A. More Implementation Details.

The statistical results of COCO are listed in Table A.

	train	validation
#images	118,287	5,000
#bboxes	860,001	36,781
#bboxes per image	7.3	7.4

Table A. Statistical results of COCO.

The training hyperparameters of LMM-Det cross three stages are listed in Table B.

Configuration	Stage I	Stage II	Stage III
Training epochs	1	5	12
Global batch size	192	480	288
Learning rate	1e-3	2e-5	2e-5
Learning rate schedule	Cosine decay		
Warmup ratio	0.03	0.05	0.05
Weight decay	0	0.05	0.05
Optimizer	AdamW		
Optimizer hyperparameters	$\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1e-8$		
Deepspeed ZeRO stage	ZeRO-2	ZeRO-3	ZeRO-3
Text max sequence length	2k	4k	10k
Training precision	bf16		

Table B. Training hyperparameters in three stages.

More details of the exploratory experiments. In Section 2.1, we re-train LLaVA with detection data (*i.e.*, Object365 and COCO) using the same hyperparameters in Stage III of Table B except for the text max sequence length , which is set to 2k in the exploratory experiments. During both training and inference stages, we output all prediction bounding boxes simultaneously. Additionally, we also predict bounding boxes for each category when conducting zero-shot detection on COCO. Unfortunately, this approach fails to improve detection performance and instead increases the number of incorrect prediction bounding boxes.

More implemetation details during training and inference. During training, we construct multi-turn conversations for each input image. To mitigate potential overfitting, we randomize each turn of the conversation and each bounding box in the target sequence for each training epoch. For each image from either COCO or Object365, we construct both positive and negative conversations at a 1:1 ratio based on the number of existing categories in the image. Specifically: 1) For an image containing n visible categories (e.g., cat, dog), we generate n positive instructions where the model is asked to output bounding boxes. 2) We then sample n negative instructions by randomly selecting non-present categories from the remaining label set (*i.e.*, $80 - n$ for COCO, $365 - n$ for Object365). 3) The maximum number of instruction rounds per image is capped at 80 for COCO and 365 for Object365, respectively. Importantly, we do not filter the Object365 dataset but retain all its categories and instances to preserve the model's broad detection capability.

During inference, we predict bounding boxes with confidence scores. Notably, the mean number of bounding boxes per image increases from 7 to 31 due to the integration of pseudo-labels and re-organized instruction data. However, the number of generated proposals is still lower than that of specialized models (e.g., 900 proposals in Salience-DETR) when calculating the AP. Therefore, we do not apply non-maximum suppression (NMS) and set a threshold for calculating AP and AR. For better visualization, we use a score threshold of 0.5 and NMS with a threshold of 0.5.

More details of zero-shot experiments. In Section 4.2, we compare LMM-Det with a variety of large multimodal models in a zero-shot manner. We provide the detailed prompt for all models in Table C. For simplicity, we omit the <image> token for KOSMOS-2 and Groma. We omit the special token in the prompt of QwenVL-2.5 such as <|im_start|> and <|im_end|>. In particular, we randomly select the templates of the referring expression comprehension (REC) task in Shikra to perform the zero-shot experiments on COCO. We illustrate an example for Shikra in Table C.

More details of versatile LMM-Det. In Section 4.4, the experiments demonstrate that LMM-Det not only unlocks object detection capabilities but also preserves inherent multimodal capabilities such as image captioning and visual question answering. Specifically, after training LMM-Det through the three stages outlined in the training recipe in Section 4.1, we add a fourth stage. In this stage, we train the projector and large language model while freezing the visual encoder, using the 665K data from LLaVA [23] and the proposed re-organized instruction data. We use the same hyperparameters as those in the fine-tuning stage of LLaVA.

B. More Quantitative Results.

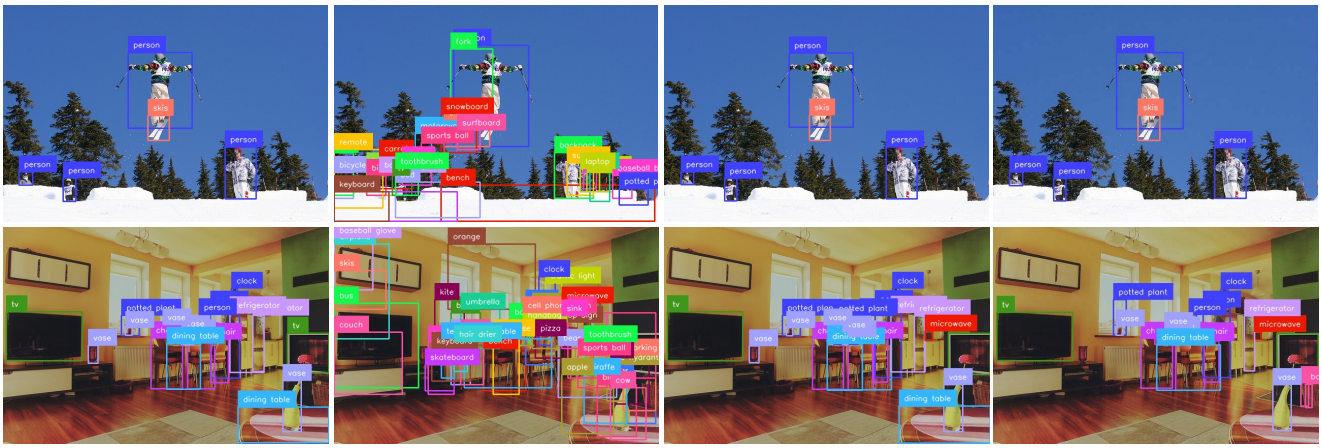
This paper reveals the root cause for the poor performance of LMMs in the field of object detection (ODet). LMM-Det can process multimodal tasks (*e.g.*, ODet + referring expression comprehension (REC) + image captioning + visual question answering) while Griffon [51] and Griffon v2 [50] focus on ODet+REC. Table D further indicates that LMM-Det† can also unify object detection with the REC task and shows mutual performance benefits (e.g., 81.4 $\Rightarrow$ 85.7).

Model	COCO	RefCOCO val	MMStar
LLaVA-7B	0.2	81.4	30.3
Griffon-13B	24.8	88.0	-
Griffon v2-13B	38.5	89.6	-
LMM-Det†-7B	47.1	85.7	32.1

Table D. More quantitative results for versatile LMM-Det†.

Model	Multi-step	CLIP emb	Prompt
LLaVA [23]	✓	×	<image>\nProvide the bounding box coordinate of the region this sentence describes if region exists in the image: <category>
Shikra [7]	✓	×	May I have the coordinates of <category> in <image>?
KOSMOS-2 [30]	✓	×	<grounding> Where is the <category>?
InternVL-2.5 [8]	×	✓	<image>\nPlease detect and label all objects in the following image and mark their positions.
Groma [27]	×	✓	[grounding] Please summarize the content of this image in detail.
LMM-Det (Ours)	✓	×	<image>\nDetect all the objects in the image that belong to the category set <category>.

Table C. Detailed prompt for performing zero-shot object detection task on COCO. "Multi-step" denotes whether to use multi-step inference to predict images. For each image, we construct 80 steps for LLaVA to predict the bounding boxes on COCO. "CLIP emb" represents whether to use CLIP embbedings. In this way, we map the unknown category to the pre-defined categories (*e.g.*, 80 categories on COCO).



Specialist Detector | LLaVA* | LMM-Det (Ours) | Ground-Truth

Figure A. Detailed predicted labels of Figure 1 in the manuscript.

Table E demonstrates that Owlv2-ViT works better than other visual encoders. In particular, the experiments (Tables (E, F, G) are trained on only COCO for simplicity.

Model	Res.	mAP	AP_{50}	AP_{75}	AR@100
CLIP-ViT-L	336	16.0	31.2	14.4	26.7
DINOv2-L	224	12.3	24.5	11.3	19.1
OWLv2-ViT-L	1008	32.6	50.5	34.4	43.1

Table E. More ablation studies of the visual encoder.

The effect of extra vocabularies is shown in Table F.

Techniques	AP	AP_{50}	AP_{75}	AR@100
LMM-Det	32.6	50.5	34.4	43.1
+ extra vocabularies	29.2	47.3	29.9	40.8

Table F. The effectiveness of extra vocabularies.

The effectiveness of the sampling strategy is listed in Table G. We use greedy decoding as the inference sampling strategy for all experiments in the manuscript.

C. More Qualitative Results.

The detailed predicted labels of all images in Figure 1 of the manuscript are shown in Figure A.

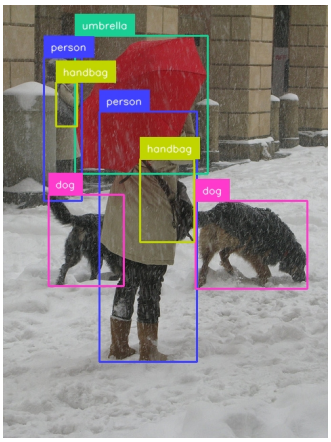
Model	Techniques	AP	AP_{50}	AP_{75}	AR@100	Cost/Img
LMM-Det	greedy decoding	32.6	50.5	34.4	43.1	4.0 s
	beam search (beam=2)	33.0	51.3	34.9	43.7	8.2 s
	beam search (beam=3)	32.8	51.1	34.5	43.6	10.9 s
	beam search (beam=4)	24.6	38.2	25.8	32.5	13.8 s
	top-p sampling	27.6	45.6	28.0	41.7	4.3 s

Table G. Inference Sampling strategy. Cost/Img denotes the inference computational cost for one image, which is measured on one Nvidia H800 GPU without TensorRT. We average the inference cost of all images on COCO val set to obtain Cost/Img.

In inference optimization, we re-organize the object detection instruction data to maintain consistency between inference and training. We show an example in Figure B.

We further provide the adjusted bounding box distribution in Figure C. From Figure C, the data distribution adjustment effectively increases the recall rate and improves overall detection performance (the model with AP=47.5).

As illustrated in the ablation study of the manuscript, the AP of LMM-Det increases from 38.7 to 47.5 after introducing the proposed strategy. We provide the qualitative results for these significant results to show the recall enhancement in Figure D.



User: <image>\n Detect all the objects in the image that belong to the category set {person}.

LMM-Det: [0.095, 0.121, 0.182, 0.452], 1.000; [0.223, 0.246, 0.448, 0.824], 1.000

User: Detect all the objects in the image that belong to the category set {dog}.

LMM-Det: [0.106, 0.439, 0.279, 0.649], 1.000; [0.445, 0.452, 0.702, 0.656], 1.000

User: Detect all the objects in the image that belong to the category set {handbag}.

LMM-Det: [0.123, 0.179, 0.171, 0.280], 1.000; [0.317, 0.356, 0.442, 0.548], 0.510

User: Detect all the objects in the image that belong to the category set {umbrella}.

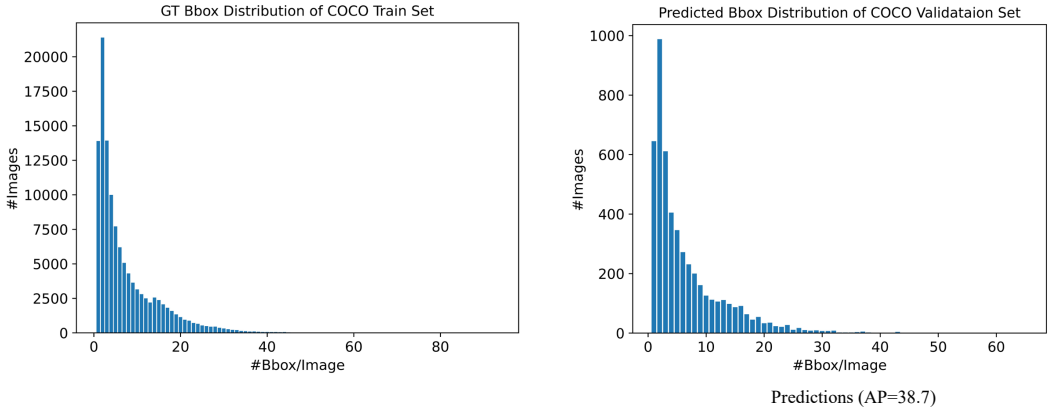
LMM-Det: [0.167, 0.071, 0.472, 0.390], 1.000

User: Detect all the objects in the image that belong to the category set {motorcycle}.

LMM-Det: There are no objects in the image that belong to the required category set.

Figure B. Illustration of the re-organization data designed for object detection over large multimodal models. More details of the prediction confidence of bounding boxes and the post-processing of LMM-Det's answers can be referred to Section A of Appendix.

Before data distribution adjustment



After data distribution adjustment

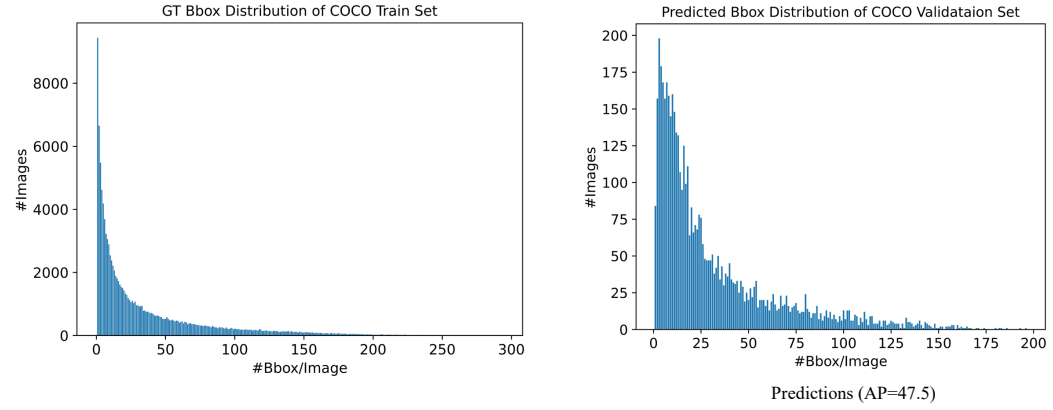
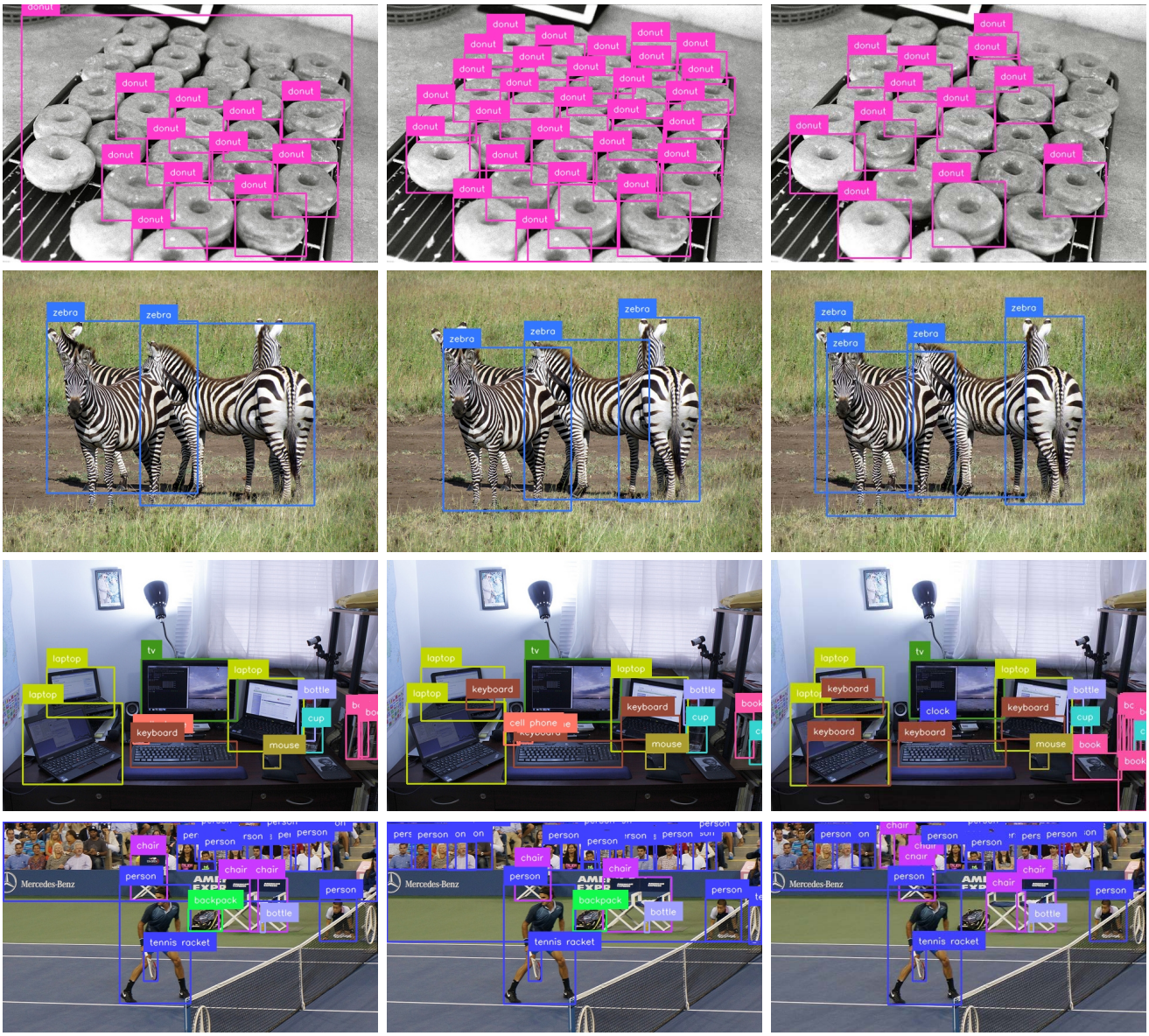


Figure C. Bounding box distribution of ground-truth and predictions before/after data distribution adjustment. The models with AP=38.7 and AP=47.5 can be referred to Table 3.



LLaVA* (AP=38.7) LMM-Det (AP=47.5) Ground-Truth

Figure D. Qualitative results for recall enhancement.