

ROD-MLLM: Towards More Reliable Object Detection in Multimodal Large Language Models

Supplementary Material

We provide additional information about the methods, datasets, and analysis results in the supplementary materials, which are organized as follows:

1. Limitations and Future Work (Appendix A)
2. Training Data Construction (Appendix B)
3. More Experimental Results (Appendix C)
4. Details for Annotation Pipeline (Appendix D)
5. Dataset Statistics and Visualization (Appendix E)
6. More Qualitative Results (Appendix F)

A. Limitations and Future Work

Our approach achieves more reliable object detection of MLLMs by implementing and improving language-based object detection. However, we currently only consider the detection of objects directly mentioned in the description, without considering detection scenarios that require reasoning, such as detecting “*objects that can be used to supplement vitamins*” in an image. These types of problems require more complex data construction logic, which we will address in future research.

B. Training Data Construction

Training Data Statistics. Tab. 7 shows all the datasets used for training. In the pre-training stage, we use a total of 3.6M samples, and in the instruction-tuning stage, we use a total of 1.5M samples.

Training Stage	Task Type	Datasets
Pre-Training	Image Caption	LAION-CC-SBU
Shared Part	Object Detection	MSCOCO Objects365
	Language Based Object Detection	ROD
	REC	RefCOCO+/g gRefCOCO GRIT-20M
	Region Caption	RefCOCOg Visual Genome
	Grounded Caption	Flickr30K Entities
Instruct-Tuning	Instruction Following	LLaVA-Instruct
	Referential Dialogue	VCR

Table 7. Datasets used for training. Shared Part represents the datasets used in both the pre-training and instruct-tuning stages, with a lower sampling rate during instructing-tuning.

Prompt Templates. Tab. 10 lists various instruction templates for different tasks. The *{expression}* represents the object expression, which can be a category or description.

The *{region anchor token}* refers to tokens that denote regions, such as *<a3>*.

C. More Experimental Results

General Object Detection. In addition to language-based object detection, we also evaluate general object detection. As shown in Tab. 8, while maintaining a comparable number of training epochs, our ROD-MLLM outperforms some specialized detectors (FRCNN-R50 and DETR-DC5) and exceeds other general MLLMs for object detection (+47.8% mAP). This shows that our method also has good results for simple category detection.

Type	Methods	Epochs	mAP	AP ₅₀	AP ₇₅	AP _S	AP _M	AP _L
S	FRCNN-R50	12	28.0	30.6	32.3	32.0	26.8	30.3
	Pix2Seq-R50	12	43.0	61.0	45.6	25.1	46.9	59.4
	DETR-DC5	12	15.5	29.4	14.5	4.3	15.1	26.9
G	Griffon	1	23.2	37.6	23.4	4.6	22.8	47.9
	ROD-MLLM*	1	34.3	48.3	37.5	14.3	37.0	54.4

Table 8. Performance of object detection on MSCOCO val2017. ROD-MLLM* refers to pre-training and fine-tuning for only one epoch. S stands for Specialists and G stands for Generalists.

Effect of Number of Region Tokens. We compare different numbers of encoded region tokens, as shown in Tab. 9. Compared to using only one token to encode a region, using four tokens to encode a region has a more significant impact on the region caption metric (CIDEr). This is because more region tokens can carry more local fine-grained visual features. Additionally, more region tokens also improve the performance of language-based object detection on D³.

Region token num	RefCOCOg		D ³		
	METEOR	CIDEr	Full	Pres	Abs
1 token per region	16.8	108.6	28.4	28.7	27.6
4 tokens per region	17.3	113.8	29.7	30.0	28.7

Table 9. Ablation of the number of tokens per region.

Effect of Ways to Build ROD (detection) Dataset. We validate the accuracy of ROD data annotation using different methods. As shown in Tab. 11, when the description is decomposed and then used for object and description matching, the annotation accuracy improves from 90.0% to 93.0%. This indicates that the MLLM’s judgment on individual conditions is more accurate than its judgment on the entire description. Furthermore, with the introduction of COT, the annotation accuracy further enhances to 95.5%. This demonstrates that allowing the MLLM to describe the object first provides a basis for the final judgment, thereby reducing annotation noise.

Task	Prompt Templates
Object Detection	Categories: <p>{expression}</p>×N. Locate the above categories in the image. Locate <p>{expression}</p>×N in the image. Point out <p>{expression}</p>×N in the picture.
REC	Where can I find <p>{expression}</p> in the image? Answer with regions. Which region matches the description <p>{expression}</p>? Answer with regions. Where exactly can I find <p>{expression}</p> in this image? Answer with regions.
Region Captioning	Provide a caption of the region <box>[{region anchor token}]</box>. Please describe <box>[{region anchor token}]</box> in details. What are the key details of the area <box>[{region anchor token}]</box> in the image?
Grounded Captioning	[grounding] What does this picture show? Please summarize briefly. [grounding] Please provide a concise summary of the image. [grounding] Give me a concise description of the image.
Referential Dialogue	What is <box>[{region anchor token}]</box> doing. Why does <box>[{region anchor token}]</box> look sad? Explain your reasoning before providing answers. Are <box>[{region anchor token}]</box> and <box>[{region anchor token}]</box> in a fight?

Table 10. Prompt templates for different tasks with three randomly selected examples for each task.

	Base	+Condition Decomposition	+COT
Accuracy	90.0%	93.0%	95.5%

Table 11. The annotation accuracy of 200 random samples of ROD (detection) using different annotation methods. “Base” represents directly matching a description and an object with the MLLM. “+Condition Decomposition” refers to first decomposing the conditions using the LLM before performing the matching. “+COT” indicates using the chain of thought method to describe the object before providing the matching result.

D. Details for Annotation Pipeline

We fully leverage the capabilities of existing MLLMs and LLMs to build our automated annotation pipeline. To minimize the noise, we decompose and simplify the annotation tasks as much as possible. For annotations requiring visual input, we use the state-of-the-art open source MLLM InternVL2-76B [8]. For text-only annotations, we use the API of the closed-source LLM DeepSeek [28]. Below, we provide the annotation details for ROD (detection) and ROD (grounding) respectively.

D.1. Building from detection dataset.

Step 1. Object Description Generation. The object detection dataset [44] provides annotations for 365 categories (e.g., person, car) of coordinate bounding boxes in a rich collection of images. Using this as our data source, we sample images for each category and ultimately obtain about 100K images to expand annotations from category to description. Given an image belonging to a certain category, we randomly select an object of that category in the image as the target for description generation. Prompts used are shown in Fig. 14, we randomly divide the description generation into three types: simple description, detailed description, and negative description. Simple and detailed de-

scriptions correspond to different lengths of descriptions, while negative descriptions focus on descriptions with negative semantics (e.g., *an airplane that is not taking off*). An image, along with an object’s category and its coordinates, will be used as input for InternVL2-76B [8] to generate a description. Therefore, the 100K images will generate a set S_{desc} containing about 100K object descriptions.

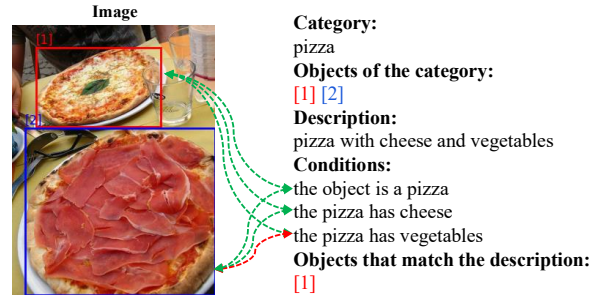


Figure 7. An example of ROD (detection) annotation. The green line represents that the object matches the condition, while the red line represents that it does not match the condition.

Step 2. Description Condition Decomposition. The goal of object detection is to find all objects in the image that meet the criteria specified by an object expression. Therefore, for the set of descriptions generated in the first step, we need to determine whether other objects in the same image also meet the criteria. For some complex descriptions (e.g., *a pickup truck with a silver body parked near a building*) and an object in the image, it is prone to errors for MLLM to directly determine whether the two match (as shown in Tab. 11). Therefore, before making a judgment, we first decompose a description into several conditions to reduce the difficulty of subsequent matching judgments. As shown in Fig. 15, we use a few-shot approach to guide the LLM DeepSeek [28] in performing condition decomposition.

Step 3. COT based Condition Judgment. As shown in Fig. 7, after the first two steps, we have the following information: Image, Category, Objects of the category, Description, and Conditions. Next, we match the Conditions with the Objects one by one. Specifically, we use the prompts shown in Fig. 16 to drive InternVL2-76B [8] for this process. We guide it to first describe the object and then determine the matching condition through instructions and examples: *First give the reason based on the object content, and then give a yes or no answer.* Using this method, we determine that in Fig. 7, only object [1] matches the description, thus obtaining a description-image sample.

D.2. Building from grounding dataset

Prompts we used for building ROD (grounding) are shown in Fig. 17, we mark the entities in the captions provided for each image in the Flickr 30K Entities [41] dataset, e.g., “<object:147225>a table</object:147225> covered with <object:147226>food</object:147226>”. Next, we use few-shot examples to prompt DeepSeek [28] to construct descriptions of objects that are present and absent. Since each constructed description corresponds to an object ID, we can map this ID to the object bounding boxes provided in the Flickr30K Entities [41].

E. Dataset Statistics and Visualization

Statistics. Tab. 12 presents the detailed statistics of our automatically annotated ROD dataset. The final ROD contains 296K different descriptions and 225K distinct images. It can be observed that the construction based on the detection dataset and the grounding dataset has characteristics of being rich in images and rich in descriptions, respectively. We also analyze the distribution of description lengths, as shown in Fig. 8a, where most descriptions contain 4 to 6 words. Additionally, we calculate the proportion of descriptions with negative semantics, as illustrated in Fig. 8b, where approximately 10.1% of the descriptions contain negative semantics such as *without* or *that is not*.

Metric	ROD (detection)	ROD (grounding)	ROD
#Images	196,041	29,781	225,822
#Desc.	62,339	235,637	296,156
#Image-Desc. Pairs	240,256	302,386	542,642
Non-Existent (%)	49.1	50.7	50.0
Avg Desc. Length	4.5	5.6	5.1
#boxes/Desc.	2.88	1.02	1.42

Table 12. Statistics of ROD Dataset.

Dataset Visualization. We present samples from the ROD dataset in Fig. 10. It can be seen that ROD contains images with diverse descriptions of matching or non-matching objects. Our ROD contains some complex samples, such as “speed limit 35 sign” when the actual sign in the image indicates a speed limit of 50. This requires the model to have

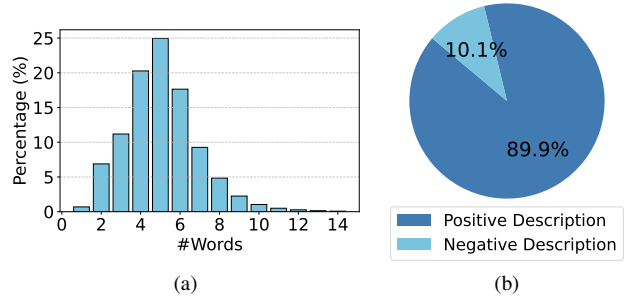


Figure 8. ROD description length distribution statistics and the proportion of negative descriptions.

fine-grained understanding and judgment capabilities to reject the description.

Dataset Noise. As shown in Fig. 9, we verify the noise categories present in ROD, which mainly include attribute, relation, position and missing.


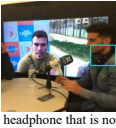


Type	Attribute	Relationship	Position	Missing
Example	 wooden cutting board	 headphone that is not connected to a device	 pizza with black olives and vegetables on the left	 metal bunk bed with white frame
Count	12	2	1	11

Figure 9. Statistics of error type for 500 randomly selected ROD.

F. More Qualitative Results

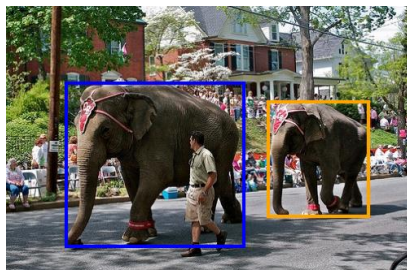
Language Based Object Detection. Fig. 12 shows more examples of language-based object detection. For the same image and different description queries, our method can identify the target or determine its absence. We also present some **Error Cases**. In the first error case, our method misses a person without a hat, likely because the person is heavily occluded by objects in the foreground, making it difficult for ROI Align to extract accurate features. In the second error case, our method incorrectly determines that there is no matching target. This is because the cow’s tail occupies a very small area in the image, and the resolution of 336 we used may limit the extraction of detailed features in that part. Increasing the resolution is one of our future improvement directions.

General Object Detection. Fig. 11 shows detection examples on the MSCOCO val2017 dataset [27]. Our method adapts well to the detection of multiple categories as well as the identification of non-existent categories.

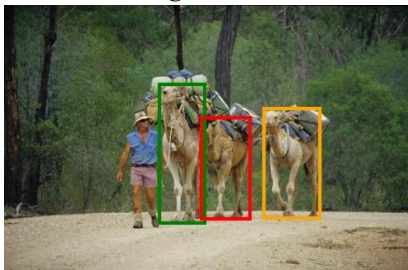
Grounded Caption and Referential Dialogue. Fig. 13 demonstrates that our method can effectively handle both localization and generation tasks. For grounded captioning, our approach accurately grounds entities to the corresponding regions in the image. For referential dialogue, our method can understand the user’s spatial references, enabling it to perform region description or reasoning tasks.

Samples of ROD (detection)

elephant with red straps



camel carrying a load walking on a gravel trail



glasses that are not blue



speed limit 35 sign (None)



pink gloves on the child (None)



tricycle without a rider (None)



Samples of ROD (grounding)

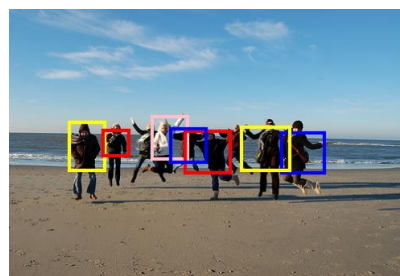
men playing guitars



orange truck with the name "Grilled Cheese Truck"



jackets worn by the group

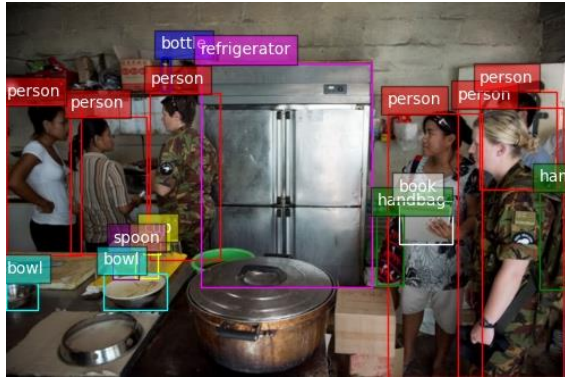


two people riding bicycles (None) *players wearing red and green (None)* *bull that is not kicking (None)*



Figure 10. Visualization of some samples from the ROD dataset.

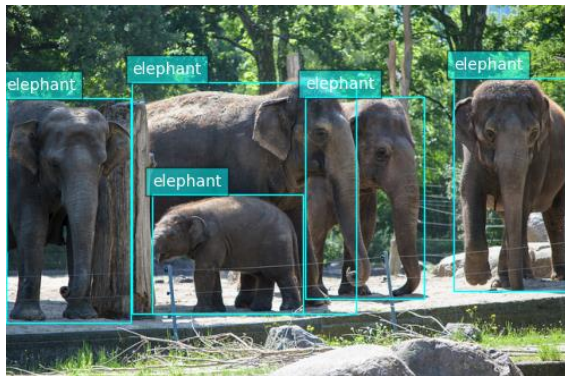
General Object Detection



Categories:

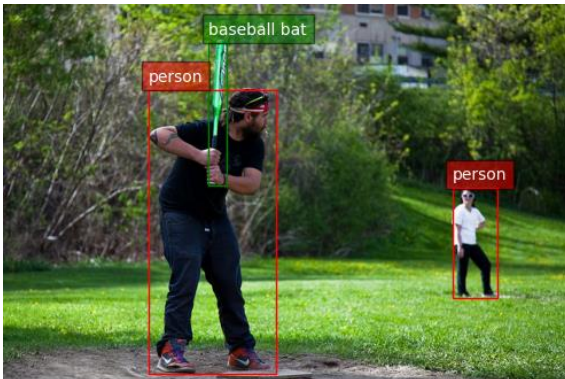
<p>person</p>, <p>bicycle</p>, <p>car</p>, <p>motor cycle</p>, <p>airplane</p>, <p>bus</p>, <p>train</p>, <p>truck</p>, <p>boat</p>, <p>traffic light</p>...
Locate the above categories in the image.

<p> person </p> <box>[<a8> <a11> <a38> <a34> <a20> <a27>]</box> , <p> handbag </p> <box>[<a47> <a17>]</box> , <p> bottle </p> <box>[<a42>]</box> , <p> cup </p> <box>[<a35> <a19>]</box> , <p> spoon </p> <box>[<a13>]</box> , <p> bowl </p> <box>[<a23> <a41>]</box> , <p> refrigerator </p> <box>[<a31>]</box> , <p> book </p> <box>[<a36>]</box>



Point out <p>elephant</p> in the image.

<box>[<a6> <a4> <a1> <a3> <a2>]</box>



Detect <p>person</p>, <p>baseball bat</p>, <p>baseball glove</p> in the picture.

<p> person </p> <box>[<a3> <a2>]</box> , <p> baseball bat </p> <box>[<a0>]</box>



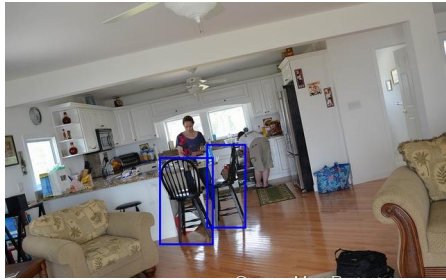
Help me find all <p>dog</p> in the image.

None



Figure 11. Visualization of examples from general object detection.

Language-Based Object Detection Examples



Locate all

Black stool chairs with back rests at the counter

 in the image.

`<box>[<a11> <a8>]</box>`



Detect

the chair that has a person wearing a dark-colored coat sitting in it

.

None



Help me find all

the vehicles with the pedals

.

`<box>[<a0> <a2>]</box>`

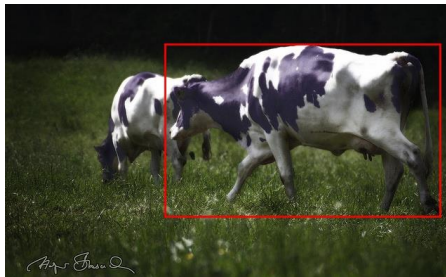


Locate

a green automobile for multiple passengers

.

None



Find objects whose semantics satisfy

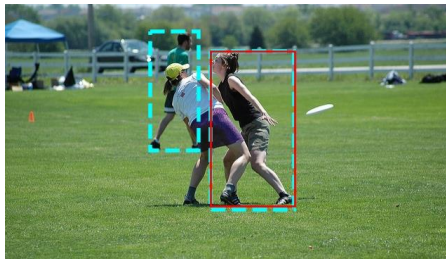
the cow that is not eating grass

.

`<box>[<a1>]</box>`



Error Cases

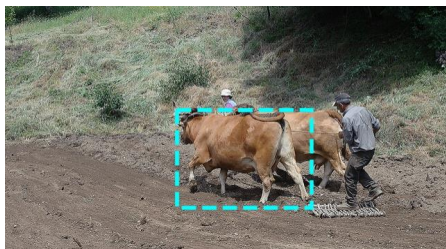


Find

a person not wearing a hat

 in the picture.

`<box>[<a2>]</box>`



Find objects whose semantics satisfy

a cow whose tail is curled on its back

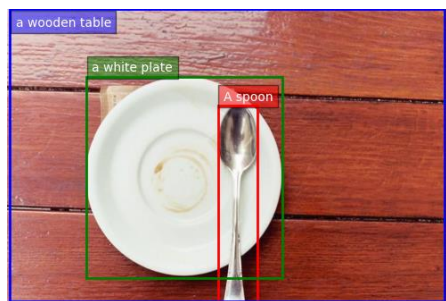
.

None



Figure 12. Visualization of examples from language based object detection.

Grounded Caption



[grounding] Please provide a concise summary of the image.

<p>A spoon</p><box>[<a0>]</box> rests on <p>a white plate</p><box>[<a1>]</box> on<p>a wooden table</p><box>[<a2>]</box> .

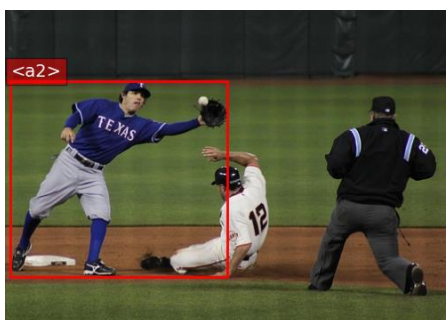


[grounding] Please briefly summarize the content of this image.

<p> Two women </p> <box>[<a0> <a1>]</box> in <p> aprons </p> <box>[<a8> <a11>]</box> are standing in a kitchen .

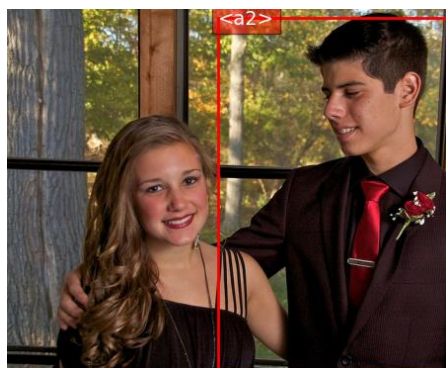


Referential Dialogue



Provide a caption of the region <box>[<a2>]</box>.

A baseball player in a blue uniform



How does <box>[<a2>]</box> feeling? Explain your reasoning before providing answers.

He is smiling and looking at the girl. So the answer to the question is: He is happy.



Figure 13. Visualization of examples from grounded caption and referential dialogue.

System Prompt:**ROD (detection) Step 1. Object Description Generation**

You are an AI visual assistant that can analyze a single image. Specific object locations within the image are provided, along with detailed coordinates. These coordinates are in the form of bounding boxes, represented as [x1, y1, x2, y2] with int numbers ranging from 0 to 1000. These values correspond to the top left x, top left y, bottom right x, and bottom right y. The task is to give a descriptive phrase for an object given.

Here are some additional requirements:

1. Output the descriptive phrase directly, do not output other information.
2. Do not mention bounding boxes in your output.
3. The subject of the descriptive phrase must be the given object. For example, if the given object is 'helmet', the phrase can be 'helmet on the man', but can not be 'man wearing the helmet'
4. Descriptive phrase can take into account information such as the object's attributes, states, relationships, behaviors, or relative positions.
5. Don't describe information you're unsure of.
6. Descriptive phrases should not exceed 50 words.

User:***Simple Description***

Examples:

“\n”.join([running horse, standing person, phone on the table, man in a suit, wooden chair, cup full of water])

Follow the above examples, give a descriptive phrase for {Object} located at {Coordinate}.

User:***Detailed Description***

Examples:

“\n”.join([person wearing a white T-shirt running, wheelchair with people sitting on it, cow with black spots eating grass, pizza next to a glass half full of water, stainless steel fork mounted on the wall, drum which is being beaten])

Follow the above examples, give a detailed descriptive phrase for {Object} located at {Coordinate}. Descriptive phrase needs to reflect the characteristics of the object.

User:***Negative Description***

Examples:

“\n”.join([dog that is not barking, tree without leaves, rider without helmet, bird that is not flying, chopsticks that are not made of iron])

Follow the above examples, give a descriptive phrase for {Object} located at {Coordinate}. Descriptive phrase needs to include negative descriptions such as "without" or "that is not". Note that although it contains a negative description, it still matches the given object.

Figure 14. Prompts for Step 1 of ROD (detection) annotation. The annotation model is InternVL2-76B [8]. {Object} represents the category name of the object, such as *printer*. {Coordinate} represents the bounding box of the object, such as [206, 416, 415, 610].

System Prompt:**ROD (detection) Step 2. Description Condition Decomposition**

You are a text analysis expert. You will receive a descriptive phrase representing a certain object. Please output the conditions contained in the phrase. Output one condition per line.

User:***Example Context***

black keyboard on desk

Assistant:

the object is a keyboard \n the color of the keyboard is black \n the keyboard is on a desk

User:

bathtub in the center of the room with woman lying in it

Assistant:

the object is a bathtub \n the bathtub is in the center of the room \n the bathtub is lied by a woman

...

User:

{Description}

Figure 15. Prompts for Step 2 of ROD (detection) annotation. The annotation model is DeepSeek [28]. {Description} represents a description obtained from Step 1.

System Prompt:**ROD (detection) Step 3. COT based Condition Judgment**

You are an AI visual assistant that can analyze a single image. Specific object locations within the image are provided, along with detailed coordinates. These coordinates are in the form of bounding boxes, represented as [x1, y1, x2, y2] with int numbers ranging from 0 to 1000. These values correspond to the top left x, top left y, bottom right x, and bottom right y. Some conditions are given. Please judge whether each object meets a certain condition. First give the reason based on the object content, and then give a yes or no answer. Please judge carefully based on the feature of the object from the image. The output format should follow the Judgment result section of the example given.

User:*Example*

Example Start

Object information:

object 1: person at [xxx, xxx, xxx, xxx]

object 2: person at [xxx, xxx, xxx, xxx]

Condition information:

condition 1: the object is a person

condition 2: the object is wearing a hat

condition 3: the object is holding a phone that is not blue

condition 4: the object is on the ground

Judgment result:

Is object 1 meet condition 1? Reason: object 1 is a person. Answer: yes.

Is object 1 meet condition 2? Reason: object 1 is wearing a headscarf, not a hat. Answer: no.

Is object 1 meet condition 3? Reason: object 1 is holding a phone, the color of the phone is red. Answer: yes.

Is object 1 meet condition 4? Reason: object 1 is located on the bed, not on the ground. Answer: no.

Is object 2 meet condition 1? Reason: object 2 is a person. Answer: yes.

Is object 2 meet condition 2? Reason: object 2 is wearing a blue cap. Answer: yes.

Is object 2 meet condition 3? Reason: object 2 is holding a phone, the color of the phone is blue. Answer: no.

Is object 2 meet condition 4? Reason: object 2 is on the ground. Answer: yes.

Example End

{Object information}

{Condition information}

Figure 16. Prompts for Step 3 of ROD (detection) annotation. The annotation model is InternVL2-76B [8].

System Prompt:**ROD (grounding) Step 1 & Step 2**

You are an AI assistant that can analyze the captions of images and generate object phrases. You receive multiple captions describing the same image. Each caption is annotated with the objects that appears in it, and the format of the annotation is `<object:object_id>object</object:object_id>`, where `object_id` is the unique ID of a kind of object and the same `object_id` refers to the same object in the image.

Based on the provided captions and object annotations, construct some object phrases that exist in the image and do not exist in the image.

Object phrases must meet the following requirements:

1. The object phrase must correspond to a object in captions, identified by `object_id`.
2. The format of the constructed object phrase is `<object:object_id>object phrase`.
3. Object phrases should take into account the information in the captions, such as attributes, states, actions, relationships.
4. Non-existent object phrases should be relevant to the information in the captions, but confusing.
5. Don't use the form "something instead of something" to construct non-existent objects.
6. The output contains both simple object phrases and longer and more complex object phrases.
7. Don't explain your output.
8. Generate 2-5 object phrases for both existent and non-existent objects.

User:**Example Context**

Captions for the image:

`<object:147223>A woman</object:147223>` in `<object:147224>a knitted brown jacket</object:147224>` looking at the camera while sitting in front of `<object:147225>a table</object:147225>` covered in

`<object:147226>food</object:147226>` and `<object:147228>food preparation tools</object:147228>` .

`<object:147223>A woman</object:147223>` in `<object:147224>an olive-brown sweater</object:147224>` gazes over `<object:147225>the table of the rustic kitchen</object:147225>`

`<object:147223>A woman</object:147223>` in `<object:147224>a brown sweater</object:147224>` sits at `<object:147225>a table</object:147225>` covered with `<object:147226>food</object:147226>` .

`<object:147223>A woman</object:147223>` in `<object:147229>pioneer dress</object:147229>` is sitting at `<object:147225>a table</object:147225>` .

`<object:147223>A woman</object:147223>` is just simply sitting in `<object:147227>a chair</object:147227>` .

Please provide the object phrases that exist and the confusing objects that do not exist.

Assistant:

Exist object phrases:

`<object:147223>woman in a knitted brown jacket`

`<object:147224>sweater worn by woman`

`<object:147225>table covered with food`

`<object:147227>chair with people sitting on it`

`<object:147226>food on the table`

Nonexist confusing object phrases:

`<object:147223>woman in white sweater`

`<object:147223>woman that is standing`

`<object:147223>man in a knitted brown jacket`

`<object:147225>table with no food`

`<object:147227>table with a man sitting next to it`

...

User:

{Captions}

Please provide the object phrases that exist and the confusing objects that do not exist.

Figure 17. Prompts for ROD (grounding) annotation. The annotation model is DeepSeek [28]