Griffon v2: Advancing Multimodal Perception with High-Resolution Scaling and Visual-Language Co-Referring

Supplementary Material

This supplementary document extends our main paper by providing more details about the dataset we have constructed, the unified representation, the implementation, the Phase Grounding results, the REG results, the result analysis on REC, and more qualitative analyses that are not included in the main paper due to the length limit. **We will release the code and data upon publication of the paper.**

1. Dataset Details

As demonstrated in the Training Pipeline of the main paper, we have collected and processed 12M data to build our pre-training and instruction-following dataset, with visual-language co-referring. In the section, We detail the data construction and the main data processing as below.

1.1. Pre-training Data Construction

To imbue the model with fine-grained perception and localization capabilities, and proficiency in visual-language co-referring, we curate a dataset of nearly 12 million localization-related instances with textual or visual reference. As illustrated in Table 8, we encompass six localization-related tasks, transforming their respective datasets into a conversational style using task-specific prompts. The data from the object counting task are utilized for visual reference, while the remaining datasets serve for textual reference. Alongside the utilization of publicly available datasets, we have derived a counting subset comprising 416K instances from OpenImages v4 and a self-collected counting dataset comprising 266K instances. The counting subset filters out images lacking categories with instance numbers exceeding 5. The self-collected counting dataset integrates data from 11 domains, as depicted in Figure 5 and Figure 6. This broad domain coverage ensures the generalization of our model without succumbing to overfitting in any particular scenario.

1.2. Instruction-following Data Construction

In contrast to the extensive data providing wide knowledge used in the pre-training phase, we leverage a smaller subset of the multi-task localization pre-training data with a greater diversity of instruction prompts, exampled in Table 9, to enhance the model's understanding of intents. Instead of manually selecting subsets from various domains, we have opted for random sampling for both the visual grounding task and object counting task. We utilize the RefCOCO series for the REG/REC and MSCOCO for object detection.

Type	Dataset Name	Vol.
REC/REG	Visual Genome	3.6M
	RefCOCO/+/g	288K
Object Detection	MSCOCO	118K
	Objects365	1.7M
Visual/Phrase Grounding	LVIS	361K
	V3Det	638K
	Flickrs30K Entities	427K
Object Counting	CA-44	22K
	OpenImages v4	416K
	Self-collected	266K
Non-existing Judging	LVIS	96K

Table 8. The statistic of the composition of pre-training data.

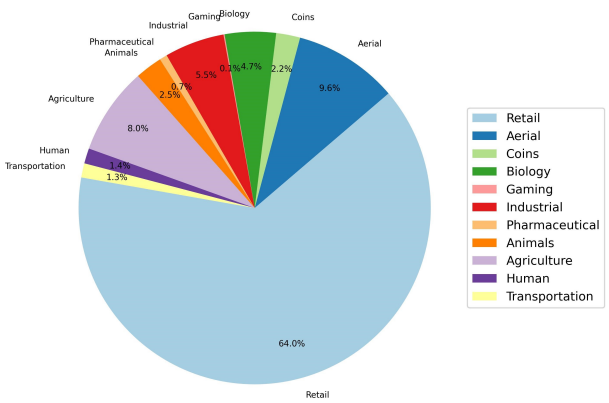


Figure 5. Data distribution of the self-collected counting data.

The data of each task realize a relative balance in terms of quantity.

1.3. Data Processing

As previously mentioned, we consolidate six tasks into a unified instruction-answer format. For REC, REG, and object detection, we adopt the processing methodology introduced by Griffon-13B [53], wherein raw annotations are directly transformed using randomly sampled instruction prompts. Regarding the visual grounding, object counting, and non-existing judging tasks, we initially convert detection-type annotations such as V3Det into instances, formulating one question for each category and enumer-

Task	Example prompts chosen from the instruction set
REC	Where is <expr> <image>? answer in [x0,y0,x1,y1] format. I am looking for the position of <expr> in <image>. Can you provide its coordinates? Help me locate and determine the coordinates of <expr> in <image>.
REG	Please generate a distinguishing description for the region <region> in the image <image>. Describe the area <region> in a unique way, given the picture <image>. Create a one-of-a-kind description for the region <region> found in the picture <image>.
Object Detection	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>.
	Examine the image<image> for any objects from the category set. Report the coordinates of each detected object. The category set includes <category set>.
	Locate and identify the objects from the category set in the image<image>. Output the coordinates of each detected object. The category set includes <category set>.
Visual Grounding	Would you kindly provide the coordinates of <expr> located in the picture <image>? Find <expr> in <image> and share its coordinates with me. In the given <image>, can you find <expr> and tell me its coordinates?
Object Counting	Detect and record the positions of objects that bear resemblance to <region> in this image.
	I want you to find all objects in the image<image> that closely match the characteristics of <region> and give me their coordinates.
	Can you identify any objects that look like <region> in this image<image>? Output their coordinates for closer inspection, analysis, and comparison.

Table 9. Examples of task templates on different types of training data. The placeholders are explained as follows: "<image>" represents the input image, "<expr>" represents the expression describing the object, "<category set>" represents the categories to be detected, and "<region>" represents the textual coordinates of the region to be asked or the locally cropped image.

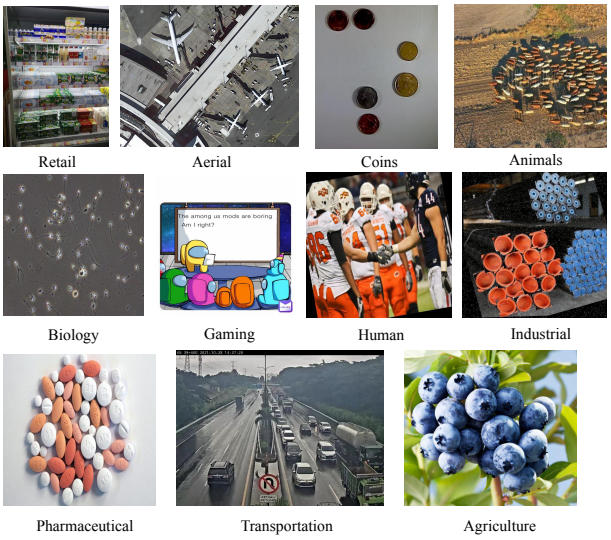


Figure 6. Data samples of the self-collected counting data.

ating all annotated categories for each image. Notably, in the case of non-existing judging data, we leverage the "neg_category_ids" annotated for each image, indicating categories unequivocally absent in that image. Subsequently, these data are integrated with randomly selected instruction templates.

2. Instruction Template Examples

In order to augment users' intent comprehension, we employ a diverse training instruction set along with a random sampling strategy. Here, we present a selection of task prompts utilized by Griffon v2 in Table 9. Each task encompasses hundreds of prompts generated by GPT-4 with specific requirements and illustrative examples. It is important to emphasize that Griffon v2 does not impose restrictions on users, allowing them the flexibility to employ their preferred natural language expressions.

3. Unified Representation with VL Co-referring

Griffon v2 employs an enhanced unified input/output representation, building upon the framework introduced in Griffon-13B, in which the input is task-specific instruction and each instance in the output is formulated as "expression-[x1, y2, x2, y2]". The representation, as preliminarily illustrated in Figure 2 from a referring perspective, has been upgraded to accommodate REG and object counting in addition to the previously supported REC, object detection, and visual/phrase grounding tasks. In Figure 7, for the REG task, we refer to the question region with normalized 3-precision coordinates, "[x1, y1, x2, y2]" uniformly, seamlessly integrating it into the instruction, with the answer describing the region. Regarding object count-

Layer	Parameters	Value
Convolution	Stride	2
	Kernel	3
	Padding	1
	inchannel	1024
	outchannel	5120
Linear	inchannel	5120
	outchannel	5120

Table 10. Hyperparameters of the designed Down-sampling Projector.

Parameter	Stage-1	Stage-2	Stage-3
batch size	256	128	128
lr	1e-3	2e-5	2e-5
lr schedule	cosine decay		
lr warmup ratio	0.03		
weight decay	0		
epoch	1		
optimizer	AdamW		
DeepSpeed stage	2		
Max Length	2048	4096	4096

Table 11. Hyperparameters of the training paradigm.

ing with visual referring, we initially employ the placeholder "<region>" in the instruction to denote the target. During training, this region is randomly selected from the bounding box annotation set of a specific category within the image, subsequently cropped out, and represented by the extracted token. The output sequence is the coordinates of detected instances concatenated with "&". During inference, it's specified by the user with screenshots or target images.

4. Implementation Details

Model parameters and training hyperparameters constitute crucial aspects of the implementation. Beyond the fundamental settings introduced in the main paper, the comprehensive lists of parameters are provided in Table 10 and Table 11. The training hyperparameters predominantly adhere to the LLaVA configuration, with the maximum length extended to 4096 to accommodate higher-resolution images and longer texts. The total training time is about 40 NVIDIA A800 days similar to 100 patch-based Monkey [25] with more data.

5. Phase Grounding and REG results

Phrase Grounding. Phrase grounding task presents a greater challenge compared to the REC task and is evaluated on Flickr30K Entities [37]. Two evaluation protocols [20] are employed, including the ANY-BOX protocol and MERGE-BOXES protocol. The ANY-BOX protocol focuses on the atomicity of each instance, while the MERGE-BOXES protocol evaluates whether the model identifies all referred objects with a merged box. Existing LVLMs are typically limited to the single referent scenario, tending to predict only one box per phase, thereby employing the MERGED-BOXES protocol. As shown in Table 12, Griffon v2 achieves state-of-the-art results in the ANY-BOX protocol and surpasses most specialists and generalists in the MERGE-BOX protocol, with more fine-grained boxes.

Type	Model	ANY	MERGED
Spec.	DDPN	-	73.5
	VisualBert	71.3	-
	MDETR	83.4	83.8
Gen.	UniTAB	-	79.6
	Ferret-13B	-	**84.8**
	Shikra-13B	-	78.4
	Griffon-13B	84.2	82.8
	Griffon v2	**84.8**	83.1

Table 12. Phrase grounding results on Flickr30K Entities[37] test set. Spec. represents specialists, while Gen. represents generalists.

Type	Model	CIDEr	Meteor
Spec.	SLR[52]	66.2	**15.9**
	ASM[46]	41.9	13.6
	Grit[47]	71.6	15.2
Gen.	KOSMOS-2[36]	60.3	12.2
	Griffon v2	**72.5**	12.1

Table 13. REG resuls on RefCOCOg [34].

REG. REG aims to generate concise descriptions for specified objects based on their region locations. We input textual coordinates for object referring and tests on the RefCOCOg [34] val set. As illustrated in Table 13, in contrast to KOSMOS-2 [36], which uses learnable embeddings for referring, we achieve superior performance in CIDEr, concentrating on the semantic similarity, while the Meteor focuses more on the accuracy of wording, making it less suitable for the open-ended description generation of LLMs.

6. More Qualitative Results

To further demonstrate the performance of our Griffon v2, we provide more visualization results on object detection (Figure 8), visual grounding (Figure 9), and object counting (Figure 10), compared to some expert models.

REC/Visual Grounding/Non-existing Judging

Instruction**:** Where is < expression, e.g., the man wearing a light brown coat and scarf in the middle > in the image?

Answer: the man wearing a light brown coat and scarf in the middle-[0.468, 0.409, 0.515, 0.647]

Object detection

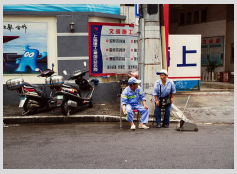
Instruction: Examine the image for any objects from the category set, which includes < e.g., car, skateboard, bus, person, motorcycle, tie, sculpture, backpack, dog and surfboard ... >.

Answer: car-[0.799, 0.436, 1.000, 0.758] & motorcycle-[0.526, 0.513, 0.553, 0.605]& bus-[0.577, 0.146, 0.846, 0.687] & bus-[0.308, 0.259, 0.496, 0.641] & person-[0.237, 0.432, 0.327, 0.727] & person-[0.061, 0.422, 0.127, 0.706] & person-[0.516, 0.442, 0.564, 0.574] & car-[0.557, 0.472, 0.58, 0.583] & car-[0.497, 0.441, 0.532, 0.476] & backpack-[0.251, 0.547, 0.276, 0.623]…

REG

Instruction: For the given image, can you provide a unique description of the < region, e.g., is [0.000, 0.358, 0.272, 0.679] >.

Answer: Motorcycle on the far left.



Object Counting

Instruction: For the given image, can you provide a unique description of the < region, e.g.,  >.

Answer: [0.000, 0.358, 0.272, 0.679] & [0.261, 0.360, 0.521, 0.675]

Figure 7. Examples of unified representation for each task. Object counting task utilizes visual referring without category information and directly outputs the object coordinates and corresponding number.

Figure 8. Comparison with Grounding DINO in object detection. Griffon v2 demonstrates a reduced occurrence of both missed detections (col. 2) and false positives (col. 1,3).

Figure 9. Comparison with Grounding DINO in visual grounding. Griffon v2 and Grounding DINO exhibit comparable visual grounding capabilities.

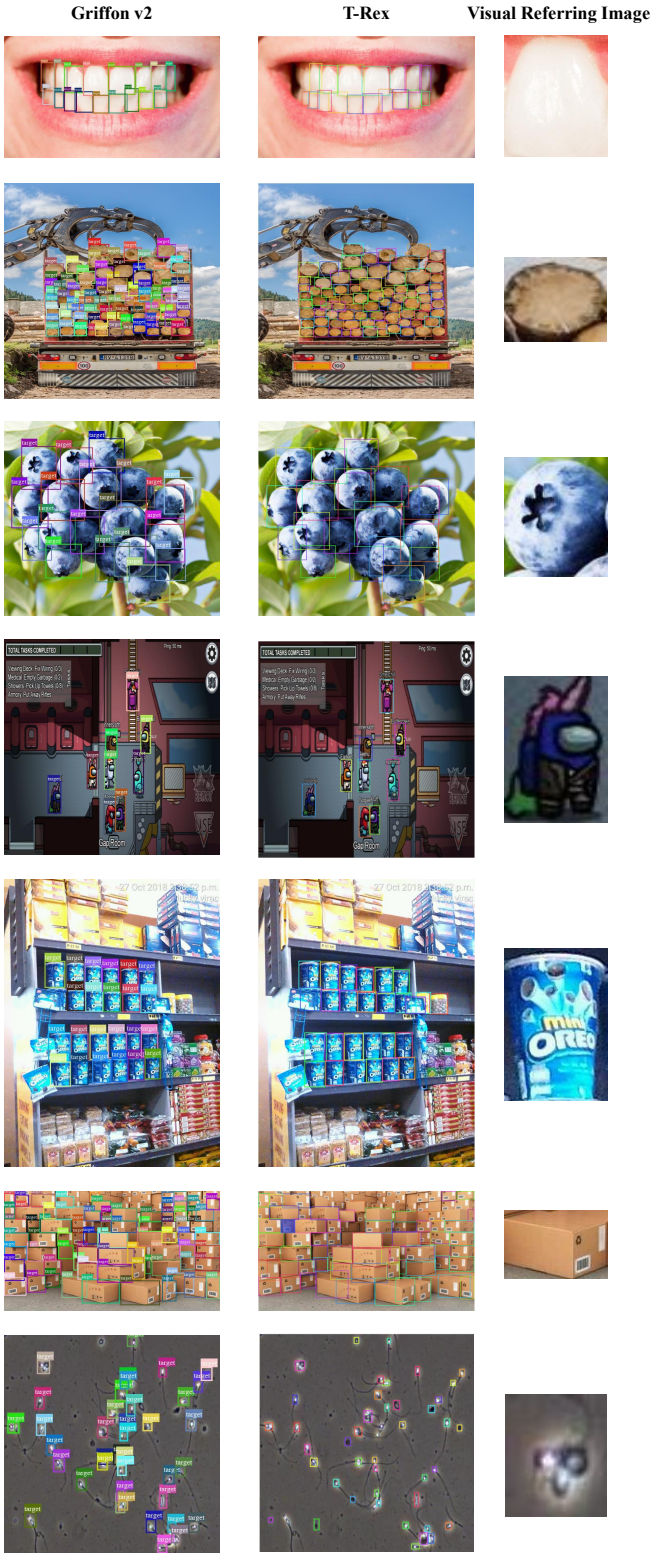


Figure 10. Comparison with T-Rex in object counting. Griffon v2 achieves counting proficiency with visual reference comparable to that of the expert model T-Rex.