Acknowledgements

We thank Alan Yuille, Eric Mintun, Hanzi Mao, and Alexander Kirillov for helpful discussions. This work was supported in part through the NYU IT High Performance Computing resources, services, and staff expertise. This research is also supported by Intel, Google TRC program, the Google Cloud Research Credits program with the award GCP19980904, the OpenAI API Researcher Access Program, and an Amazon Research Award Fall 2023.

A. Implementation Details

A.1. Data Curation for VQA LLM

Since our VQA LLM will now work with the VWM that has searched targets, we need to perform additional instruction tuning to train the VQA LLM. We describe the training data as follows.

Negative data for target objects reasoning (100k) The VQA LLM must first identify the target objects that are 1) required to answer the question, and 2) missing or not clear enough in the initial global image features. To facilitate this, we construct (image, question, answer) data where the question pertains to one or two objects not present in the image. Additionally, we construct questions about details of certain objects, deliberately made too small to be captured by the CLIP encoder. This is achieved by choosing objects with bounding box sizes smaller than 20×20. The appropriate response to such questions is a straightforward acknowledgment that the question cannot be answered, along with a clear enumeration of all the additional target objects required. We construct 100k data on COCO2017 [25] with questions generated by GPT-3.5.

VQA data (167k) This data consists of three parts: GQA data (70k) from [15], VQA data focused on object attributes (51k), and VQA data focused on spatial relationship (46k). In the GQA subset, we utilize the original dataset's GT annotations about specific objects mentioned in the questions. We select a portion of this data, treating the mentioned objects as search targets in the VWM during training. Additionally, we rephrase the short answers in GQA into full sentences using GPT-3.5. For the object attribute data, we utilize the VAW [38] data, transforming them into question-answer pairs in a standard format that inquire about certain object attributes, and consider these objects as search targets. Regarding the spatial relationship data, we use the COCO2017 dataset to generate questions about the relative spatial positioning of two objects within an image, treating these two objects as the search targets.

For the GQA part of the 167k VQA data, our target is to find questions where the annotated objects mentioned in the question are critical to correctly answer the question. Therefore, we first evaluate InstructBLIP on GQA questions with annotated objects in the question and only keep questions that can be correctly answered by it. Then we use the image inpainting model LaMa [45] to erase all the mentioned objects in the corresponding images and re-evaluate the InstructBLIP model with the modified images and only keep the questions that can not be correctly answered after this modification. Through these process, we have selected a subset of GQA questions where the annotated objects are important and we use them to construct our VQA data.

For the VAW object attribution part of the 167k VQA data, we create open-ended questions and binary questions about objects' attributes. For the open-ended questions, we consider attribute types including 'color', 'material', 'hair color', 'pattern', 'face expression', 'pose', 'activity', 'opaqeness', and 'texture'. For the binary questions, we additionally include attribute types 'state' and 'optical property'. All these attribute types follow the definition from the VAW dataset. We use fixed templates for both types of questions. For the open-ended questions, the question template is "What is the [attribute type] of the [object name]?" and for the binary questions, the template is "Is the [attribute type] of the [object name] [attribute value]?" The corresponding answer to open-ended questions is "The [attribute type] of [object name] is [attribute value]." and the answer to binary questions is "Yes/No, the [attribute type] of [object name] is/is not [attribute value]." Besides, we use the same strategy as the GQA part to filter the questions with the InstructBLIP model and the object erasing process.

LLaVA Instruction Tuning (120k) To maintain the general multimodal question answering and instruction following capabilities, we also include the LLaVA-80K instruction tuning data, of which the image sources are also COCO. Additionally, we identify object entities in the questions that match with COCO categories and have box annotations, creating an additional set of 40k data.

For this additional 40K data, we first extract all the noun phrases in the questions/instructions of the LLaVA-80K data. Then we choose noun phrases that are matched with the object category names defined by COCO. Note that we augment some original category names with more common synonyms (*e.g.* add 'man' and 'woman' for the 'person' category). Then we check whether there exist annotated instances of this category in this corresponding image. If so, we keep this sample and use the annotated instances together with their bounding box information as the target objects for our training.

A.2. Data Curation for Visual Search Model

The training data of our visual search model includes two parts. The first part is the detection and segmentation data and the second part is the VQA data which includes possible locations QAs and LLaVA-80K instruction tuning data.

The COCO-Stuff [3], LVIS-PACO part [40], refCOCO(+/g) [18, 33], and refCLEF [18] datasets are used as both detection and segmentation data. Objects365 v2 [42] and GoldG [17] datasets are only used as detection data.

The textual contextual cues of our visual search model are in the form of possible location expressions about the target objects. So we construct (image, question, answer) pairs about objects' possible locations. We randomly sample a subset of images from COCO2017. For each image, we randomly sample 2 objects that are absent in this image but appear in the five images that are most similar to it (based on CLIP embedding). The question is always asking "What is the most likely position of [object]". Then we provide the image information (5 captions and a list of existing objects) to GPT-3.5 and ask it to provide the possible location of the absent objects and use its response as the answer. The complete prompt is shown in Table 6.

A.3. Model Training

For the VQA LLM, we use the Vicuna-7b-1.3 [61] as the language model. Following the common practice of current MLLMs, the

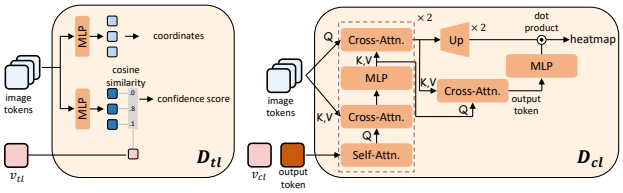

Figure 6. Detailed structure of the *target* localization decoder D_{tl}, and the *search cue* localization decoder D_{cl}.

training process has two stages, a feature alignment stage and an instruction tuning stage. For the alignment stage, the linear layer projection module and the resampler projection module are separately trained along with a frozen language model and vision encoder on the subset of image-text pairs from the 558K LAION-CC-SBU subset used in LLaVA. For the instruction tuning stage, we use the constructed 387k data to train the LLM along with the projection modules, and only the vision encoder is frozen in this stage.

For the alignment stage, the linear projection module and the resampler projection module are separately trained with batch size 256. The linear projection module is trained for 1 epoch with learning rate 10^{-3} and the resampler module is trained for 5 epochs with learning rate 2×10^{-4}. For the instruction tuning stage, the model is trained for 3 epochs with learning rate 2×10^{-5} and batch size 128. To reduce the computational cost, during the training, we use the linear projection module to project the search target's feature only when there is just one object, otherwise, we use the linear projection module to project the global image feature and use the resampler for the searched objects. The exact input sequence for the LLM constructed from the VWM is:

```
<Image>
Additional visual information to focus
on:
{Target Object 1's Name} <Object> at
location [x1, y1, x2, y2];
{Target Object 2's Name} <Object> at
location [x1, y1, x2, y2];
...
Question
```

Here `<Image>` is the feature tokens of the image and `<Object>` is the feature tokens of the target object stored in the VWM.

For the visual search model, we adopt the LLaVA-7B-v1.1 as the MLLM. We use the OWL-ViT-B-16's vision encoder as the image backbone. The D_{cl} resembles the mask decoder in SAM [19] , and the D_{tl} is implemented with two linear heads, one for coordinate prediction and the other for confidence score prediction. The detailed structure of these two modules is shown in Fig 6. The D_{cl} is trained with segmentation loss which consists of binary cross-entropy loss and DICE loss. During inference, the logits output from the D_{cl} is used as the search cue heatmap. The D_{tl} is trained with set prediction loss similar to DETR [4] with focal loss [26] for coordinates regression. The whole model is trained for 100K steps with batch size 64 and learning rate 10^{-4}. The sampling ratio of general detection/segmentation datasets (Objects365 v2, COCO-Stuff, LVIS-PACO), referring detection/segmentation datasets (refCOCO, refCOCO+, refCOCOg, refCLEF, GoldG), and VQA data is 15:8:15.

During training, the pre-trained MLLM is trained with LoRA [13] with the word embeddings layer being trainable. The image encoder of the localization module and the coordinates MLP in D_{tl} are frozen. The confidence score MLP and the D_{cl} are trainable.

A.4. Visual Search Process

When entering the next level during the visual search process, we use a simple strategy and recursively divide the image into 4 non-overlapping equal-sized patches.[1] In order to maintain a square-like aspect ratio for each patch during the search, we adjust our division strategy based on the image's orientation. For landscape images (*i.e.*, where the width is greater than twice the height), we divide the image vertically. Conversely, for portrait images (*i.e.*, where height exceeds twice the width), we divide it horizontally. In all other cases, we split the image both horizontally and vertically. This approach to patching is depicted in Fig 7.

During the search process, we first set a relatively higher threshold and terminate when a target is located with a confidence score higher than this threshold. If the whole search process is completed without any target being found, we adjust the threshold to a lower value and find the target with the highest confidence score during the whole search process and accept it if its confidence score passes the adjusted threshold. In the scenario where the visual search is needed to find a certain target, finding all instances in a high-resolution image requires scanning the whole image exhaustively, making the search strategy less meaningful. Therefore, our current visual search process focuses on finding a single target object instead of locating all targets exhaustively. However, if we successfully locate multiple targets directly on the global image without the need for further search, we add all of them to the VWM. When evaluated on V^*Bench, the target with the highest confidence score is accepted as the searched target if no target passes the threshold during the search process. In our implementation, the higher threshold is set to 0.5 and the lower threshold is set to 0.3. And the threshold δ is set to $max(3.0, 6.0 \times 0.7^l)$, where l is the image sub-dividing level.

To let the MLLM in the visual search model generate the search cue heatmap corresponding to the contextual cue, we extract the noun phrases in the textual contextual cue which is a possible location expression, and prompt the MLLM to locate the phrase and output the heatmap corresponding to this contextual cue.

B. V^*Bench Examples and Two Special Subsets

Some examples of our proposed V^*Bench are shown in Fig 8. Besides the regular attribute recognition and spatial relationship reasoning sub-tasks, we additionally collect two special subsets and add them to our V^*Bench for exploratory study. The first subset contains 30 VQA samples which require the model to recognize and understand textual characters or digits on certain objects in

[1] A corner case is that this simple strategy might fail when targets are located at the boundaries of patches. One can use overlapping patches or variable-sized patches based on the heatmap distribution, if necessary.

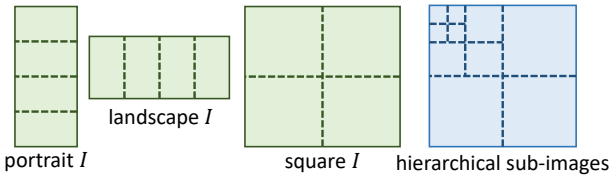

Figure 7. Images are recursively divided into four patches based on their aspect ratio. Landscape images are divided vertically. Portrait images are divided horizontally.

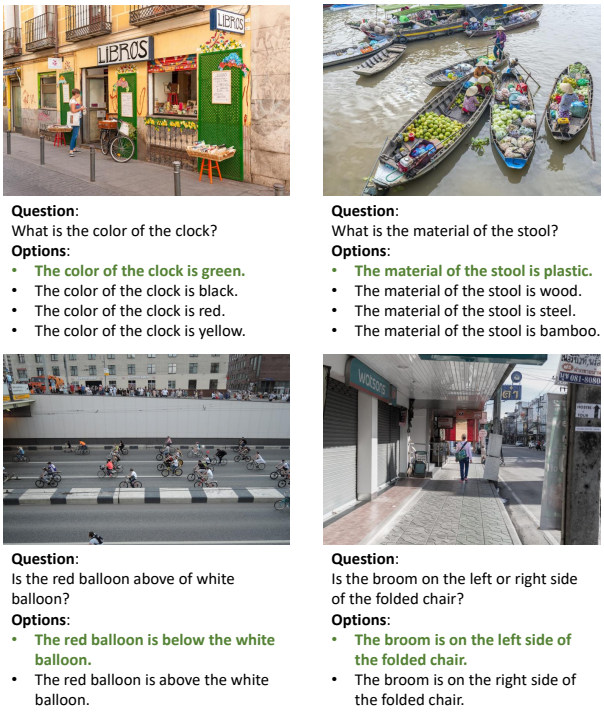

Figure 8. Examples of the V^* Benchmark. The top row belongs to the attribute recognition task while the bottom row belongs to the spatial relationship reasoning task. The correct option is in green.

the image and we denote this subset as OCR. To better expose the problem of current MLLMs and even the leading multimodal system GPT-4V, we collect 17 samples on which GPT-4V would fail but our simple model with visual search mechanism success and denote them as GPT-4V-hard. We also evaluate LLaVA-1.5, MM-REACT, GPT-4V, and SEAL on these two subsets, and the results are shown in Fig 10. As the MM-React system relies on the external OCR detection model and GPT-4V also likely has an OCR module, their performance on the OCR task is decent. However, for MM-React, the external OCR model detects all the texts in the image and provides them to the LLM in a bottom-up manner. Therefore, it is easy to choose the correct option merely based on the detected texts in the scene when the image only contains a few texts. When there are multiple text contents in the image or the question needs the model to fully understand the context of the

Question:
What is the number on that blue board?
Options:
- **The number on that blue board is 2050.**
- **The number on that blue board is 2013.**
- The number on that blue board is 2030.
- The number on that blue board is 2023.

Figure 9. An example on which the MM-React system with an external OCR detection model fails. The correct option is in green and the option chosen by MM-React is in red.

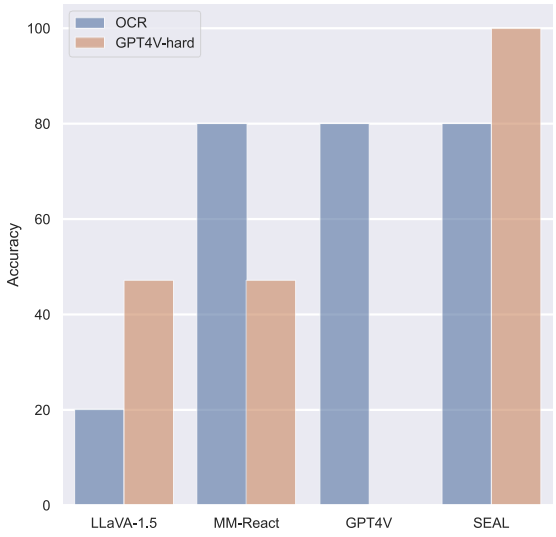

Figure 10. Comparison between SEAL and top MLLM systems on the OCR and GPT4V-hard sub-tasks.

text or its location, the system would fail. An example is shown in Fig 9.

We also provide the image sources of examples in Figure 2 of the main text below:

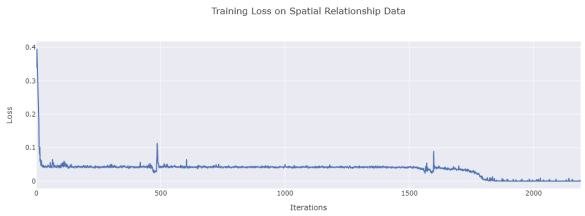

Figure 11. The loss curve of training on spatial relationship VQA data. The "grokking" happens after a certain number of optimization steps.

Row-1, Column-1: Web Source
Row-1, Column-2: Japanese animated film *Children Who Chase Lost Voices* [43]
Row-1, Column-3 Web Source
Row-1, Column-4: Web Source
Row-2, Column-1: SR-RAW dataset [60]
Row-2, Column-2: Personal Photo in Orlando Disney
Row-2, Column-3: Web Source
Row-2, Column-4: SR-RAW dataset [60]
Row-3, Column-1: SR-RAW dataset [60]
Row-3, Column-2: Web Source
Row-3, Column-3: SAM dataset [19]
Row-3, Column-4: SR-RAW dataset [60]
And the image source of the example in Figure 1 of the main text is the Japanese animated film *Weathering with You* [44]

C. Learning Spatial Relationship from Coordinates

We provide the numerical coordinates of the search targets to the VQA LLM as the spatial information about the searched targets. We find that though it seems intuitive and simple to recognize the relative spatial relationship with the coordinates, it is not trivial for the VQA LLM to understand the numerical coordinates and understand the spatial relationship between search targets by comparing their coordinates. We conduct additional experiments to train the VQA LLM only on the constructed 46K spatial relationship related VQA data and the loss curve is shown in Fig11. We can see that instead of gradually decreasing, the loss suddenly drops to 0 after a certain number of optimization steps, suggesting that the model has learned to correctly compare the numerical coordinates for determining spatial relationships. As this kind of grokking needs a certain amount of optimization steps, one might need to improve the ratio of the spatial relationship related VQA data when mixing it with a larger amount of general multimodal instruction tuning data (*e.g*. training data for LLaVA-1.5 [27]) to ensure the model can correctly understand the numerical coordinates.

”””
You are an AI visual assistant that can analyze a single image. You receive five captions, each describing the same image you are observing. And you will also be given a list of objects which are in the image.
Captions:
{}
Objects: {}
You will be given two target objects, and your task is to use your common sense knowledge and the information about the image to describe the most possible location of the given target objects in the image. Your answer must be a short expression describing the possible location referring to some other existing entities in the image. Answer concisely in less than 10 words.
Examples:
Target Object: bird
Most Possible Location: in the sky
Target Object: flag
Most Possible Location: on the roof of the building
###
Target Object 1: {}
Target Object 2: {}
Output Format
Most Possible Location of Target Object 1:
Most Possible Location of Target Object 2:
”””

Table 6. Prompt for GPT-3.5 to generate possible location expression of an object which is absent in the image.

”””
You are an AI visual assistant that can analyze a single image. You receive five captions, each describing the same image you are observing.
Captions:
{}
You will be given 1 target object and you need to imagine certain attributes of it using your imagination and assume it is in the image.
You need to ask 2 questions about the provided target object for each of the following question types:
Type 1: certain object attributes
Type 2: relative positions between objects
Type 3: interactions between objects
Your question could also involve other objects or provided information about the image. Do not ask questions about the existence of the objects. Your questions should not contain any uncertainty about the presence of the target object. Make sure each question involves the target object provided below. Ask short and natural questions of less than 20 words.
Target Object:
{}
###
Output format:
Type 1
Question 1:
Question 2:
Type 2
Question 1:
Question 2:
Type 3
Question 1:
Question 2:
”””

Table 7. Prompt for GPT-3.5 to generate question-answer pairs about one target object which is absent in the image.

”””
You are an AI visual assistant that can analyze a single image. You receive five captions, each describing the same image you are observing.
Captions:
{}
You will be given 2 target objects and you need to imagine certain attributes of them using your imagination and assume they are in the image.
You need to ask 2 questions around the provided target objects for each of the following question types:
Type 1: direct attribute questions about the two target objects or simple logical comparison of the attributes
Type 2: positions of the 2 target objects or relative positions between them or other objects
Type 3: interactions between the two target objects while other objects might be used as reference
Target Objects:
{}
Your question could also involve other objects or provided information about the image. Do not ask questions about the existence of the objects. Your questions should not contain any uncertainty about the presence of the target objects. Ask short and natural questions of less than 20 words. Each question must contain both the {} and the {}.
###
Output format:
Type 1
Question 1:
Question 2:
Type 2
Question 1:
Question 2:
Type 3
Question 1:
Question 2:
”””

Table 8. Prompt for GPT-3.5 to generate question-answer pairs about two target objects which are absent in the image.

”””
Assume there is an object of type {} in an image, you need to come up with two visual questions asking about the visual details of the {}. Make sure the questions are so detailed that it is very hard to answer if the {} is very small in the image. Do not ask about the existence of the object.
Examples:
Object: shirt; Question: what is the text printed on the shirt?
Object: cup; Question: Does the cup have a dotted pattern?
Ask 2 reasonable questions about the {} and each question should be less than 20 words.
Object: {}
Question 1:
Question 2:
”””

Table 9. Prompt for GPT-3.5 to generate question-answer pairs about the details of a small object in the image.