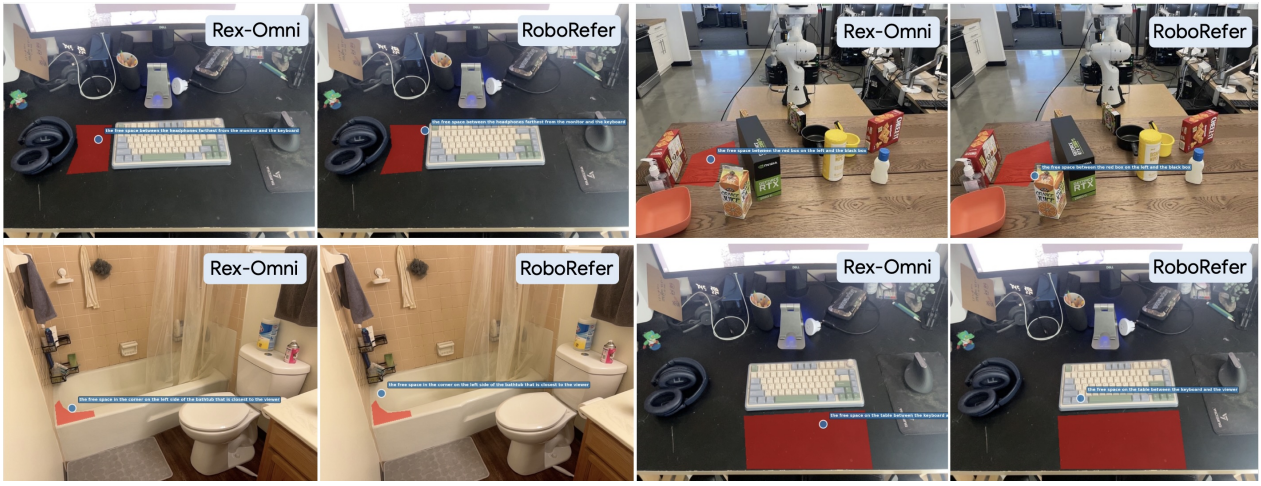


Figure 14: Visualization of Spatial Pointing results across models. Masks indicate correct areas.

Type	Model	Score Thresh.	COCO Keypoint			AP10K Keypoint		
			F1@OKS 0.5	F1@OKS 0.95	F1@OKS mOKS	F1@OKS 0.5	F1@OKS 0.95	F1@OKS mOKS
Open-set	X-Pose	0.3 (COCO) or 0.05 (AP10K)	**66.3**	**39.6**	**57.2**	17.0	2.1	8.7
MLLM	Rex-Omni-SFT	-	39.9	17.5	29.3	27.4	2.2	13.0
	Rex-Omni	-	44.4	17.9	32.6	**30.1**	**3.0**	**14.6**

Table 12: Evaluation results for keypoint estimation on COCO and AP10K datasets.

5.11. Keypoint

Benchmark, Evaluation Setting, and Metrics: COCO is a benchmark dataset designed to evaluate 2D human pose estimation and instance-level keypoint detection capabilities in unconstrained environments. It comprises a large-scale collection of images featuring people in diverse and complex natural scenes. Each annotated person instance includes a set of 17 predefined body joints, forming a standard human skeleton. AP10K is a benchmark designed to advance the field of 2D animal pose estimation, addressing the challenge of anatomical variation across species. The benchmark standardizes keypoint annotation with a unified definition of 17 body keypoints for mammals, reptiles, and birds. Following the COCO protocol, we adopt Object Keypoint Similarity (OKS) as the evaluation metric. We report F1 score at OKS thresholds of 0.5, 0.95, and the mean over thresholds from 0.5 to 0.95 in increments of 0.05.

Results: As shown in Table 12, the open-set expert model X-Pose achieves the strongest performance on COCO keypoint detection, particularly at lower OKS thresholds. However, it generalizes poorly to AP10K, where its performance drops sharply. In contrast, Rex-Omni demonstrates more balanced results across both human and animal keypoint benchmarks. Although its absolute scores on COCO lag behind X-Pose, Rex-Omni substantially outperforms it on AP10K, highlighting its superior cross-domain generalization. Furthermore, the consistent improvement from Rex-Omni-SFT to the full Rex-Omni model validates the effectiveness of our two-stage training pipeline for enhancing keypoint reasoning. Representative visualization results are provided in Figure 15.

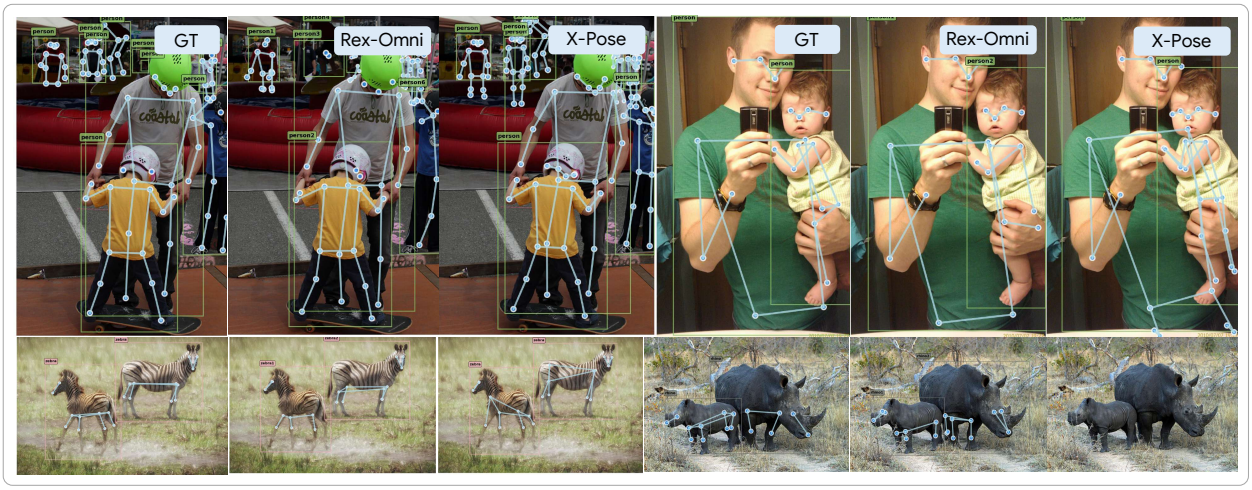


Figure 15: Qualitative comparison of keypoint detection predictions from different models.

6. In-depth Analysis of Rex-Omni

In this section, we conduct a comprehensive analysis to investigate and elucidate the efficacy of Rex-Omni's key design components. Our aim is to provide a deeper understanding of how each architectural choice, training strategy (including the role of GRPO), and data design collectively influences the model's overall performance across various visual perception tasks.

6.1. Why GRPO Works

Rex-Omni adopts a two-stage training strategy, beginning with supervised fine-tuning (SFT) and followed by GRPO-based reinforcement learning. Across all coordinate prediction benchmarks, the GRPO-enhanced model consistently outperforms its SFT-only counterpart. To investigate the source of these gains, we analyze the model's behaviors and highlight key error patterns that GRPO effectively mitigates.

6.1.1. Training Dynamics

To better understand how Rex-Omni acquires its visual perception capability, we analyze the performance trajectory during both the SFT and GRPO stages as training progresses. Figure 16 illustrates model performance on representative benchmarks as a function of training steps (measured by the amount of data seen).

During the SFT stage, performance exhibits a steady and gradual improvement. As the model is exposed to more training data, it progressively learns to align visual inputs with coordinate outputs, leading to consistent but incremental gains across benchmarks. However, once SFT concludes, performance tends to plateau, suggesting that further improvements from additional supervised exposure are limited. In contrast, the GRPO stage produces a strikingly different trajectory. With only a small number of training steps, the model experiences a rapid performance jump across benchmarks. Notably, this improvement cannot be attributed simply to more data exposure, since the GRPO stage involves far fewer samples than SFT. Instead, the results suggest that the SFT-trained model already possesses strong latent capabilities that remain underutilized. GRPO, by introducing behavior-aware rewards and sequence-level feedback, effectively unlocks this hidden potential, enabling the model to achieve a substantial leap in performance with minimal additional training.

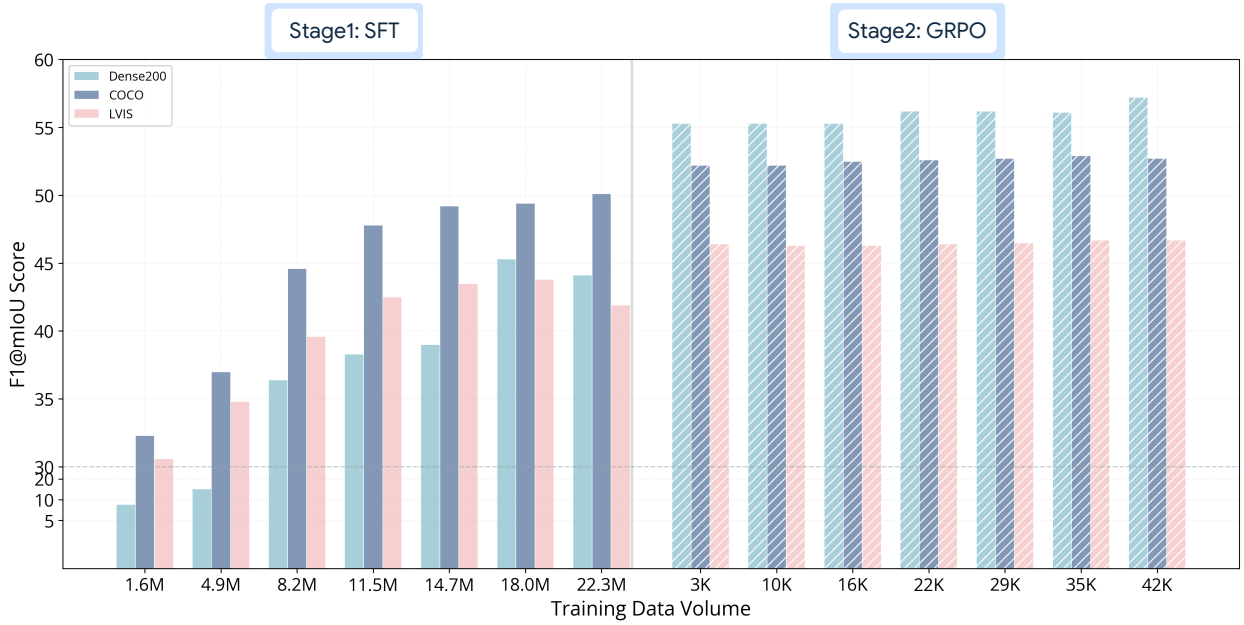


Figure 16: Model performance across SFT and GRPO training stages. F1@mIoU scores are shown for three datasets (Dense200, COCO, LVIS) with training data volume increasing.

Remove Duplicate	COCO				LVIS				VisDrone			
	SFT		GRPO		SFT		GRPO		SFT		GRPO	
	F1@0.5	Remov.	F1@0.5	Remov.	F1@0.5	Remov.	F1@0.5	Remov.	F1@0.5	Remov.	F1@0.5	Remov.
No	68.2	-	72.0	-	60.3	-	64.3	-	55.6	-	61.6	-
Yes	70.1	1.23%	72.6	0.08%	61.3	1.38%	64.7	0.06%	62.3	15.3%	62.1	0.1%

Table 13: Performance comparison (F1@IoU=0.5) of SFT and GRPO models before and after removing duplicate predictions across COCO, LVIS, and VisDrone. This highlights GRPO's effectiveness in mitigating repetitive outputs and its impact on overall performance.

Taken together, these dynamics reveal that GRPO's benefit lies not in extending supervised learning, but in reshaping model behavior to better exploit existing capabilities. In the following subsections, we delve deeper into the specific mechanisms behind this improvement, beginning with how GRPO corrects problematic behaviors learned during SFT.

6.1.2. Behavioral Correction via GRPO

Duplicate Predictions. A major error pattern is the tendency to generate repeated predictions. Under SFT, the model is trained with full teacher forcing, conditioning on ground-truth tokens at each step—so it rarely encounters or corrects this issue. In contrast, GRPO requires the model to generate sequences autonomously and provides reward-based feedback. Repeated coordinates receive low rewards, effectively discouraging duplication and promoting more coherent predictions.

To verify this effect, we analyzed predictions from both the SFT-only and GRPO-trained models, focusing on repeated outputs. A repeated prediction is defined as a coordinate sequence where the same value appeared consecutively at least 10 times, with the total number of predicted boxes exceeding twice the ground-truth count. We removed such duplicates and re-evaluated F1 scores. As shown in Table 13, the SFT-only model exhibited substantial improvements after duplicate removal (e.g., +1.23% on COCO, +1.38% on LVIS, and +15.3% on VisDrone), whereas the GRPO model showed minimal gains (e.g., +0.08% on COCO, +0.1% on VisDrone). This indicates that SFT-trained

Remove Large box	Dense200							
	SFT				GRPO			
	F1@IoU 0.5	F1@IoU 0.95	F1@IoU mIoU	Remov.	F1@IoU 0.5	F1@IoU 0.95	F1@IoU mIoU	Remov.
No	59.1	8.8	44.9	-	78.4	10.3	58.3	-
Yes	74.6	11.2	56.7	20.5%	81.8	9.4	60.0	3.5%

Table 14: Impact of large box prediction removal on F1-score (IoU=0.5, IoU=0.95, mIoU) for SFT and GRPO models on the Dense200 dataset. The "Remov." column indicates the percentage of large box predictions removed from the total outputs.

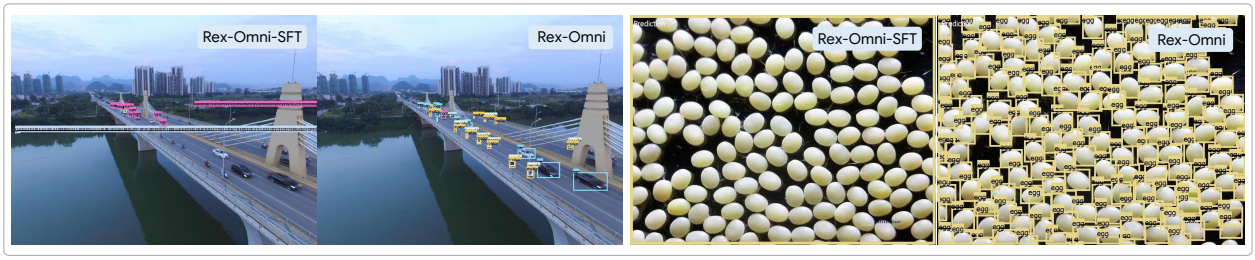


Figure 17: Illustration of challenges in MLLM-based object detection and their mitigation. The left side qualitatively compares predictions from SFT and GRPO models, highlighting GRPO's effectiveness in reducing duplicate outputs. The right side visualizes the large-box prediction failure mode, where models inappropriately predict overly large bounding boxes encompassing multiple objects.

models produce significantly more repeated predictions than GRPO-trained models. After removing duplicates, the performance gap between SFT and GRPO narrowed, becoming nearly negligible on dense datasets like VisDrone. Visual examples of these differences are shown in Figure 17 (left). These findings confirm that GRPO effectively suppresses duplicate predictions, which is a key factor in Rex-Omni's overall performance improvement.

Large-box Predictions. Another behavioral issue observed, especially in dense object detection scenarios, is the tendency of models to predict a single large bounding box that encompasses multiple dense objects. This failure mode was also highlighted in our benchmarking of dense object detection (Section 5.3). To investigate this, we conducted an experiment on the Dense200 dataset. A large box prediction was defined as a scenario where only one bounding box was predicted in the image, and its area exceeded 95% of the total image size. We then analyzed instances of such large box predictions from both the SFT-only and GRPO-trained models, removing these samples from evaluation.

As shown in Table 14, the SFT-only model exhibited a substantial 20.5% of its total predictions as large boxes, leading to a significant performance improvement (e.g., F1@IoU=mIoU increased from 44.9 to 56.7) once these large box predictions were removed. In stark contrast, the GRPO-trained model had only 3.5% of its predictions categorized as large boxes, and consequently, showed a much smaller performance change (F1@IoU=mIoU increased from 58.3 to 60.0) upon their removal. This clearly indicates that GRPO's behavior-aware optimization effectively discourages models from producing such overly large, encompassing bounding boxes for dense scenes. This failure mode is visually exemplified on the right side of Figure 17.

6.1.3. Improvement in Coordinate Precision?

We hypothesize that the cross-entropy loss used in SFT lacks geometry-awareness, whereas GRPO can exploit geometry-aware rewards to refine coordinate precision. To validate this, we evaluate coordinate precision on COCO, LVIS, and HumanRef.

Stage	COCO			LVIS			HumanRef		
	F1@IoU 0.5	F1@IoU 0.95	F1@IoU mIoU	F1@IoU 0.5	F1@IoU 0.95	F1@IoU mIoU	F1@IoU 0.5	F1@IoU 0.95	F1@IoU mIoU
SFT	80.5	23.5	63.0	74.2	33.0	56.6	85.2	67.2	60.0
GRPO	81.1	23.3	63.5	75.0	32.9	56.9	86.4	68.0	61.2

Table 15: Impact of GRPO on coordinate precision. The table reports F1-scores (at IoU=0.5, IoU=0.95, and mIoU) for SFT and GRPO models across COCO, LVIS and HumanRef datasets, specifically for instances where both models achieve consistent ground-truth matching. This analysis highlights GRPO's modest contribution to refining coordinate precision.

Method	COCO			LVIS			Dense200		
	F1@IoU 0.5	F1@IoU 0.95	F1@IoU mIoU	F1@IoU 0.5	F1@IoU 0.95	F1@IoU mIoU	F1@IoU 0.5	F1@IoU 0.95	F1@IoU mIoU
SFT	68.2	15.8	50.4	60.3	20.7	44.2	60.2	10.6	46.4
GRPO	72.0	15.9	52.9	64.3	20.7	46.9	78.4	10.3	58.3
SFT-Sampling-Best	64.6	9.0	44.0	56.6	13.8	38.7	38.2	2.0	24.6
SFT-Sampling-Vote	72.6	16.8	54.0	59.8	14.8	41.3	50.6	3.9	34.7

Table 16: Impact of GRPO on the likelihood of sampling correct predictions. This table compares the F1-scores (IoU=0.5, IoU=0.95, mIoU) of SFT, GRPO, SFT-Sampling-Best, and SFT-Sampling-Vote models across COCO, LVIS, and Dense200 datasets. It illustrates GRPO's role in enhancing the probability and inherent quality of correct outputs, and explores SFT's potential under various sampling strategies.

Specifically, for each test sample, We only include samples where both the SFT and GRPO models produce a number of predicted boxes that exactly matches the ground-truth count. Furthermore, for these selected samples, each predicted box from both models must achieve an IoU exceeding a predefined matching threshold with its corresponding ground-truth box. This filtering strategy allows us to effectively isolate the analysis to focus exclusively on the subtle differences in coordinate precision. As shown in Table 15, GRPO yields only modest gains over SFT. For example, F1@mIoU increases slightly from 63.0 to 63.5 on COCO and from 56.6 to 56.9 on LVIS. These results suggest that SFT already provides sufficient capacity for learning accurate coordinates and tight localization. Thus, GRPO's primary advantage lies not in boosting raw coordinate precision, but in correcting behavioral deficiencies such as duplicate predictions and large-box outputs, as discussed earlier.

6.1.4. Elevating the Likelihood of Correct Predictions

Beyond behavioral correction and coordinate refinement, we examine GRPO's impact from a sampling-probability perspective. We hypothesize that SFT models inherently possess the ability to generate accurate predictions, but their inference randomness reduces the likelihood of consistently sampling optimal outputs. GRPO, by contrast, leverages reward-guided exploration to increase this likelihood.

To empirically test this, we conducted high-temperature sampling experiments using the SFT model on COCO, LVIS, and Dense200. We simulated GRPO's rollout by sampling 8 candidate predictions per test instance (using temperature 1.2, top-k 50, top-p 0.99). From these, we derived two SFT-based metrics: **SFT-Sampling-Best**: The highest F1-score achieved across 8 independent full-dataset test runs of the SFT model. **SFT-Sampling-Vote**: For each test sample, the best prediction (highest F1-score against ground truth) is chosen from its 8 sampled outputs. These sample-wise best predictions are then aggregated for overall performance. This estimates SFT's maximal performance if optimal predictions were reliably selected at the sample level.

As shown in Table 16, on COCO the SFT-Sampling-Vote score (72.6 F1@0.5) exceeds both GRPO (72.0) and base SFT (68.2), indicating that SFT has a latent capacity for accurate predictions and

Model	COCO@100			Dense200@100		
	boxes/img (Avg.)	output tokens/img (Avg.)	tokens/box (Avg.)	boxes/img (Avg.)	output tokens/img (Avg.)	tokens/box (Avg.)
SEED1.5-VL	4.2	631.0	148.8	73.1	5446.3	74.5
Rex-Omni	5.9	45.3	7.6	86.7	439.0	5.1

Table 17: Comparison of output tokenization efficiency between SEED1.5-VL and Rex-Omni. This table reports the average number of boxes per image, output tokens per image, and tokens per box on 100 randomly sampled images from COCO and Dense200, highlighting Rex-Omni's superior token efficiency.

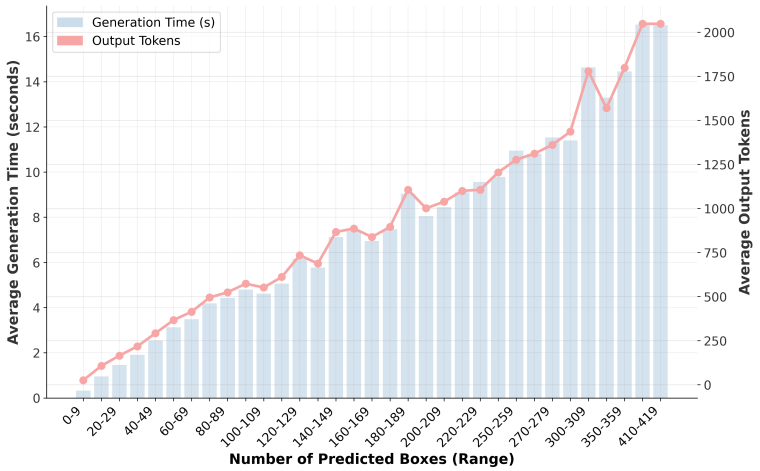


Figure 18: Inference speed and output token analysis. The plot shows the average generation time (seconds) and the average number of output tokens as a function of the number of predicted boxes. The experiment was conducted on a single NVIDIA A100 GPU with vLLM deployment in BF16 precision, without model acceleration or compression.

GRPO mainly improves sampling consistency on simpler datasets. However, on LVIS and Dense200, neither SFT-Sampling-Best nor SFT-Sampling-Vote approaches GRPO's performance, showing that for complex tasks GRPO plays a deeper role by enabling inherently more coherent and precise predictions. These findings suggest that GRPO's benefits vary by task complexity: increasing sampling probability on simpler settings, and fundamentally enhancing prediction quality in more challenging ones.

6.2. Inference Efficiency and Speed

The efficiency of coordinate representation is crucial, as it directly affects output length and inference speed. We compare Rex-Omni, which encodes quantized coordinates using special tokens, with SEED1.5-VL, which represents relative coordinates without special tokens. To evaluate this, we sampled 100 images each from COCO and Dense200, and measured the average boxes per image, total output tokens per image, and tokens per box. As summarized in Table 17, Rex-Omni achieves far greater tokenization efficiency. For example, on COCO it requires only 7.6 tokens per box on average, compared to 148.8 for SEED1.5-VL, with total output length reduced from 631.0 to 45.3 tokens per image. Similar improvements are observed on Dense200, confirming that dedicated special tokens substantially enhance efficiency, especially in dense object settings.

Beyond tokenization efficiency, we further examine practical inference speed. Figure 18 illustrates the relationship between the number of predicted boxes, output token length, and average generation time, measured on a single NVIDIA A100 GPU using vLLM with BF16 precision (no acceleration or compression applied). Both generation time and token count grow approximately linearly with

the number of predicted boxes: detecting a few objects (0–29) takes under 2 seconds, whereas detecting hundreds of objects (e.g., 410–419) exceeds 16 seconds. These findings indicate that current MLLM-based detectors are slower than traditional optimized detectors, with speed scaling directly with the number of detected objects. Nevertheless, this limitation could be mitigated through acceleration strategies such as quantization or distillation.

7. Conclusion

In this work, we have introduced Rex-Omni, a 3B-parameter MLLM that systematically addresses the challenges of MLLM-based object detection. Through efficient coordinate tokenization with special tokens, large-scale data generation via custom engines, and a novel SFT+GRPO two-stage training pipeline, we bridge the gap between precise localization and deep language understanding. Our extensive experiments demonstrate that Rex-Omni achieves state-of-the-art or highly competitive zero-shot performance across a wide array of visual perception tasks. Crucially, our analysis validates that while SFT provides a strong foundation, GRPO-based post-training is essential for correcting SFT-induced behavioral deficiencies, such as duplicate and large-box predictions, a key contribution towards robust MLLM-based detectors. Despite its strong performance, limitations such as inference speed remain. We believe that future work in model acceleration and advanced reward-guided sampling will be critical next steps. In summary, Rex-Omni represents a significant step forward, demonstrating that the behavioral and geometric limitations of MLLMs can be systematically overcome, thereby paving the way for the next generation of versatile, language-aware perception systems.

8. Related Work

Regresson-based Object Detection Methods. Object detection has long been a cornerstone task in computer vision, with regression-based methods historically dominating the field. The core principle of these methods is to predict a bounding box by regressing its properties typically including center coordinates (x, y) and dimensions (width, height) as normalized offsets from a predefined reference. Over the years, these methods have undergone significant evolution, progressing from early anchor-based CNN models like YOLO [86], SSD [60], and Faster R-CNN [87], to anchor-free approaches such as CornerNet [43], CenterNet [20], and FCOS [102]. A major paradigm shift occurred with the introduction of Transformer-based detectors like DETR [8], which framed object detection as a direct set prediction problem. This line of work was further advanced by models such as Deformable DETR [134] and DINO [122], which significantly improved performance and convergence speed. Beyond these paradigmatic changes, the continuous improvement of regression-based detectors has been fueled by numerous incremental yet crucial innovations. These include architectural enhancements like Feature Pyramid Networks (FPN)[55], advancements in loss functions like Focal Loss[56], and sophisticated data augmentation techniques such as MixUp [124] and Mosaic. It is the cumulative effect of these extensive and persistent efforts that has propelled regression-based object detectors to their current state of high performance and practical usability.

Open-set Object Detection Methods. A long-term objective of object detection is to develop models capable of identifying an arbitrary number of object categories without task-specific fine-tuning, thereby addressing the challenges of real-world, dynamic scenarios. Open-set object detection represents a significant paradigm shift towards this goal, transcending the limitations of closed-set detection by empowering models to identify objects beyond a predefined set of categories. The prevalent approach to this challenge is text-prompted open-vocabulary object detection [49, 59, 35, 115, 130, 88, 22, 13, 68]. These methods typically leverage powerful pre-trained vision-language models like CLIP [80] or BERT [37] to align textual descriptions with visual represen-

tations, demonstrating impressive zero-shot recognition capabilities. However, these models struggle with complex or nuanced descriptions due to their limited language understanding. To overcome this, visual prompts [32, 28, 88, 103, 84, 39, 135, 45] have been introduced, allowing models to recognize objects using visual examples like boxes or points. Visual prompts are effective for rare or hard-to-describe objects but are less general than text prompts. Recent models like T-Rex2 [32] combine both text and visual prompts, using contrastive learning to leverage the strengths of each. This integration allows models to perform well across a wider range of object categories and real-world scenarios. While traditional open-set detectors achieve category-level generalization, they still lack deeper language understanding, making it challenging to handle context-rich real-world scenarios.

MLLM-based Object Detection Methods. To overcome the shallow language understanding of traditional open-set detectors, a promising direction is to directly leverage the powerful reasoning capabilities of Multimodal Large Language Models (MLLMs) for object-level perception. The core idea is to reframe object detection as a language modeling task. Inspired by Pix2Seq [10], a significant body of work has emerged that represents bounding box coordinates as a sequence of discrete, quantized tokens [76, 9, 116, 106, 120]. These models, including Kosmos-2, Shikra, Ferret, and CogVLM, directly generate coordinate sequences through the standard next-token prediction mechanism of LLMs. This approach elegantly unifies object detection with the native capabilities of language models. However, as discussed in our introduction, this conceptually elegant approach faces significant practical challenges. While MLLMs excel at high-level image understanding, they often struggle with the fine-grained spatial precision required for object detection. Existing methods frequently suffer from limitations such as low recall rates, coordinate drift, and spurious duplicate predictions. We posit that these issues stem from two fundamental challenges: the inherent difficulty of learning a precise mapping from discrete tokens to a continuous pixel space using cross-entropy loss, and the behavioral deficiencies induced by the teacher-guided nature of Supervised Fine-Tuning (SFT). Addressing these challenges is the primary motivation for the design of Rex-Omni.

References

[1] Pravesh Agrawal, Szymon Antoniak, Emma Bou Hanna, Devendra Chaplot, Jessica Chudnovsky, Saurabh Garg, Theophile Gervet, Soham Ghosh, Amélie Héliou, Paul Jacob, et al. Pixtral 12b. *arXiv preprint arXiv:2410.07073*, 2024.

[2] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pp. 3686–3693, 2014.

[3] Chongyang Bai, Xiaoxue Zang, Ying Xu, Srinivas Sunkara, Abhinav Rastogi, Jindong Chen, et al. Uibert: Learning generic multimodal representations for ui understanding. *arXiv preprint arXiv:2107.13731*, 2021.

[4] 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.

[5] Ali Furkan Biten, Ruben Tito, Lluis Gomez, Ernest Valveny, and Dimosthenis Karatzas. Ocr-idl: Ocr annotations for industry document library dataset. In *European Conference on Computer Vision*, pp. 241–252. Springer, 2022.

[6] Minwoo Byeon, Beomhee Park, Haecheon Kim, Sungjun Lee, Woonhyuk Baek, and Saehoon Kim. Coyo-700m: Image-text pair dataset. https://github.com/kakaobrain/coyo-dataset, 2022.

[7] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11621–11631, 2020.

[8] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pp. 213–229. Springer, 2020.

[9] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023.

[10] Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection. *arXiv preprint arXiv:2109.10852*, 2021.

[11] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. In *ICLR*, 2022.

[12] Hiuyi Cheng, Peirong Zhang, Sihang Wu, Jiaxin Zhang, Qiyuan Zhu, Zecheng Xie, Jing Li, Kai Ding, and Lianwen Jin. M6doc: A large-scale multi-format, multi-type, multi-layout, multi-language, multi-annotation category dataset for modern document layout analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15138–15147, 2023.

[13] Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-world: Real-time open-vocabulary object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16901–16911, 2024.

[14] Chee Kheng Chng, Yuliang Liu, Yipeng Sun, Chun Chet Ng, Canjie Luo, Zihan Ni, ChuanMing Fang, Shuaitao Zhang, Junyu Han, Errui Ding, et al. Icdar2019 robust reading challenge on arbitrary-shaped text-rrc-art. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1571–1576. IEEE, 2019.

[15] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

[16] Cheng Cui, Ting Sun, Manhui Lin, Tingquan Gao, Yubo Zhang, Jiaxuan Liu, Xueqing Wang, Zelun Zhang, Changda Zhou, Hongen Liu, et al. Paddleocr 3.0 technical report. *arXiv preprint arXiv:2507.05595*, 2025.

[17] Xiyang Dai, Yinpeng Chen, Bin Xiao, Dongdong Chen, Mengchen Liu, Lu Yuan, and Lei Zhang. Dynamic head: Unifying object detection heads with attentions. In *CVPR*, pp. 7373–7382, 2021.

[18] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. *arXiv preprint arXiv:2409.17146*, 2024.

[19] Dawei Du, Pengfei Zhu, Longyin Wen, Xiao Bian, Haibin Lin, Qinghua Hu, Tao Peng, Jiayu Zheng, Xinyao Wang, Yue Zhang, et al. Visdrone-det2019: The vision meets drone object detection in image challenge results. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*, pp. 0–0, 2019.

[20] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6569–6578, 2019.

[21] Matthew Foutter, Daniele Gammelli, Justin Kruger, Ethan Foss, Praneet Bhoj, Tommaso Guffanti, Simone D'Amico, and Marco Pavone. Space-llava: a vision-language model adapted to extraterrestrial applications. *arXiv preprint arXiv:2408.05924*, 2024.

[22] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *European conference on computer vision*, pp. 540–557. Springer, 2022.

[23] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

[24] Dong Guo, Faming Wu, Feida Zhu, Fuxing Leng, Guang Shi, Haobin Chen, Haoqi Fan, Jian Wang, Jianyu Jiang, Jiawei Wang, et al. Seed1. 5-vl technical report. *arXiv preprint arXiv:2505.07062*, 2025.

[25] Agrim Gupta, Piotr Dollar, and Ross Girshick. 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.

[26] Ju He, Shuo Yang, Shaokang Yang, Adam Kortylewski, Xiaoding Yuan, Jie-Neng Chen, Shuai Liu, Cheng Yang, Qihang Yu, and Alan Yuille. Partimagenet: A large, high-quality dataset of parts. In *European Conference on Computer Vision*, pp. 128–145. Springer, 2022.

[27] Zheng Huang, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shijian Lu, and CV Jawahar. Icdar 2019 robust reading challenge on scanned receipts ocr and information extraction. In *International conference on document analysis recognition*, 2019.

[28] Qing Jiang, Feng Li, Tianhe Ren, Shilong Liu, Zhaoyang Zeng, Kent Yu, and Lei Zhang. T-rex: Counting by visual prompting. *arXiv preprint arXiv:2311.13596*, 2023.

[29] 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 *European Conference on Computer Vision*, pp. 38–57. Springer, 2024.

[30] Qing Jiang, Yuqin Yang, Yuda Xiong, Yihao Chen, Zhaoyang Zeng, Tianhe Ren, Lei Zhang, et al. Chatrex: Taming multimodal llm for joint perception and understanding. *arXiv preprint arXiv:2411.18363*, 2024.

[31] Qing Jiang, Xingyu Chen, Zhaoyang Zeng, Junzhi Yu, and Lei Zhang. Rex-thinker: Grounded object referring via chain-of-thought reasoning. *arXiv preprint arXiv:2506.04034*, 2025.

[32] 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 *European Conference on Computer Vision*, pp. 38–57. Springer, 2025.

[33] Qing Jiang, Lin Wu, Zhaoyang Zeng, Tianhe Ren, Yuda Xiong, Yihao Chen, Qin Liu, and Lei Zhang. Referring to any person, 2025. URL https://arxiv.org/abs/2503.08507.

[34] Xuan Ju, Ailing Zeng, Jianan Wang, Qiang Xu, and Lei Zhang. Human-art: A versatile human-centric dataset bridging natural and artificial scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 618–629, 2023.

[35] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. MDETR - modulated detection for end-to-end multi-modal understanding. In *ICCV*, pp. 1760–1770, 2021.

[36] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In *2015 13th international conference on document analysis and recognition (ICDAR)*, pp. 1156–1160. IEEE, 2015.

[37] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pp. 4171–4186, 2019.

[38] Rahima Khanam and Muhammad Hussain. Yolov11: An overview of the key architectural enhancements. *arXiv preprint arXiv:2410.17725*, 2024.

[39] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloé Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross B. Girshick. Segment anything. *arXiv: 2304.02643*, 2023.

[40] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper R. R. Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, and Vittorio Ferrari. The open images dataset V4: unified image classification, object detection, and visual relationship detection at scale. *arXiv: 1811.00982*, 2018.

[41] Rollyn Labuguen, Jumpei Matsumoto, Salvador Blanco Negrete, Hiroshi Nishimaru, Hisao Nishijo, Masahiko Takada, Yasuhiro Go, Ken-ichi Inoue, and Tomohiro Shibata. Macaquepose: a novel "in the wild" macaque monkey pose dataset for markerless motion capture. *Frontiers in behavioral neuroscience*, 14:581154, 2021.

[42] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9579–9589, 2024.

[43] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 734–750, 2018.

[44] Dongxu Li, Yudong Liu, Haoning Wu, Yue Wang, Zhiqi Shen, Bowen Qu, Xinyao Niu, Guoyin Wang, Bei Chen, and Junnan Li. Aria: An open multimodal native mixture-of-experts model. *arXiv preprint arXiv:2410.05993*, 2024.

[45] Feng Li, Hao Zhang, Peize Sun, Xueyan Zou, Shilong Liu, Chunyuan Li, Jianwei Yang, Lei Zhang, and Jianfeng Gao. Segment and recognize anything at any granularity. In *European Conference on Computer Vision*, pp. 467–484. Springer, 2024.

[46] Hang Li. Cdla: A chinese document layout analysis (cdla) dataset, 2021.

[47] Jiefeng Li, Can Wang, Hao Zhu, Yihuan Mao, Hao-Shu Fang, and Cewu Lu. Crowdpose: Efficient crowded scenes pose estimation and a new benchmark. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10863–10872, 2019.

[48] Kaixin Li, Ziyang Meng, Hongzhan Lin, Ziyang Luo, Yuchen Tian, Jing Ma, Zhiyong Huang, and Tat-Seng Chua. Screenspot-pro: Gui grounding for professional high-resolution computer use. *arXiv preprint arXiv:2504.07981*, 2025.

[49] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10965–10975, 2022.

[50] Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, Ming Zhou, and Zhoujun Li. Tablebank: Table benchmark for image-based table detection and recognition. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 1918–1925, 2020.

[51] Xiaojie Li, Lu Yang, Qing Song, and Fuqiang Zhou. Detector-in-detector: Multi-level analysis for human-parts. In *Asian Conference on Computer Vision*, pp. 228–240. Springer, 2018.

[52] Kevin Qinghong Lin, Linjie Li, Difei Gao, Zhengyuan Yang, Zechen Bai, Weixian Lei, Lijuan Wang, and Mike Zheng Shou. Showui: One vision-language-action model for generalist gui agent. In *NeurIPS 2024 Workshop on Open-World Agents*, volume 1, 2024.

[53] 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*, pp. 740–755, 2014.

[54] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, volume 8693, pp. 740–755, 2014.

[55] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pp. 2117–2125, 2017.

[56] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pp. 2980–2988, 2017.

[57] Chang Liu, Yujie Zhong, Andrew Zisserman, and Weidi Xie. Countr: Transformer-based generalised visual counting, 2023. URL https://arxiv.org/abs/2208.13721.

[58] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. Dab-detr: Dynamic anchor boxes are better queries for detr. *arXiv preprint arXiv:2201.12329*, 2022.

[59] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

[60] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pp. 21–37. Springer, 2016.

[61] Yuqi Liu, Bohao Peng, Zhisheng Zhong, Zihao Yue, Fanbin Lu, Bei Yu, and Jiaya Jia. Seg-zero: Reasoning-chain guided segmentation via cognitive reinforcement. *arXiv preprint arXiv:2503.06520*, 2025.

[62] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1096–1104, 2016.

[63] Shangbang Long, Siyang Qin, Dmitry Panteleev, Alessandro Bissacco, Yasuhisa Fujii, and Michalis Raptis. 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.

[64] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[65] Shiyin Lu, Yang Li, Yu Xia, Yuwei Hu, Shanshan Zhao, Yanqing Ma, Zhichao Wei, Yinglun Li, Lunhao Duan, Jianshan Zhao, et al. Ovis2. 5 technical report. *arXiv preprint arXiv:2508.11737*, 2025.

[66] Simon M Lucas, Alex Panaretos, Luis Sosa, Anthony Tang, Shirley Wong, Robert Young, Kazuki Ashida, Hiroki Nagai, Masayuki Okamoto, Hiroaki Yamamoto, et al. Icdar 2003 robust reading competitions: entries, results, and future directions. *International Journal of Document Analysis and Recognition (IJDAR)*, 7(2):105–122, 2005.

[67] Hongchen Luo, Wei Zhai, Jing Zhang, Yang Cao, and Dacheng Tao. Learning affordance grounding from exocentric images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2252–2261, 2022.

[68] Chuofan Ma, Yi Jiang, Xin Wen, Zehuan Yuan, and Xiaojuan Qi. Codet: Co-occurrence guided region-word alignment for open-vocabulary object detection. *Advances in neural information processing systems*, 36:71078–71094, 2023.

[69] Chuofan Ma, Yi Jiang, Jiannan Wu, Zehuan Yuan, and Xiaojuan Qi. Groma: Localized visual tokenization for grounding multimodal large language models. *arXiv preprint arXiv:2404.13013*, 2024.

[70] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L. Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, pp. 11–20, 2016.

[71] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, et al. Simple open-vocabulary object detection. In *European conference on computer vision*, pp. 728–755. Springer, 2022.

[72] Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. *Advances in Neural Information Processing Systems*, 36:72983–73007, 2023.

[73] Thanh Nguyen, Chau Pham, Khoi Nguyen, and Minh Hoai. Few-shot object counting and detection. In *European Conference on Computer Vision*, pp. 348–365. Springer, 2022.

[74] OpenAI. Gpt-4v(ision) system card. https://cdn.openai.com/papers/GPTV_System_Card.pdf, 2023.

[75] Jer Pelhan, Alan Lukežič, Vitjan Zavrtanik, and Matej Kristan. Dave – a detect-and-verify paradigm for low-shot counting, 2024. URL https://arxiv.org/abs/2404.16622.

[76] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*, 2023.

[77] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*, 2023.

[78] Birgit Pfitzmann, Christoph Auer, Michele Dolfi, Ahmed S Nassar, and Peter Staar. Doclaynet: A large human-annotated dataset for document-layout segmentation. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 3743–3751, 2022.

[79] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[80] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, volume 139, pp. 8748–8763, 2021.

[81] 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*, pp. 8748–8763, 2021.

[82] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7141–7151, 2023.

[83] Viresh Ranjan, Udbhav Sharma, Thu Nguyen, and Minh Hoai. Learning to count everything. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3394–3403, 2021.

[84] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[85] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[86] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 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.

[87] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016.

[88] 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.

[89] Tianhe Ren, Qing Jiang, Shilong Liu, Zhaoyang Zeng, Wenlong Liu, Han Gao, Hongjie Huang, Zhengyu Ma, Xiaoke Jiang, Yihao Chen, et al. Grounding dino 1.5: Advance the" edge" of open-set object detection. *arXiv preprint arXiv:2405.10300*, 2024.

[90] Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun. Crowdhuman: A benchmark for detecting human in a crowd. *arXiv preprint arXiv:1805.00123*, 2018.

[91] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8430–8439, 2019.

[92] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

[93] Baoguang Shi, Cong Yao, Minghui Liao, Mingkun Yang, Pei Xu, Linyan Cui, Serge Belongie, Shijian Lu, and Xiang Bai. Icdar2017 competition on reading chinese text in the wild (rctw-17). In *2017 14th iapr international conference on document analysis and recognition (ICDAR)*, volume 1, pp. 1429–1434. IEEE, 2017.

[94] Min Shi, Hao Lu, Chen Feng, Chengxin Liu, and Zhiguo Cao. Represent, compare, and learn: A similarity-aware framework for class-agnostic counting, 2022. URL https://arxiv.org/abs/2203.08354.

[95] Amanpreet Singh, Guan Pang, Mandy Toh, Jing Huang, Wojciech Galuba, and Tal Hassner. Textocr: Towards large-scale end-to-end reasoning for arbitrary-shaped scene text. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8802–8812, 2021.

[96] Hongbin Sun, Zhanghui Kuang, Xiaoyu Yue, Chenhao Lin, and Wayne Zhang. Spatial dual-modality graph reasoning for key information extraction. *arXiv preprint arXiv:2103.14470*, 2021.

[97] Xian Sun, Peijin Wang, Zhiyuan Yan, Feng Xu, Ruiping Wang, Wenhui Diao, Jin Chen, Jihao Li, Yingchao Feng, Tao Xu, et al. Fair1m: A benchmark dataset for fine-grained object recognition in high-resolution remote sensing imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 184:116–130, 2022.

[98] Yipeng Sun, Jiaming Liu, Wei Liu, Junyu Han, Errui Ding, and Jingtuo Liu. Chinese street view text: Large-scale chinese text reading with partially supervised learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9086–9095, 2019.

[99] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10781–10790, 2020.

[100] Core Team, Zihao Yue, Zhenru Lin, Yifan Song, Weikun Wang, Shuhuai Ren, Shuhao Gu, Shicheng Li, Peidian Li, Liang Zhao, Lei Li, Kainan Bao, Hao Tian, Hailin Zhang, Gang Wang, Dawei Zhu, Cici, Chenhong He, Bowen Ye, Bowen Shen, Zihan Zhang, Zihan Jiang, Zhixian Zheng, Zhichao Song, Zhenbo Luo, Yue Yu, Yudong Wang, Yuanyuan Tian, Yu Tu, Yihan Yan, Yi Huang, Xu Wang, Xinzhe Xu, Xingchen Song, Xing Zhang, Xing Yong, Xin Zhang, Xiangwei

Deng, Wenyu Yang, Wenhan Ma, Weiwei Lv, Weiji Zhuang, Wei Liu, Sirui Deng, Shuo Liu, Shimao Chen, Shihua Yu, Shaohui Liu, Shande Wang, Rui Ma, Qiantong Wang, Peng Wang, Nuo Chen, Menghang Zhu, Kangyang Zhou, Kang Zhou, Kai Fang, Jun Shi, Jinhao Dong, Jiebao Xiao, Jiaming Xu, Huaqiu Liu, Hongshen Xu, Heng Qu, Haochen Zhao, Hanglong Lv, Guoan Wang, Duo Zhang, Dong Zhang, Di Zhang, Chong Ma, Chang Liu, Can Cai, and Bingquan Xia. Mimo-vl technical report, 2025. URL https://arxiv.org/abs/2506.03569.

[101] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv: 2312.11805*, 2023.

[102] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9627–9636, 2019.

[103] Ao Wang, Lihao Liu, Hui Chen, Zijia Lin, Jungong Han, and Guiguang Ding. Yoloe: Real-time seeing anything. *arXiv preprint arXiv:2503.07465*, 2025.

[104] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19844–19854, 2023.

[105] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[106] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*, 2023.

[107] 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.

[108] Jiannan Wu, Muyan Zhong, Sen Xing, Zeqiang Lai, Zhaoyang Liu, Zhe Chen, Wenhai Wang, Xizhou Zhu, Lewei Lu, Tong Lu, et al. Visionllm v2: An end-to-end generalist multimodal large language model for hundreds of vision-language tasks. *Advances in Neural Information Processing Systems*, 37:69925–69975, 2024.

[109] Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, Yian Wang, Qiushi Sun, Chengyou Jia, Kanzhi Cheng, Zichen Ding, Liheng Chen, Paul Pu Liang, et al. Os-atlas: A foundation action model for generalist gui agents. *arXiv preprint arXiv:2410.23218*, 2024.

[110] Zhiyu Wu, Xiaokang Chen, Zizheng Pan, Xingchao Liu, Wen Liu, Damai Dai, Huazuo Gao, Yiyang Ma, Chengyue Wu, Bingxuan Wang, et al. Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding. *arXiv preprint arXiv:2412.10302*, 2024.

[111] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3974–3983, 2018.

[112] Fan Yang, Lei Hu, Xinwu Liu, Shuangping Huang, and Zhenghui Gu. A large-scale dataset for end-to-end table recognition in the wild. *Scientific Data*, 10(1):110, 2023.

[113] Yuxiang Yang, Junjie Yang, Yufei Xu, Jing Zhang, Long Lan, and Dacheng Tao. Apt-36k: A large-scale benchmark for animal pose estimation and tracking. *Advances in Neural Information Processing Systems*, 35:17301–17313, 2022.

[114] Yuxiang Yang, Yingqi Deng, Yufei Xu, and Jing Zhang. Aptv2: benchmarking animal pose estimation and tracking with a large-scale dataset and beyond. *arXiv preprint arXiv:2312.15612*, 2023.

[115] 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. *Advances in Neural Information Processing Systems*, 35:9125–9138, 2022.

[116] Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. Ferret: Refer and ground anything anywhere at any granularity. *arXiv preprint arXiv:2310.07704*, 2023.

[117] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2636–2645, 2020.

[118] Hang Yu, Yufei Xu, Jing Zhang, Wei Zhao, Ziyu Guan, and Dacheng Tao. Ap-10k: A benchmark for animal pose estimation in the wild. *arXiv preprint arXiv:2108.12617*, 2021.

[119] Wentao Yuan, Jiafei Duan, Valts Blukis, Wilbert Pumacay, Ranjay Krishna, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. Robopoint: A vision-language model for spatial affordance prediction for robotics. *arXiv preprint arXiv:2406.10721*, 2024.

[120] Yufei Zhan, Yousong Zhu, Zhiyang Chen, Fan Yang, Ming Tang, and Jinqiao Wang. Griffon: Spelling out all object locations at any granularity with large language models. In *European Conference on Computer Vision*, pp. 405–422. Springer, 2025.

[121] Ao Zhang, Yuan Yao, Wei Ji, Zhiyuan Liu, and Tat-Seng Chua. Next-chat: An lmm for chat, detection and segmentation, 2023. URL https://arxiv.org/abs/2311.04498.

[122] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022.

[123] Haotian Zhang, Haoxuan You, Philipp Dufter, Bowen Zhang, Chen Chen, Hong-You Chen, Tsu-Jui Fu, William Yang Wang, Shih-Fu Chang, Zhe Gan, et al. Ferret-v2: An improved baseline for referring and grounding with large language models. *arXiv preprint arXiv:2404.07973*, 2024.

[124] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[125] Rui Zhang, Yongsheng Zhou, Qianyi Jiang, Qi Song, Nan Li, Kai Zhou, Lei Wang, Dong Wang, Minghui Liao, Mingkun Yang, et al. Icdar 2019 robust reading challenge on reading chinese text on signboard. In *2019 international conference on document analysis and recognition (ICDAR)*, pp. 1577–1581. IEEE, 2019.

[126] Song-Hai Zhang, Ruilong Li, Xin Dong, Paul Rosin, Zixi Cai, Xi Han, Dingcheng Yang, Haozhi Huang, and Shi-Min Hu. Pose2seg: Detection free human instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 889–898, 2019.

[127] Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. Detrs beat yolos on real-time object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16965–16974, 2024.

[128] Zhiyuan Zhao, Hengrui Kang, Bin Wang, and Conghui He. Doclayout-yolo: Enhancing document layout analysis through diverse synthetic data and global-to-local adaptive perception. *arXiv preprint arXiv:2410.12628*, 2024.

[129] Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. Publaynet: largest dataset ever for document layout analysis. In *2019 International conference on document analysis and recognition (ICDAR)*, pp. 1015–1022. IEEE, 2019.

[130] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, 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.

[131] Enshen Zhou, Jingkun An, Cheng Chi, Yi Han, Shanyu Rong, Chi Zhang, Pengwei Wang, Zhongyuan Wang, Tiejun Huang, Lu Sheng, et al. Roborefer: Towards spatial referring with reasoning in vision-language models for robotics. In *Adv. Neural Inform. Process. Syst.*, 2025.

[132] Chenchen Zhu, Fanyi Xiao, Andrés Alvarado, Yasmine Babaei, Jiabo Hu, Hichem El-Mohri, Sean Culatana, Roshan Sumbaly, and Zhicheng Yan. Egoobjects: A large-scale egocentric dataset for fine-grained object understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 20110–20120, 2023.

[133] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Yuchen Duan, Hao Tian, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

[134] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2020.

[135] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. *Advances in neural information processing systems*, 36:19769–19782, 2023.

A. Appendix

A.1. More Visualization Results

To provide a more comprehensive and intuitive understanding of Rex-Omni's capabilities, this section presents additional qualitative results across a wide range of visual perception tasks. These visualizations complement the quantitative results reported in the main paper, offering further insights into the model's performance in diverse and challenging scenarios. We showcase more visualization results for the following tasks:

- Common and Long-tailed Object Detection (Figure 19)
- Dense Object Detection (Figure 20)
- Object Referring (Figure 21)
- Object Pointing (Figure 22)
- Layout Grounding (Figure 23)
- OCR (Optical Character Recognition) (Figure 24)

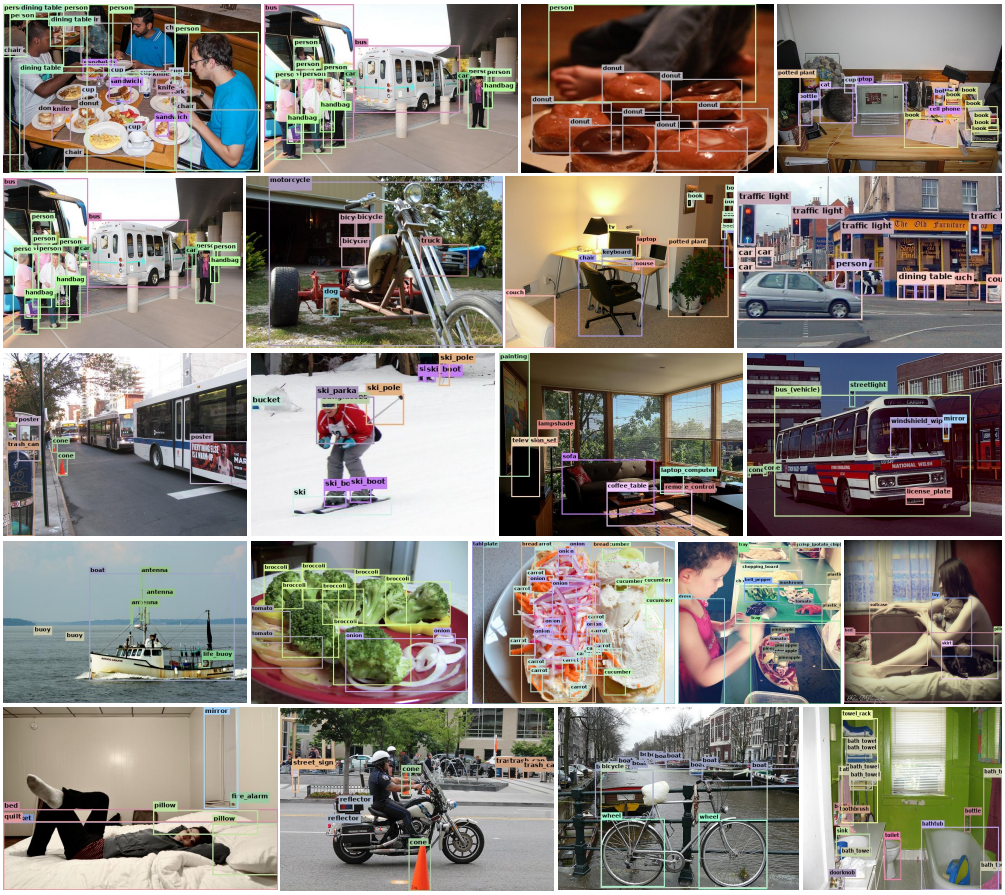


Figure 19: Visualization results of Rex-Omni on common and long-tailed object detection task.

Figure 20: Visualization results of Rex-Omni on dense object detection task.

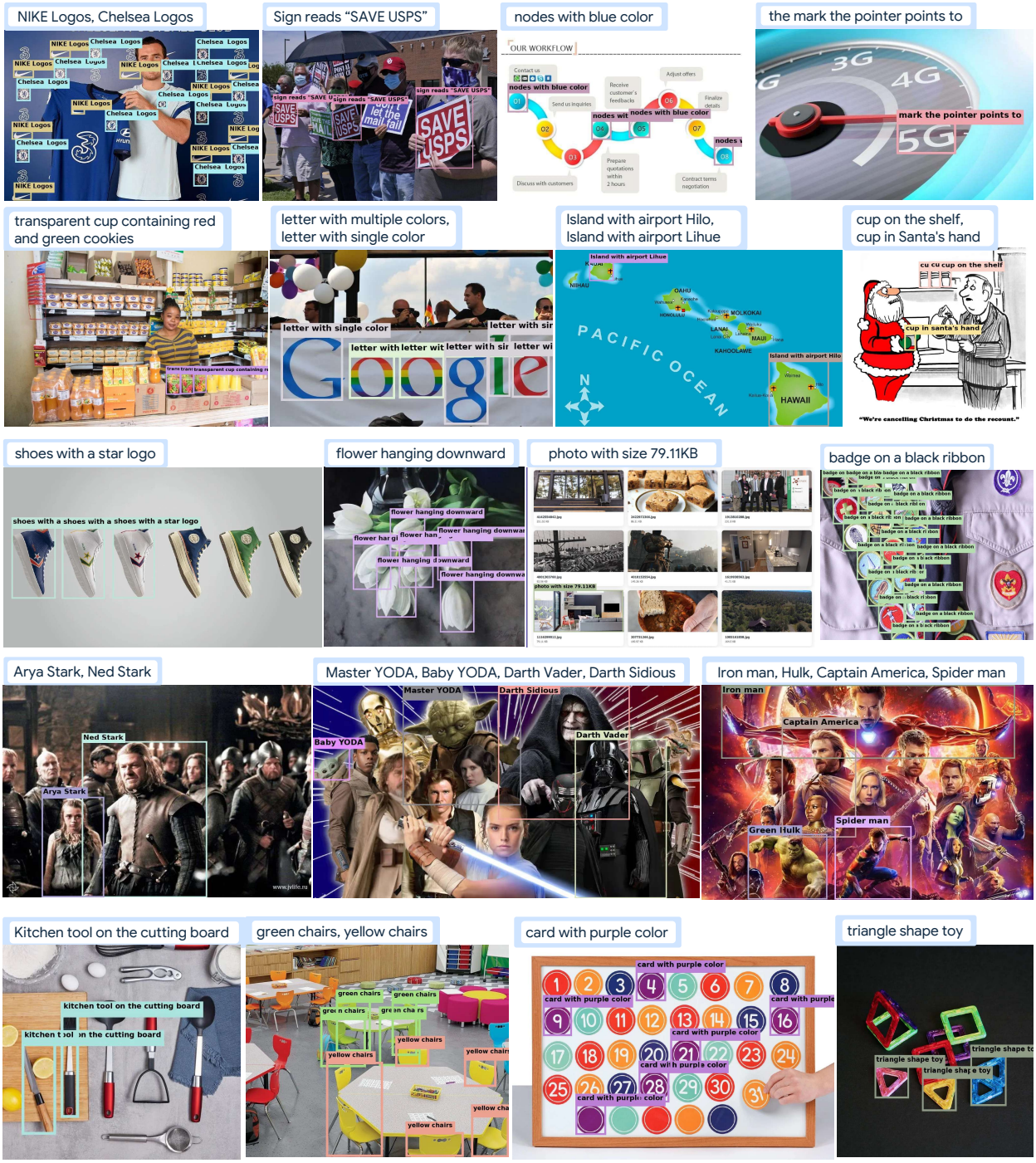


Figure 21: Visualization results of Rex-Omni on object referring task.

Figure 22: Visualization results of Rex-Omni on object pointing task.

Figure 23: Visualization results of Rex-Omni on layout grounding task.

Figure 24: Visualization results of Rex-Omni on OCR task.