This supplementary document expands upon our main paper by offering additional insights into our proposed Language-prompted Detection Dataset, implementation details, and more visualization results, constrained by length limit. In Appendix A, we delineate the data processing procedure, data reformatting, and instruction template generation, illustrated with examples in Appendix C. The hyperparameter settings for our paper are outlined in Appendix B. Lastly, we employ a demo style to present additional visualization results for various tasks.

A Details of Language-prompted Detection Dataset

Overall. After collecting data from public datasets, we further organize, process, and optimize the data, and propose the Language-prompted Detection Dataset. Overall, we have 6M pre-training data and nearly 900K instruction following data. Detailed statistics of the Language-prompted Detection Dataset can be found in Tab. 6.

Table 6: The statistic of the composition of Language-prompted Detection Dataset. OD represents object detection, and PG represents the phrase grounding with positive tokens for referents. 1 *vs.* 1 denotes the Single Referent scenario, n *vs.* n denotes Multi Categories with Multi Objects scenario, 1 *vs.* n denotes One Category with Multi Objects, and None denotes None-Existing scenario.

	Vol.	Type	Source Datasets
Pre-training	4M	REC	RefCOCO, RefCOCO+/g, Visual Genome
	2M	OD	Objects365, MSCOCO
Instruction-following	288K	1 *vs.* 1	RefCOCO, RefCOCO+/g
	118K	n *vs.* n	MSCOCO
	427K	1 *vs.* n	Flickr30K Entities
	40K	None	LVIS

Data Processing. The image and annotation processing involves three critical steps to ensure optimal data quality. Initially, we exclude images with a maximum resolution smaller than 250 pixels to prevent adverse effects on training. Subsequently, we rigorously review each image's annotations, discarding those with discrepancies between actual and annotated sizes – a common issue with internet-sourced images. Lastly, we validate the annotations to eliminate inaccuracies, focusing on refining bounding boxes and rectifying incomplete category labels.

Data Reformatting. We have tailored our approach to reformat diverse source datasets into a unified language-prompted style, aligning with specific scenarios:

– For the One Category with Multi Objects data, unlike traditional methods using Flickr30K Entities that focus on caption-based object grounding, we extract phrases and their corresponding bounding boxes. Each phrase and its bounding boxes constitute a single annotation, aligning more closely with everyday usage.
– For the Non-existing data, we first construct from the LVIS dataset with its negative categories annotations. For each image containing the negative categories, we randomly select one category and pair it with the image to form an entry. Additionally, to increase the variety of the text granularity, we mine negative referents from the attributes of the object such as the color and position with GPT-4V exampled in Fig. 5.
– For the Multi Categories with Multi Objects, we concatenate the annotations of each image in the object detection datasets, ensuring comprehensive coverage of this scenario.

Instruction Templates Generation. With the reformatted data, we utilize the template generation method proposed in Sec. 3.1 (main paper) to generate various templates for different scenarios. As shown in Tab. 8, we input the information listed in the table to the GPT-4V and the required templates will be returned. We provide some examples of the templates we generate in Fig. 6.

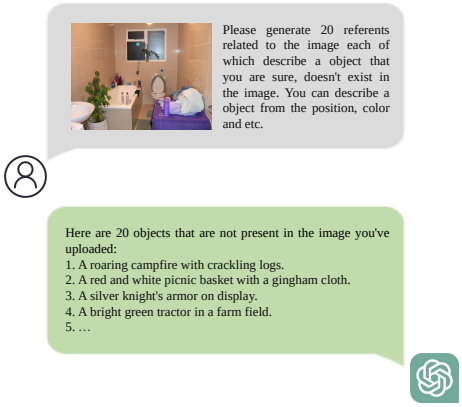


Fig. 5: Non-existing data mining with GPT-4V. We input the requirement, example and image to the GPT-4V to ask it to return the objects that don't appeat in the image with descriptions in color, position and *etc*.

B Implementation Details

Griffon use the same set of the hyperparameters as the original LLaVA as shown in Tab. 7. Due to the finetuning of the visual encoder and the increase of the amount of data, we increase the global batch size to 256 to speed up the training

and linearly scale the learning rate. We train the **Griffon** with 4×8 NVIDIA A100/800s.

Table 7: Hyperparameters of **Griffon**. The batch size and learning rate are increased in the pre-training stage compared with the original LLaVA.

Hyperparameter	Pre-training	Instruction Tuning
batch size	256	128
lr	4e-5	2e-5
lr schedule	cosine decay	
lr warmup ratio	0.03	
weight decay	0	
epoch	1	1
optimizer	AdamW	
DeepSpeed stage	2	

C Task Instruction Prompt Examples

As demonstrated in Sec. 3.1, we create instruction templates for each task, and random sample one template to form one piece of data during training. Here we list some examples of the instruction templates we used in our model in Fig. 6.

D More Visualizations

In this section, we present additional visualizations to showcase the capabilities of **Griffon** in object localization from text descriptions of varying granularity. For each scenario, distinct images and instruction inputs are used. As illustrated in Fig. 7, **Griffon** excels in distinguishing the targeted objects from others, accurately localizing all visible objects from free-form text inputs at any granularity, and determining the presence of specified instances. Notably, as highlighted in the first example of Fig. 7c and the third example of Fig. 7d, **Griffon** adeptly identifies all targets in densely populated scenes and doesn't overlooks small objects.

Table 8: Instruction inputs to the GPT-4V to generate the templates. The Single Referent and Non-existing scenarios share the same templates.

Inputs to GPT-4V to Generate Templates
Single Referent & Non-existing: Please refer to the given example and write 60 instruction templates to guide the model to localize the target object according to the input referent. The referent is represented by <expr>. An example is "Where is <expr> in the image". The expression of generated templates should be different.
One Category with Multi Objects: Please refer to the given example and write 60 instruction templates to guide the model to detect objects according to the input phrase, output the coordinates of each detected object and concatenate them with &. An example is "Please help me locate <some stores> in the image <image>, and concatenate them with & if detecting multiple objects.". The first <> contains the phrase describing the group of objects. <image> indicate the image input. Use <expr> to represent the phrase to be detected. The expression of generated templates should be different.
Multi Categories with Multi Objects: Please refer to the given example and write 60 instruction templates to guide the model to detect objects of those categories in the category set and output the coordinates of each detected object. One example is "Identify and locate all the objects from the category set in the image<image>. Please provide the coordinates for each detected object. The category set includes <category set>. The output format for each detected object is class name-[top-left coordinate, bottom-right coordinate], *e.g.* person-[0.001,0.345,0.111,0.678]. Concatenate them with &.

Template Examples for Single Referent and Non-existing Scenario

1. Where does the image feature <expr>? Point it out.
2. Can you elucidate the location of <expr>?
3. In the image, where can <expr> be observed?
4. Navigate me to the section containing <expr>.
5. Illustrate the position of <expr> in the visual.
6. Which quadrant of the image is <expr> located in?
7. Help me discern the exact location of <expr>.
8. Indicate the placement of <expr> in the scene.
9. Provide a detailed location of <expr> within the photo.
10. Direct my attention to the spot showcasing <expr>.
11. What location does <expr> hold in the picture <image>? Inform me of its coordinates.
12. Identify the position of <expr> in <image> and share its coordinates.
13. I'd like to request the coordinates of <expr> within the photo <image>.
14. How can I locate <expr> in the image <image>? Please provide the coordinates.
15. I am interested in knowing the coordinates of <expr> in the picture <image>.
16. Assist me in locating the position of <expr> in the photograph <image> and its bounding box coordinates.
17. In the image <image>, I need to find <expr> and know its coordinates. Can you please help?

(a) Template examples for Single Referent and Non-existing scenario.

Template Examples for Multi Categories with Multi Objects Scenario

1. Identify and locate all the objects from the category set in the image<image>. Please provide the coordinates for each detected object. The category set includes <category set>. The output format for each detected object is class name-[top-left coordinate, bottom-right coordinate], e.g. person-[0.001,0.345,0.111,0.678]. Concatenate them with &.
2. Detect the presence of objects belonging to the category set in the image<image> and provide the corresponding coordinates for each object. The category set includes <category set>. The output format for each detected object is class name-[top-left coordinate, bottom-right coordinate], e.g. person-[0.001,0.345,0.111,0.678]. Concatenate them with &.
3. Scan the image<image> for any objects from the category set and report the coordinates of each detected object. The category set includes <category set>. The output format for each detected object is class name-[top-left coordinate, bottom-right coordinate], e.g. person-[0.001,0.345,0.111,0.678]. Concatenate them with &.
4. Find all the objects in the image<image> that belong to the category set. Output the coordinates of each detected object. The category set includes <category set>. The output format for each detected object is class name-[top-left coordinate, bottom-right coordinate], e.g. person-[0.001,0.345,0.111,0.678]. Concatenate them with &.
5. Locate and identify the objects from the category set in the image<image>. Please provide the coordinates for each detected object. The category set includes <category set>. The output format for each detected object is class name-[top-left coordinate, bottom-right coordinate], e.g. person-[0.001,0.345,0.111,0.678].. Concatenate them with &.
6. Detect and locate objects belonging to the <category set> in the image <image> and provide their coordinates.
7. Locate and identify the objects from the <category set> in the image <image>.
8. Analyze the image<image> and detect any objects from the <category set>. Output the coordinates of each detected object.
9. Examine the image<image> for any objects from the <category set>. Report the coordinates of each detected object.
10. Perform object detection on the image<image> using the <category set>.

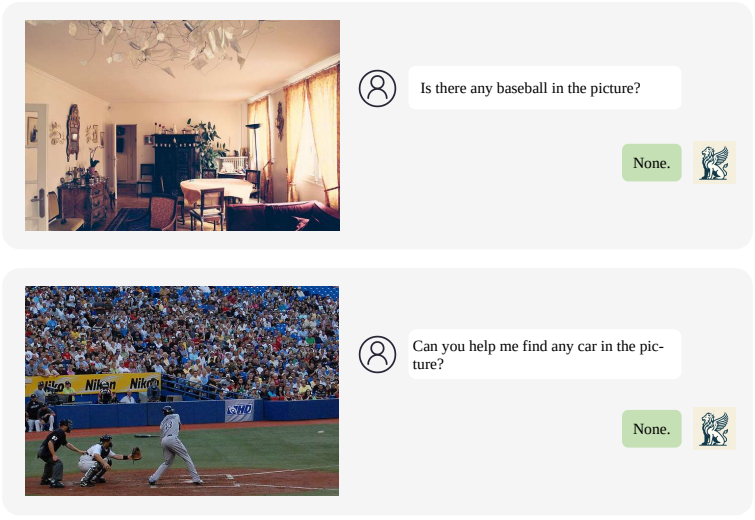
(b) Template examples for Multi Categories with Multi Objects scenario.

Template Examples for One Category with Multi Objects Scenario

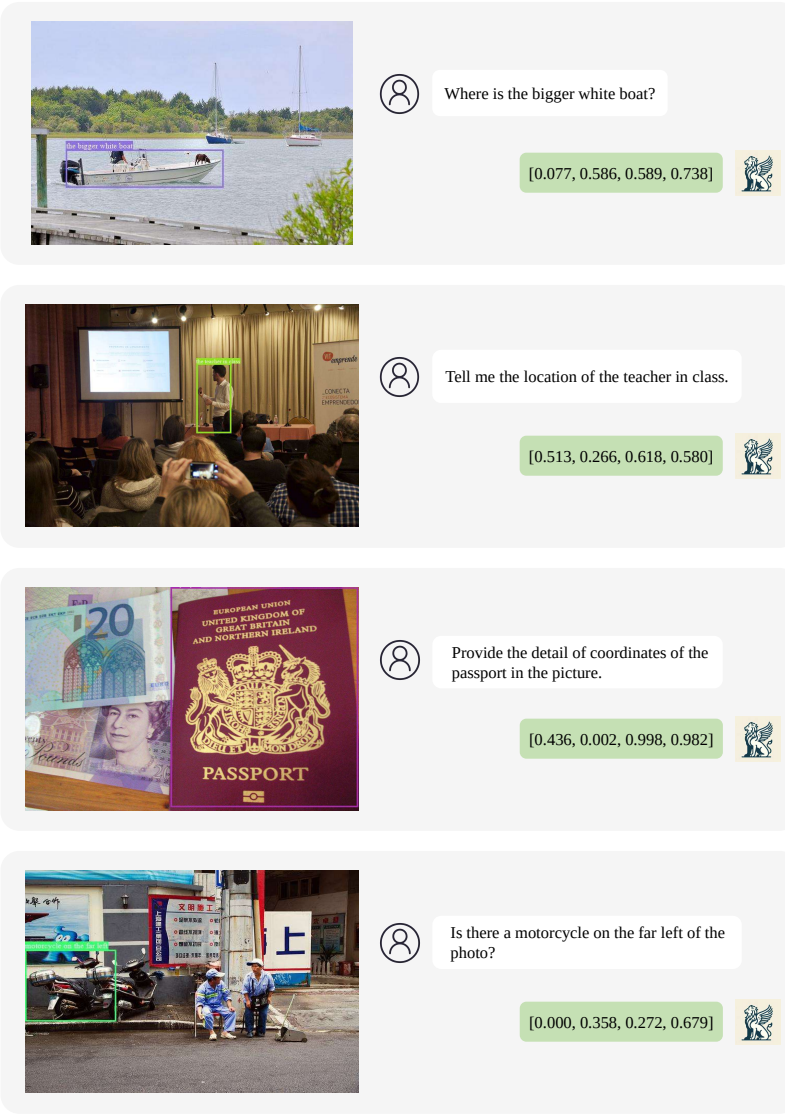
1. In this picture<image>, can you show me where <expr> are located? If it corresponds to multiple objects, please output them one by one and connect them with &.
2. In this picture<image>, identify and mark the location of <expr>. If it corresponds to multiple objects, please output them one by one and connect them with &.
3. Indicate where <expr> are in this image<image> and provide their locations. If it corresponds to multiple objects, please output them one by one and connect them with &.
4. Help me find where <expr> are in this image<image> and give me their positions. If it corresponds to multiple objects, please output them one by one and connect them with &.
5. Mark and provide the location of <expr> in this image<image>, please. If it corresponds to multiple objects, please output them one by one and connect them with &.
6. Pinpoint and provide the coordinates of <expr> in the following image<image>. If it corresponds to multiple objects, please output them one by one and connect them with &.
7. Mark where <expr> are in this photo<image> and give me their coordinates.
8. I need assistance locating <expr> in this picture<image>. Can you mark them?
9. Please detect and provide the coordinates for <expr> in this photograph<image>.
10. Can you find <expr> in this image<image> and mark their location?
11. Identify and mark the location of <expr> in this photo<image>, please.

(c) Template examples for One Category with Multi Objects scenario.

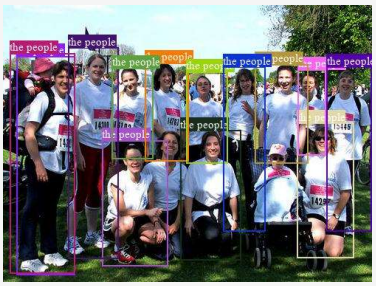
Fig. 6: Template examples for all four scenarios. The format description can also be dropped as shown in the latter half part of the examples.



(a) Demo of the Non-existing scenario.



(b) Demo of the Single Referent scenario.

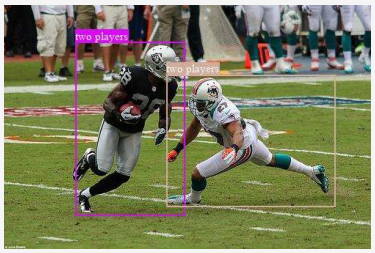


In this picture, identify and locate all the people in the front.

all the people-[0.268, 0.489, 0.446, 0.942]&all the people-[0.480, 0.453, 0.645, 0.920]&all the people-[0.797, 0.426, 0.951, 0.913]&all the people-[0.036, 0.185, 0.194, 0.975]&all the people-[0.384, 0.208, 0.502, 0.565]&all the people-[0.176, 0.154, 0.312, 0.915]&all the people-[0.799, 0.232, 0.875, 0.542]&all the people-[0.683, 0.214, 0.826, 0.571]&all the people-[0.496, 0.241, 0.609, 0.571] &all the people-[0.877, 0.223, 0.998, 0.817]&all the people-[0.295, 0.228, 0.406, 0.556]&all the people-[0.596, 0.223, 0.710, 0.817]&all the people-[0.018, 0.174, 0.194, 0.969]

I need assistance locating soldiers in this picture. Can you mark them?

soldiers-[0.426,0.049,0.647,0.949]&soldiers-[0.138, 0.051, 0.362, 0.973]&soldiers-[0.759, 0.315, 0.859, 0.795] &soldiers-[0.605, 0.225, 0.779, 0.842]



Identify and mark the location of the two players in this picture.

the two players-[0.194, 0.152, 0.493, 0.862] &the two players-[0.442, 0.286, 0.902,0.828]

(c) Demo of the One Category with Multi Objects scenario.

(d) Demo of the Multi Categories with Multi Objects scenario.

Fig. 7: Demo of Griffon at the four scenarios. **Griffon** can handle different types of inputs and output accurate localization results.