

CIVET: Systematic Evaluation of Understanding in VLMs

Massimo Rizzoli^{1 †}, Simone Alghisi^{1 †}, Olha Khomyn¹, Gabriel Roccabruna^{2 ‡},
Seyed Mahed Mousavi¹, Giuseppe Riccardi¹

¹ Signals and Interactive Systems Lab, University of Trento, Italy

² Amazon

{massimo.rizzoli, s.alghisi, giuseppe.riccardi}@unitn.it

Abstract

While Vision-Language Models (VLMs) have achieved competitive performance in various tasks, their comprehension of the underlying structure and semantics of a scene remains understudied. To investigate the understanding of VLMs, we study their capability regarding object properties and relations in a controlled and interpretable manner. To this scope, we introduce CIVET¹, a novel and extensible framework for systematic evaluation via controlled stimuli. CIVET addresses the lack of standardized systematic evaluation for assessing VLMs' understanding, enabling researchers to test hypotheses with statistical rigor. With CIVET, we evaluate five state-of-the-art VLMs on exhaustive sets of stimuli, free from annotation noise, dataset-specific biases, and uncontrolled scene complexity. Our findings reveal that 1) current VLMs can accurately recognize only a limited set of basic object properties; 2) their performance heavily depends on the position of the object in the scene; 3) they struggle to understand basic relations among objects. Furthermore, a comparative evaluation with human annotators reveals that VLMs still fall short of achieving human-level accuracy.

1 Introduction

Recent advancements have shown that Vision-Language Models (VLMs) have achieved competitive performance on several vision-language tasks. However, data used to train and evaluate these models is limited in size, and may suffer from annotation errors (Schuhmann et al., 2021), label imbalance (Acharya et al., 2019), visual scene biases (e.g.

[†] Equal contribution.

[‡] Work done while at University of Trento, prior to joining Amazon.

¹We release all the materials of CIVET and encourage the community to extend this framework and its components for different evaluation and training settings: <https://github.com/sislab-unitn/CIVET>.

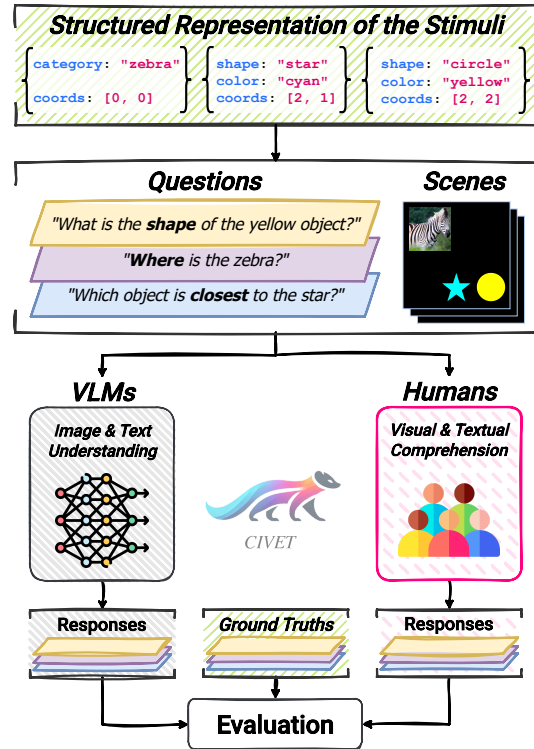



Figure 1: CIVET framework enables a systematic evaluation of VLMs. The framework includes customizable Stimuli, and their Structured Representation; deterministic generation of Stimuli instances (Questions and corresponding visual Scenes); and comparative assessment of VLMs and Humans understanding.

objects often in the center (Kirillov et al., 2023)), and scene complexity (e.g. number and type of objects (Lin et al., 2014)). This may affect evaluation results and the outcome of the learning process, hindering VLMs' generalization capabilities and providing data- or task-specific performance. Indeed, studies on object classification with randomly augmented (Roth et al., 2023) and spurious non-visual descriptions (Esfandiarpour et al., 2024) suggest that VLMs may exploit dataset biases and statistical shortcuts rather than understanding the underlying structure and semantics of the scene.


Different from previous works that focus on solving specific tasks using limited data (Thrush et al., 2022; Paiss et al., 2023; Fu et al., 2023; Chen et al., 2024), we conduct a broader and exhaustive investigation into VLMs’ understanding. Specifically, we frame our study on three research questions: **RQ1: Can VLMs accurately recognize basic object properties?** Recognizing object properties is essential to distinguishing between similar elements of a scene and generalizing to previously unseen objects using known attributes. **RQ2: Is their performance robust to variations in object positioning?** Although a model may recognize different object properties, understanding requires consistent performance even when other variables change, such as object position. **RQ3: Can VLMs identify basic relations among objects?** To support reasoning and understanding of visual scenes, models must go beyond recognizing individual objects and properties, and capture how objects relate to each other and interact in the world.

Answering these questions rigorously requires a systematic evaluation that includes carefully designed and controlled stimuli. However, available evaluation frameworks based on generative models (Peng et al., 2024) may suffer from hallucinations and lack the scalability required for systematic experimentation. Meanwhile, earlier deterministic approaches (Andreas et al., 2016; Johnson et al., 2017) — while more controllable — were not designed to ensure uniform distributions of visual scenes, precluding systematic evaluation. Additionally, since these stimuli may have been included in VLMs’ training data, their reliability as evaluation tools is further compromised.

To address these issues, we introduce CIVET , a novel and extensible framework for systematic evaluation via controlled stimuli. Unlike previous work, CIVET allows systematic evaluation with statistical guarantees, free from external confounding factors, achieved with precise control over the content of visual and textual stimuli and deterministic generation. Using CIVET, we systematically evaluate the performance of five state-of-the-art VLMs in recognizing properties and relations of elementary objects by generating tailored sets of stimuli, balanced in terms of position, property values, and labels. Since these elementary objects might overestimate performance, we also evaluate the VLMs on real-world objects (i.e., objects from MS COCO (Lin et al., 2014)). Finally, we conduct a study with human annotators and compare their

performance to that of the VLMs. Figure 1 shows an overview of the CIVET framework.

In summary, the main contributions of this paper are:

- CIVET , a novel and extensible framework to systematically evaluate VLMs’ understanding;
- Exhaustive evaluation of the understanding of five state-of-the-art VLMs in recognizing object properties and relations;
- Comparative evaluation of VLMs and humans’ performance on the same set of stimuli.

2 Literature Review

Several works have evaluated VLMs across a wide range of tasks, including VQA (Goyal et al., 2017; Yue et al., 2024; Chen et al., 2024), reasoning with external knowledge (Fu et al., 2022, 2023), counting (Acharya et al., 2019; Paiss et al., 2023), and understanding object relations (Krishna et al., 2017; Thrush et al., 2022; Yuksekgonul et al., 2022). These works focus on specific tasks without assessing VLMs’ understanding. Indeed, their evaluation is based on real-world visual scenes paired with human-annotated ground truths, which often suffer from issues such as label imbalance (Acharya et al., 2019), positional biases (e.g., relevant objects appearing centrally (Kirillov et al., 2023)), and unknown or uncontrolled scene complexity, including occlusions, distractors, and ill-posed questions. To address some of these limitations, other frameworks have been proposed to assess the performance on visual-language tasks in a controlled setting. While SPEC (Peng et al., 2024) proposed to study VLMs’ understanding of objects’ properties and relations by generating realistic visual scenes, it uses diffusion models, which are known to suffer from hallucination (Aithal et al., 2024; Kim et al., 2025). Unlike CIVET, SPEC provides no guarantee of being free of annotation error, hindering the interpretability of results. On the other hand, earlier deterministic approaches, such as SHAPES (Andreas et al., 2016) and CLEVR (Johnson et al., 2017), eliminate annotation errors but do not ensure uniform distributions of visual scenes, precluding systematic evaluation. Additionally, since the (pre-)training data of VLMs is often undisclosed, they may have been partially included in the pretraining. This makes them unsuitable to answer our research questions, as good performance may not

reflect VLMs’ scene understanding. Furthermore, the 3D scenes of CLEVR include additional complexity (e.g., occlusions, reflections, and shadows) that may confound our evaluation.

3 CIVET Framework

We introduce CIVET, a framework designed to address the lack of standardized, systematic evaluation to assess VLMs’ understanding. CIVET enables systematic investigation of open research questions in the field by leveraging an exhaustive set of controlled visual scenes and natural language inputs. We formalize the framework to make it extensible, allowing researchers to adapt it to diverse evaluation objectives. Following the definitions given in Ontology (Rettler and Bailey, 2024), an *object* is defined as an instance of an *entity*² and is characterized by a given set of properties. A *property* is a characteristic of an object, such as its shape or color. Moreover, the way objects stand to each other is called a *relation*, such as their relative position (e.g., on top, or in front), or relative size (e.g., smaller, or larger). We define a world as a set of objects and their relations, where objects are characterized by property-value pairs. Each world is a structured representation of a stimulus, which can be used to generate a scene (i.e., a visual representation of the world), and a set of natural language questions about its objects, properties, and relations. To evaluate the understanding of an aspect of the scene, we fix that aspect and marginalize over all combinations of the remaining variables. For example, to assess recognition of a particular shape like a star, we consider all scenes that contain a star, and marginalize over variations in color and position. This allows us to isolate the model’s understanding of shape by averaging out the influence of other factors.

4 Experimental Settings

To rigorously answer our research questions, we need tailored sets of stimuli that are free from annotation error and visual biases, and are balanced in terms of position, property values, and labels. Using CIVET, we systematically evaluate the understanding of five VLMs by generating these controlled sets of stimuli.

²Where an entity is “*independent, separate, or self-contained existence*”, from the Merriam-Webster dictionary.

Experiment	Question Template
Properties	What is the $\langle \text{property} \rangle$ of the object?
Absolute Position	Where is the $\langle \text{sheen} \rangle$ $\langle \text{color} \rangle$ $\langle \text{shape} \rangle$?
Relative Position	Where is the $\langle \text{shape}_1 \rangle$ positioned with respect to the $\langle \text{shape}_2 \rangle$?
Relative Distance	What is the closest object to the $\langle \text{shape} \rangle$?
Relative Size	What is the size of the $\langle \text{shape}_1 \rangle$ with respect to the $\langle \text{shape}_2 \rangle$?

Table 1: Natural language question templates used in each experiment. We make the questions closed-ended by appending “Choose from [$\langle \text{options} \rangle$]”, and replacing $\langle \text{options} \rangle$ with the corresponding answer options.

4.1 Settings

We design five settings to address our research questions: **Single Object** and **Single Object w. COCO** for the recognition of object properties (RQ1) and independence to object position (RQ2); and **Relative Position**, **Relative Size**, and **Relative Distance** for the recognition of relations among objects (RQ3). In all settings, the task requires answering closed-ended questions (Table 1) about a scene. Each scene is a 9×9 grid that corresponds to the visual representation of the world (containing its set of objects). For each question, we provide the set of possible answer options by appending to the question “Choose from [$\langle \text{options} \rangle$]” (where $\langle \text{options} \rangle$ is a comma-separated list of all possible answer options to the question). We limit the order bias by shuffling the options so that each possible order appears uniformly in the textual input. Additionally, as models tended to respond with open answers, we condition the models prepending the instruction “Answer with as few words as possible.”. We discuss this solution in detail in Appendix A.2.

Single Object (RQ1, RQ2) To answer our questions about the model’s ability to recognize object properties (i.e., *shape*, *color*, *sheen*) and its position w.r.t. the background (i.e., *absolute position*), we consider worlds containing exactly a single object, eliminating other confounding factors. As objects, we consider all the combinations of 4 shapes (*square*, *circle*, *triangle*, *star*), 6 colors (*red*, *green*, *blue*, *cyan*, *magenta*, *yellow*), and 3 values for sheen (either *no sheen*, or *matte* or *glossy sheen*). Then, for each object we create 81 different visual scenes by placing it in each pos-

Model	Category	Position
<i>Random Baseline</i>	33	11
<i>LLaVA-NeXT 7B</i>	91	37
<i>LLaVA-NeXT 13B</i>	80	51
<i>Molmo-O 7B</i>	70	53
<i>Qwen2-VL 7B</i>	97	52
<i>CLIP</i>	67	15

Table 2: Accuracy (%) of each model when considering visual scenes containing a *Since Object* extracted from COCO (among *zebra*, *giraffe*, and *elephant*) and querying about their *category* and *absolute position* (w.r.t. the background). Results are based on 1344×1344 images.

sible cell of our 9×9 grid (for a total of 5,832 scenes). When querying about the absolute position of the object, we divide the scene into 9 equal sections of 3×3 cells. Then, we assign to each section (top-to-bottom, left-to-right) a unique label (i.e., *top left*, *top center*, *top right*, *center left*, *center*, *center right*, *bottom left*, *bottom center*, *bottom right*) and use them as ground truth.

Single Object w. COCO (RQ1, RQ2) Since elementary objects might overestimate VLM performance, we complement synthetic worlds with real-world objects extracted from COCO images (Lin et al., 2014). For each bounding box, we use CLIP to classify the category of the contained object. We select the three object categories with the best performance, namely *giraffe*, *elephant*, *zebra*. For each category, we analyze the 10 objects with the highest similarity (dot product) to their category and manually select the best one (i.e., containing a single, non-occluded instance of the category). Similarly to its synthetic counterpart, we answer our questions about the model’s capability to recognize the object properties (i.e., *category*) and its position w.r.t. the background (i.e., *absolute position*) by designing a set of stimuli containing a single object. We consider all combinations of 3 categories (*giraffe*, *elephant*, *zebra*) and cell placement in our 9×9 grid (for a total of 243 scenes).

Relative Position (RQ3) To understand whether VLMs can identify basic relations among multiple objects, we first assess their performance on *relative position*, which concerns the placements of objects in the scene to one another. Based on the results of the previous experiments, we select two *yellow* objects with different shapes as discriminants (i.e., *yellow star* and *yellow triangle*). We

Model	Shape	Color	Sheen	Position
<i>Random Baseline</i>	25	17	50	11
<i>LLaVA-NeXT 7B</i>	98	88	50	42
<i>LLaVA-NeXT 13B</i>	97	76	64	47
<i>Molmo-O 7B</i>	100	98	59	62
<i>Qwen2-VL 7B</i>	99	99	60	61
<i>CLIP</i>	95	95	49	14

Table 3: Accuracy (%) of each model when considering visual scenes containing a *Single Object* and querying about its *shape*, *color*, *sheen*, and *absolute position* (w.r.t. the background). Results are based on 672×672 images.

then construct a set of stimuli by placing the two objects in all combinations of cells (for a total of 6,480 visual scenes) and query about the object’s relative position. As ground truths, we consider 8 possible answers: 4 where the objects are on the same row or column (*directly above*, *directly left*, *directly right*, *directly below*), and 4 where the object are offset on both row and column (*above left*, *above right*, *bottom left*, *bottom right*)

Relative Size (RQ3) As a second type of relation, we assess the performance of VLMs in recognizing the *relative size* of an object w.r.t. another. In this setting, we consider four *yellow* objects with different shape and size³ combinations (i.e., *regular yellow star*, *small yellow star*, *regular yellow triangle*, and *small yellow triangle*). For each combination of two objects, we then generate one visual scene for each pair of cells (for a total of 25,920).

Relative Distance (RQ3) Finally, we assess their performance on relative distance. We design a setting considering three *yellow* objects with different shapes (i.e., *yellow star*, *yellow triangle*, and *yellow circle*). Due to the large number of combinations, we place each object in one of 9 sections (see Single Object in Section 4.1). For each resulting configuration, we sample the cell uniformly from the section (for a total of 4,374 scenes).

4.2 Models

We select representative VLMs covering several architectures and training strategies: LLaVA-NeXT 7B (Liu et al., 2024), Molmo 7B-O (Deitke et al., 2024), and Qwen2-VL-7B-Instruct (Wang et al., 2024). To study whether scaling the text decoder

³The *small* size is a quarter of the *regular* size (half the width and height).

affects the performance, we also evaluate LLaVA-NeXT 13B. As it is used as the vision encoder for LLaVA-NeXT and Molmo, we also consider CLIP ViT-L/14-336px (Radford et al., 2021) to understand its contribution on the performance of the VLMs. Since CLIP was trained to maximize the similarity between text and an image, we map our closed-ended question-answering task into a classification task, where the classes are the answer options. We follow standard practice for zero-shot image classification with CLIP (Radford et al., 2021), i.e., we use CLIP to encode the image and each option, and select the option with the highest similarity to the image.

In general, most VLMs include a vision encoder and a text decoder, but they differ in how they combine these components and present visual information to the text decoder. Both LLaVA-NeXT and Molmo combine their pre-trained encoder and decoder using a projection layer. However, LLaVA-NeXT only trains the projection layer, while Molmo fine-tunes the whole architecture. Qwen2-VL instead trains the vision encoder and the text decoder jointly, forcing them to learn a shared feature representation (without the need for a projection layer). Regarding the vision encoder, LLaVA-NeXT and Molmo rely on the pre-trained vision encoder of CLIP, which can only handle images of exactly 336×336 pixels. For higher-resolution images, these models must either resize or subdivide images into smaller patches, potentially losing detail. Differently, the vision encoder of Qwen2-VL natively handles images of different resolutions, without any resizing. Additional details about the GPU requirements and the models are available in Appendix A.1.

5 Evaluation

We assess the capability of five state-of-the-art VLMs to predict the underlying structure and semantics of a scene. For each model, we report its performance when recognizing the property of an object (RQ1), evaluate its robustness to variations in the object positioning (RQ2), and measure its ability to identify relations among objects (RQ3). Finally, we compare the model’s performance with that of humans, and the human performance with the ground truth of the stimuli. Following preliminary experiments on image and object sizes (see Appendix A.3), we select the best setting, i.e., *regular* size objects with 672×672 images for

Model	Color					
	R	G	B	Y	M	C
LLaVA-NeXT 7B	100	87	83	95	99	56
LLaVA-NeXT 13B	88	86	74	98	74	2
Molmo-O 7B	98	98	96	99	98	96
Qwen2-VL 7B	100	100	99	100	97	100
CLIP	100	99	88	100	100	82

Table 4: F1-Score (%) of each model when considering visual scenes containing a single object and querying about its *color*. There are six possible colors: red (R), green (G), blue (B), yellow (Y), magenta (M), and cyan (C). Results are based on 672×672 images.

elementary objects, and 1344×1344 for COCO objects.

5.1 RQ1: Can VLMs accurately recognize basic object properties?

We report results on VLMs’ recognition of basic properties and absolute positions when analyzing a single, elementary object. To consolidate our findings and avoid overestimating VLMs performance, we extend our study by testing these models on real-world objects.

Single Object Table 3 shows the results of the Single Object experiment (Section 4.1) considering elementary objects. Since all classes in our dataset are balanced, we report performance in terms of accuracy.

Among our set of properties, *shape* was the easiest to recognize, with models obtaining at least 95%. Conversely, VLMs achieved the worst performance when predicting the *sheen* of the object, with the best accuracy reaching 64%. Regarding the *color*, the LLaVA-NeXT models obtained the worst results, with the smaller 7B model being 12% more accurate than its larger 13B counterpart, suggesting that scaling the text decoder does not always improve performance. When considering colors individually (Table 4), both models showed lower performance on green, blue, and especially cyan (with the smaller 7B model achieving 56% F1, while the larger 13B model only 2%). This can be partly explained by the performance of their vision encoder, CLIP, which showed lower performance on blue and cyan. Nevertheless, despite using CLIP, Molmo showed no relevant difference in performance on these colors, obtaining almost perfect accuracy (along with Qwen2-VL). Similarly, Molmo and Qwen2-VL achieved the best results

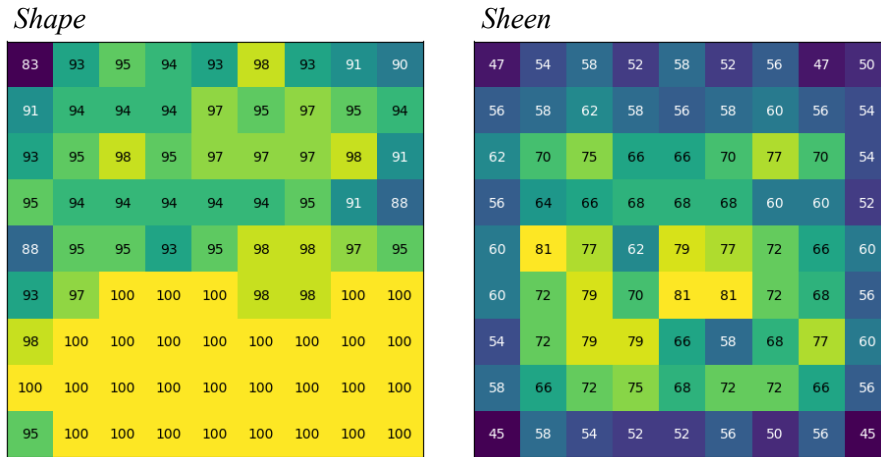


Figure 2: Accuracy (%) of LLaVA-NeXT 13B in each cell of our 9×9 world when queried about the *Shape* & *Sheen* of a Single Object. Results are based on 672×672 images.

when querying about the *absolute position* of an object, while LLaVA-NeXT models achieved around 15% less accuracy. When comparing the performance of the other VLMs with CLIP, no significant differences can be observed when querying for the properties (i.e., *shape*, *color*, *sheen*) of an object. However, while CLIP obtained close to random performance when predicting the *absolute position* of an object (w.r.t. the background), LLaVA-NeXT and Molmo showed considerable improvements (up to 35%), suggesting that the LLMs used as text decoders may have a positive impact on visual tasks. However, the different fine-tuning data may also be responsible for the improvements.

Single Object w. COCO To extend our findings to real-world scenes, we experiment with objects extracted from the COCO dataset. We evaluate the models when predicting the *category* and the *absolute position* of the object, and report the results in Table 2. When querying about the object *category*, all VLMs showed higher accuracy than CLIP, suggesting that fine-tuning and the additional text decoder (LLM) can benefit visual-language tasks. Similarly to the synthetic objects, increasing the size of the text decoder (from 7B to 13B) does not always improve performance. Regarding the *absolute position*, performance with COCO objects follows an analogous trend w.r.t. synthetic objects (with CLIP performing close to random).

5.2 RQ2: Is VLM performance robust to variations in object positioning?

Since understanding requires consistent performance across varying conditions, we study whether changes in object position within the visual scene

affect the recognition of object properties. We then investigate how VLMs associate natural language position (e.g., *top left*) with specific regions of a scene, and compare their predictions to those of humans using identical stimuli.

Effect of Object Position on Accuracy We measure the accuracy in each cell of our 9×9 world and report additional results in Appendix A.5. In all Single Object experiments, we find that accuracy is not uniform over all cells, but varies considerably when changing the position of the object. As an example, we show the results of LLaVA-NeXT 13B on *shape* and *sheen* in Figure 2. Regarding the *shape* of an object, LLaVA-NeXT 13B performed worst in the top corners (83% and 90%), but achieved 100% accuracy in almost every cell of the bottom part of the scene, suggesting the tendency to look at the last visual tokens. When querying about the *sheen*, the overall performance of LLaVA-NeXT 13B was poor (64% as shown in Table 3). However, Figure 2 shows that its performance on *sheen* reached 81% near the center, while it dropped to a minimum of 45% towards the corners.

Regarding the *absolute position* of the object, models show higher performance in the corners and the center. Because these models are mostly trained on image-caption pairs, their definition of *top left* could refer to a different part of the scene. Since we arbitrarily assigned each cell to a particular absolute position (see Single Object in 4.1), we report the models’ position assignment for each cell (Figure 3) when considering the object that models recognized best (*yellow star*). Although each VLM tends to use a different position assign-

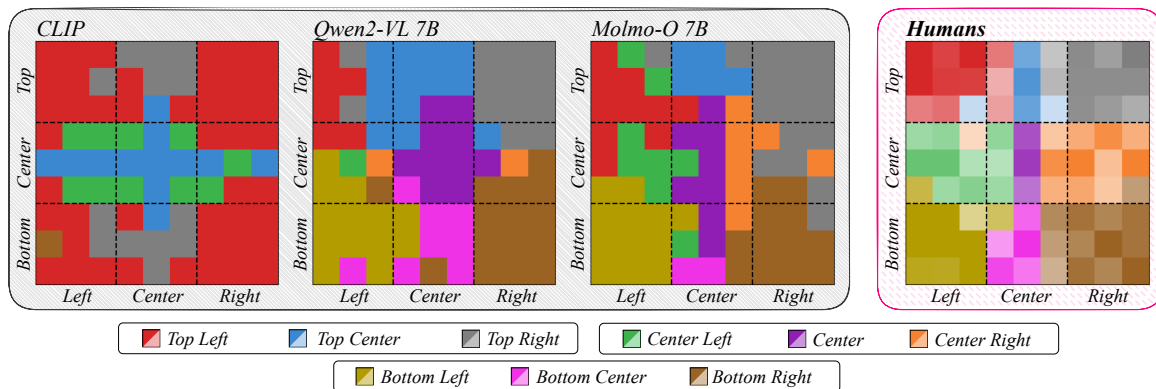


Figure 3: "Where is the yellow star?" - Responses of CLIP, Qwen, Molmo-O 7B, and Humans when asking a closed-ended question about the position of a yellow star on a black background. The question was asked by placing the object in all cells of a 9×9 grid. Since we obtained multiple human annotations for the same stimulus, we report the majority vote. Dashed lines delimit the ground truth sections, and colors indicate the response for each cell. For Humans, the colors fade to white to represent the decrease in agreement (% votes for the majority class). Results are based on 672×672 images.

ment, Qwen2-VL and Molmo assign the corners and the center to the correct position. In particular, they obtained almost perfect accuracy on the *top right*, *bottom right*, and *bottom left* sections of the scene, indicating that there may be a bias in their training data. Moreover, these models rarely invert top with bottom or left with right, suggesting that they have some understanding of position. The only exception is CLIP, which showed close to random performance and assigned *top center* to the central section of the scene and *top left* to most of the other cells.

Human Evaluation: Absolute Position To get more insights about our arbitrary position assignment, we conducted a human evaluation with a subset of input stimuli used with VLMs. For the evaluation, we selected all 81 visual scenes containing one *yellow star* (i.e., one for each cell of our 9×9 world), given that it was the *shape-color* combination with the highest average F1 across all models. Similarly to what we did for the models, we asked human annotators "Where is the yellow star?" while providing the set of possible answers. We report the guidelines and the user interface for the annotation in Appendix A.8. For the annotation, we recruited 124 English-speaking Amazon Mechanical Turkers⁴ and assigned 10 stimuli to each. Of these participants, 91 were approved after quality control and compensated with \$2.00⁵. Each stimulus was annotated 8 times and the inter-annotator agreement as measured by Fleiss' κ (Fleiss, 1971)

⁴<https://www.mturk.com/>

⁵This corresponds to \$24.00/hour, given that the task took 5 minutes on average

was 0.61 (substantial agreement).

Based on the annotation results, humans achieved an accuracy of 73%, higher than all the models. When considering majority voting to determine the *absolute position* of the object (see Figure 3), humans separate the vertical component in three equal bands, granting them an almost perfect performance on the vertical component (*top*, *center*, *bottom*). Instead, for the horizontal component (*left*, *center*, *right*), humans tend to shrink the area for the center to only the central row of positions, leaving more to left and right. Regarding their confidence (i.e., the ratio between the most voted label and total votes), humans show higher agreement near the corners and the center, while lower near borders and around the center. This suggests that humans are capable of locating a position in the scene (e.g., *center*, or *top left*), but are biased towards *left* and *right*.

When comparing with the models, Molmo is the only model assigning a narrow set of cells in the center for the horizontal component, showing similar performance to humans (possibly due to human-annotated positions in its training data (Deitke et al., 2024)). On the other hand, Qwen2-VL assigned a higher number of cells to both *top-center* and *bottom-center*, resembling our position assignment.

5.3 RQ3: Can VLMs identify basic relations among objects?

Table 5 shows the accuracy when predicting the *relative position*, *relative distance*, and *relative size* among the objects. Regarding the *relative position*, all VLMs achieved a higher accuracy than

Model	Relative		
	Position	Distance	Size
<i>Random Baseline</i>	13	50	33
<i>LLaVA-NeXT 7B</i>	24	54	30
<i>LLaVA-NeXT 13B</i>	38	59	33
<i>Molmo-O 7B</i>	24	76	30
<i>Qwen2-VL 7B</i>	46	83	54
<i>CLIP</i>	20	51	49

Table 5: Accuracy (%) of each model when considering visual scenes containing different objects and querying about the *relative position*, *relative distance*, and *relative size* of one object w.r.t. the others. Results are based on 672×672 images.

CLIP (20%), with Qwen2-VL obtaining the highest performance (46%). Results show that relations about objects on the same row/column (i.e., *directly above*, *directly left*, *directly right*, *directly below*) were harder to predict, with Qwen2-VL achieving an average of 18.5% F1. Additionally, CLIP achieved 33% F1 on *above left* but 0% on all other relations, suggesting the presence of a strong bias. Additional results can be found in Appendix A.6.

When queried about the *relative distance* of an object (i.e., identifying the closest object), CLIP and LLaVA-NeXT models (7B & 13B) achieved performance close to random. As shown in Table 6, this is partly due to the failure of LLaVA-NeXT models to detect the closest object to a *triangle*. However, when predicting the *shape* of a Single Object, LLaVA-NeXT models obtained almost a perfect F1-score on *triangle* ($\geq 96\%$ F1). CLIP shows a similar problem with the *circle*, which, although it was the shape with the lowest performance, obtained an F1 score of 91%. These findings suggest that, despite being able to recognize the *shape* of an object, some VLMs are unable to use this property to refer to an object, making their performance task-dependent.

Relative size proved to be the hardest relation to predict since only Qwen2-VL and CLIP achieved better than random performance. As reported in Table 7, the other VLMs never predicted *same* correctly (while Qwen2-VL achieved 47% F1 and CLIP 66% F1). Despite its performance, CLIP showed poor results on *smaller* and *larger* ($\leq 17\%$), indicating a bias for *same*.

Model	Relative Distance		
	Circle	Star	Triangle
<i>LLaVA-NeXT 7B</i>	62	64	14
<i>LLaVA-NeXT 13B</i>	67	71	10
<i>Molmo-O 7B</i>	76	79	71
<i>Qwen2-VL 7B</i>	83	85	81
<i>CLIP</i>	0	55	65

Table 6: F1-Score (%) of each model when considering visual scenes containing multiple objects and querying about the *relative distance* of one object w.r.t. other two objects. Results are based on 672×672 images.

Model	Relative Size		
	Smaller	Same	Larger
<i>LLaVA-NeXT 7B</i>	41	0	38
<i>LLaVA-NeXT 13B</i>	45	0	42
<i>Molmo-O 7B</i>	42	0	33
<i>Qwen2-VL 7B</i>	57	47	61
<i>CLIP</i>	17	66	0

Table 7: F1-Score (%) of each model when considering visual scenes containing multiple objects and querying about the *relative size* of one object w.r.t. another. Results are based on 672×672 images.

6 Conclusion

In this work, we studied whether state-of-the-art VLMs understand the underlying structure and semantics of a visual scene. To respond to the lack of standardized and systematic evaluation, we introduce CIVET, a framework to systematically assess VLM’s understanding via controlled stimuli. Our study reveals that 1) VLMs are only capable of recognizing certain properties, 2) their performance heavily depends on the position of the object, and 3) they struggle to identify basic relations among objects. Moreover, a comparative analysis with human annotators shows that VLMs fall short of human-level accuracy. Our findings indicate that VLMs have limited understanding, which limits their generalization in learning. We encourage further community engagement to extend CIVET and promote novel training paradigms that are pedagogical rather than utility driven, i.e. fine-tuning for downstream tasks.

Limitations

Due to the limited computational resources, we could not experiment with larger models, limiting

the results on the effect of model size to 7B and 13B models. Regarding the effect of the objects' size, further study is needed, as we only considered two variants. Furthermore, differences in the sets of crowd workers may result in variations in the human evaluation.

Ethical Statement

The engagement of crowd-workers for human evaluation does not introduce any ethical concern since the task solely consisted of annotating the position of a yellow star on a black background, which has a low cognitive load.

Acknowledgments

We acknowledge the support of the MUR PNRR project FAIR - Future AI Research (PE00000013) funded by the NextGenerationEU.

References

- Manoj Acharya, Kushal Kafle, and Christopher Kanan. 2019. [Tallyqa: Answering complex counting questions](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):8076–8084.
- Sumukh K Aithal, Pratyush Maini, Zachary C. Lipton, and J. Zico Kolter. 2024. [Understanding hallucinations in diffusion models through mode interpolation](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 134614–134644. Curran Associates, Inc.
- Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. 2024. [Spatialvlm: Endowing vision-language models with spatial reasoning capabilities](#). In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14455–14465.
- Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. 2024. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. *arXiv preprint arXiv:2409.17146*.
- Reza Esfandiarpour, Cristina Menghini, and Stephen Bach. 2024. [If CLIP could talk: Understanding vision-language model representations through their preferred concept descriptions](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9797–9819, Miami, Florida, USA. Association for Computational Linguistics.
- Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.
- Xingyu Fu, Sheng Zhang, Gukyeong Kwon, Pramuditha Perera, Henghui Zhu, Yuhao Zhang, Alexander Hanbo Li, William Yang Wang, Zhiguo Wang, Vittorio Castelli, Patrick Ng, Dan Roth, and Bing Xiang. 2023. [Generate then select: Open-ended visual question answering guided by world knowledge](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2333–2346, Toronto, Canada. Association for Computational Linguistics.
- Xingyu Fu, Ben Zhou, Ishaan Chandratreya, Carl Vondrick, and Dan Roth. 2022. [There's a time and place for reasoning beyond the image](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1138–1149, Dublin, Ireland. Association for Computational Linguistics.
- Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913.
- Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. 2017. [Clevr: A diagnostic dataset for compositional language and elementary visual reasoning](#). In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1988–1997.
- Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. 2024. [Prismatic vlms: Investigating the design space of visually-conditioned language models](#). In *International Conference on Machine Learning (ICML)*.
- Seunghoi Kim, Chen Jin, Tom Diethe, Matteo Figini, Henry F. J. Tregidgo, Asher Mullokandov, Philip Teare, and Daniel C. Alexander. 2025. [Tackling structural hallucination in image translation with local diffusion](#). In *Computer Vision – ECCV 2024*, pages 87–103, Cham. Springer Nature Switzerland.
- Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. 2023. [Segment anything](#). In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026.
- Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannic Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73.
- Hugo Laurençon, Leo Tronchon, Matthieu Cord, and Victor Sanh. 2024. [What matters when building vision-language models?](#) In *Advances in Neural*

- Information Processing Systems*, volume 37, pages 87874–87907. Curran Associates, Inc.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.
- Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024. [Llava-next: Improved reasoning, ocr, and world knowledge](#).
- Roni Paiss, Ariel Ephrat, Omer Tov, Shiran Zada, Inbar Mosseri, Michal Irani, and Tali Dekel. 2023. [Teaching clip to count to ten](#). In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3147–3157.
- Wujian Peng, Sicheng Xie, Zuyao You, Shiyi Lan, and Zuxuan Wu. 2024. Synthesize diagnose and optimize: Towards fine-grained vision-language understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13279–13288.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. [Learning transferable visual models from natural language supervision](#). In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.
- Bradley Rettler and Andrew M. Bailey. 2024. Object. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*, Summer 2024 edition. Metaphysics Research Lab, Stanford University.
- Karsten Roth, Jae Myung Kim, A. Sophia Koepke, Oriol Vinyals, Cordelia Schmid, and Zeynep Akata. 2023. [Waffling around for performance: Visual classification with random words and broad concepts](#). In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15700–15711.
- Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. 2021. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*.
- Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. 2022. [Winoground: Probing vision and language models for visio-linguistic compositionality](#). In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5228–5238.
- Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. 2024. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.
- Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhua Chen. 2024. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of CVPR*.
- Mert Yuksekogunul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. 2022. When and why vision-language models behave like bags-of-words, and what to do about it? *arXiv preprint arXiv:2210.01936*.

A Appendix

A.1 Experimental Details

Most experiments were executed using one NVIDIA A100 with 40 GiB. The only exception was Qwen2-VL-7B-Instruct, which required one NVIDIA A100 with 80 GiB when considering images of size 1344×1344 . In all experiments, we used greedy generation to generate the answers to our closed-ended questions. Regarding the models, we considered the following HuggingFace checkpoints:

1. LLaVA-NeXT 7B, <https://huggingface.co/llava-hf/llava-v1.6-vicuna-7b-hf>
2. LLaVA-NeXT 13B, <https://huggingface.co/llava-hf/llava-v1.6-vicuna-13b-hf>
3. Molmo 7B-O, <https://huggingface.co/allenai/Molmo-7B-0-0924>
4. Qwen2-VL-7B-Instruct, <https://huggingface.co/Qwen/Qwen2-VL-7B-Instruct>
5. CLIP ViT-L/14-336px, <https://huggingface.co/openai/clip-vit-large-patch14>

A.2 Answers Length and "Other" values

In a preliminary experiment on Single Object (see Section 4.1), we noticed that only Molmo answered with one token while the other VLMs answered the questions with 8 to 15 tokens (tokenized with NLTK⁶). Because of this, only manual checking would have been appropriate, as exact matching would have resulted in underestimating the performance of more verbose models. For this reason, we prepended the instruction "*Answer with as few words as possible.*" as an attempt to condition the model to generate only the property values as the answer. Table 10 shows this had the desired effect of reducing the number of tokens to 1 (± 0.5 for Qwen2-VL only) for the property questions, and 2 tokens (± 0.5 at most) for the position questions.

Since we used greedy generation, we also measured how often the models responded with "other" answers: either something outside the provided set or multiple options from the set. When computing accuracy, we considered "other" answers as mistakes. As shown in Table 11, Molmo was the only model that never generated "other" answers. Instead, Qwen2-VL shows a small percentage of "other" answers (below 2%) regardless of the question, while the two LLaVA-NeXT models only generate them when querying about color (under 1%

⁶<https://www.nltk.org/>

for 7B, and under 6.8% for 13B). Additionally, adding the instruction for shorter answers reduced the frequency of "other" answers. We report the full tables for the answer lengths in Table 10 and the percentage of "other" answers in Table 11.

A.3 Effect of Image and Object Size

As high-resolution images are known to increase the performance of VLMs (Laurençon et al., 2024; Karamcheti et al., 2024), we performed additional experiments to understand how different image and object sizes could affect the model performance. We experiment with three image sizes (336×336 , 672×672 , and 1344×1344) and two object sizes (*regular*, and *small*) to understand how these parameters affect the models' performance when predicting object properties and absolute position.

Table 12 shows the accuracy of each model when considering the three image sizes. Increasing the image size affects CLIP performance only negligibly (variations are likely to be attributed to the re-size operation). On the other hand, VLMs achieve higher accuracy with larger images, suggesting that encoding multiple high-resolution patches helps capture more details. When considering synthetic objects, increasing the image size to 672×672 leads to a higher performance. However, further increasing the image size to 1344×1344 does not provide any significant improvement. When considering COCO objects, CLIP, LLaVA-NeXT, and Molmo show a similar trend. Instead, Qwen2-VL is the only model improving accuracy (97% on *category*), suggesting that encoding the whole image with no resizing is more advantageous than encoding multiple high-resolution patches (LLaVA-NeXT, and Molmo). Regarding the size of the object, using *small* objects leads to a drop in performance in most cases for all models.

Table 8 shows the difference in accuracy between *regular* (i.e., Table 12) and *small* objects (i.e., resized by $\frac{1}{4}$). Similarly, Table 9 shows the difference in accuracy between *regular* and *small* objects when considering COCO Objects. In both cases, most model performance is higher when considering *regular* objects.

As image and object size affect performance, for the rest of the experiments we report only the results in the best setting, i.e., *regular* size objects with 672×672 images for synthetic objects, and 1344×1344 for COCO objects.

A.4 Single Object

We provide the F1-Score for the remaining Single Object experiments when considering images with 672×672 pixels.

Tables 13, 14, and 15 show the F1-Score of each model when considering only one object and querying about its *shape*, *sheen*, and *position* (w.r.t. the background), respectively.

A.5 Effect of Object Position on Accuracy

We report additional findings when investigating the accuracy of VLMs in each cell of our 9×9 world. Figure 4 shows the results for LLaVA-NeXT 7B and Molmo when asking about the shape of an object. Similarly to its larger counterpart (shown in Figure 2), LLaVA-NeXT 7B performed worse in the top corners, obtaining almost perfect accuracy in the bottom part of the scene and showing the same tendency to look at the last visual tokens. Since both LLaVA-NeXT models suffer from this bias, it may be related to the data used to train the projection layer. On the other hand, Molmo, whose accuracy was 100% when recognizing the shape of an object, is the only model showing almost perfect accuracy across the whole scene (except for the cell in the upper right corner).

When queried about the *color* of a Single Object, Molmo and Qwen2-VL obtained 98% and 99% accuracy, respectively (see Table 3). However, when looking at Figure 5 it is possible to notice how performance depends on the position of the object. Similarly to *shape* (Figure 4), Molmo obtained the worst accuracy when the object was placed in the top right corner of the scene, possibly indicating the presence of a bias. In general, cells with higher and lower accuracy are randomly spread across the whole scene. This also happens for Qwen2-VL, with the only difference that cells with the lowest accuracy are more present in the last row (i.e., bottom part) of the scene.

Figure 6 shows Molmo accuracy when asking about the *sheen* of an object when placed in different cells of our world. Similarly to LLaVA-NeXT 13B (see Figure 2), Molmo accuracy is higher towards the cells in the center. However, performance is on average worse for all the cells in the right section of the scene (especially in the top right).

A.6 Relative Position

Table 16 shows the F1-Score of each model when considering visual scenes containing two objects

and querying about the *relative position* of one object w.r.t. the other.

A.7 Experiments on COCO

Table 17 shows the accuracy of each model when considering different objects from the COCO dataset (i.e., zebra, giraffe, and elephant). Additionally, Table 18 shows the F1-Score of each model when considering visual scenes containing different objects from COCO (i.e., zebra, giraffe, and elephant) and querying about their *category*.

A.8 Human Evaluation

We report the guidelines provided to the human annotators in Figure 7 and the user interface for the annotation task in Figure 8.

Model	Image Size	Shape	Color	Sheen	Position
<i>LLaVA-NeXT 7B</i>	336	↑7	↑2	0	↑1
	672	↑6	0	↓1	0
	1344	↑5	↓1	0	0
<i>LLaVA-NeXT 13B</i>	336	↑6	0	↑3	↑1
	672	↑4	0	↑6	↑2
	1344	↑3	0	↑12	↑2
<i>Molmo-O 7B</i>	336	↑5	↑1	0	↑4
	672	0	↑2	↑1	↑2
	1344	0	↑1	↑1	↑2
<i>Qwen2-VL 7B</i>	336	↑9	↑2	↑3	↑1
	672	↑1	↑1	↑5	↑1
	1344	↑1	0	↑4	↑1
<i>CLIP</i>	336	↑4	↓2	0	↓1
	672	0	0	↓1	0
	1344	↑1	↑1	↓1	0

Table 8: Difference in accuracy between *Regular* (i.e., Table 12) and *small* objects (i.e., resized by $\frac{1}{4}$)

Model	Image Size	Category	Position
<i>LLaVA-NeXT 7B</i>	336	↑15	↑2
	672	↑50	↑5
	1344	↑47	↑11
<i>LLaVA-NeXT 13B</i>	336	↑26	↑5
	672	↑28	↑1
	1344	↑15	↑12
<i>Molmo-O 7B</i>	336	↑1	↑7
	672	↑34	↑14
	1344	↑37	↑7
<i>Qwen2-VL 7B</i>	336	↑32	↑16
	672	↑28	↑13
	1344	↑9	↑5
<i>CLIP</i>	336	↑16	↑2
	672	↑9	↑2
	1344	↑11	↓1

Table 9: Difference in scuracy between *regular* (i.e., Table 17) and *small* objects (i.e., resized by $\frac{1}{4}$) when considering COCO objects.

Model	Image Size	Shape	Color	Sheen	Position
<i>LLaVA-NeXT 7B</i>	336	1 ± 0.00	1 ± 0.00	1 ± 0.00	2 ± 0.35
	672	1 ± 0.00	1 ± 0.00	1 ± 0.00	2 ± 0.35
	1344	1 ± 0.00	1 ± 0.00	1 ± 0.00	2 ± 0.35
<i>LLaVA-NeXT 13B</i>	336	1 ± 0.00	1 ± 0.00	1 ± 0.00	2 ± 0.47
	672	1 ± 0.00	1 ± 0.00	1 ± 0.00	2 ± 0.46
	1344	1 ± 0.00	1 ± 0.00	1 ± 0.00	2 ± 0.46
<i>Molmo-O 7B</i>	336	1 ± 0.00	1 ± 0.00	1 ± 0.00	2 ± 0.30
	672	1 ± 0.00	1 ± 0.00	1 ± 0.00	2 ± 0.29
	1344	1 ± 0.00	1 ± 0.00	1 ± 0.00	2 ± 0.29
<i>Qwen2-VL 7B</i>	336	1 ± 0.07	1 ± 0.06	1 ± 0.48	2 ± 0.29
	672	1 ± 0.01	1 ± 0.01	1 ± 0.44	2 ± 0.34
	1344	1 ± 0.05	1 ± 0.01	1 ± 0.29	2 ± 0.33

Table 10: Answers length (i.e., average number of tokens with its standard deviation) of each model when considering different image sizes.

Model	Image Size	Shape	Color	Sheen	Position
<i>LLaVA-NeXT 7B</i>	336	0.00	0.69	0.00	0.01
	672	0.00	0.35	0.00	0.00
	1344	0.00	0.33	0.00	0.00
<i>LLaVA-NeXT 13B</i>	336	0.00	6.76	0.00	0.00
	672	0.00	2.04	0.00	0.00
	1344	0.00	1.84	0.00	0.00
<i>Molmo-O 7B</i>	336	0.00	0.00	0.00	0.00
	672	0.00	0.00	0.00	0.00
	1344	0.00	0.00	0.00	0.00
<i>Qwen2-VL 7B</i>	336	0.70	0.98	1.57	0.30
	672	0.30	0.89	1.63	0.33
	1344	0.27	1.27	1.39	0.36

Table 11: Percentage of "other" answers (i.e., answers outside the provided set of options, or multiple options from the set) generated by a model when considering different image sizes.

Model	Image Size	Shape	Color	Sheen	Position
<i>Random Baseline</i>		25	17	50	11
<i>LLaVA-NeXT 7B</i>	336	98	90	50	37
	672	98	88	50	42
	1344	97	88	50	42
<i>LLaVA-NeXT 13B</i>	336	100	71	57	37
	672	97	76	64	47
	1344	97	76	67	47
<i>Molmo-O 7B</i>	336	100	94	54	64
	672	100	98	59	62
	1344	100	98	59	62
<i>Qwen2-VL 7B</i>	336	98	98	55	52
	672	99	99	60	61
	1344	99	98	61	64
<i>CLIP</i>	336	99	89	50	13
	672	95	95	49	14
	1344	95	95	49	14

Table 12: Accuracy (%) of each model when considering visual scenes containing a single object and querying about its *shape*, *color*, *sheen*, and *absolute position* (w.r.t. the background).

LLaVA-NeXT 7B

90	91	94	97	95	100	95	95	90
94	94	97	95	98	100	97	95	97
94	98	97	100	100	100	97	98	95
94	97	100	98	100	100	98	98	97
90	100	98	100	97	98	97	97	93
90	97	100	98	97	97	95	97	97
97	100	100	100	100	100	100	100	100
98	100	100	100	100	100	100	100	100
94	100	100	100	100	100	100	100	97

Molmo-O 7B

100	100	100	100	100	100	100	100	98
100	100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100	100

Figure 4: Accuracy (%) of LLaVA-NeXT 7B & Molmo 7B in each cell of our 9×9 world when queried about the *shape* of a Single Object. Results are based on 672×672 images.

Molmo-O 7B

95	95	97	97	95	95	95	94	91
95	94	93	95	97	97	97	97	97
97	95	100	95	98	94	98	97	100
100	95	100	98	100	97	98	98	97
98	100	100	98	98	97	98	100	98
100	97	95	100	97	100	98	98	97
94	98	97	98	98	97	98	100	100
97	98	97	97	95	97	95	100	97
95	97	97	100	97	95	97	98	98

Qwen2-VL 7B

97	98	97	97	98	100	98	97	98
100	100	98	95	97	95	98	98	100
98	97	98	100	95	97	100	98	97
98	95	98	98	100	98	98	100	98
100	97	97	100	100	98	100	98	100
98	100	100	100	100	98	98	98	100
97	98	100	100	100	100	100	98	97
97	98	98	97	97	98	97	98	100
98	100	100	95	97	100	98	95	95

Figure 5: Accuracy (%) of Molmo 7B & Qwen2-VL 7B in each cell of our 9×9 world when queried about the *color* of a Single Object. Results are based on 672×672 images.

Model	Shape			
	Square	Circle	Triangle	Star
<i>LLaVA-NeXT 7B</i>	96	96	100	100
<i>LLaVA-NeXT 13B</i>	97	98	96	96
<i>Molmo-O 7B</i>	100	100	100	100
<i>Qwen2-VL 7B</i>	98	99	99	100
<i>CLIP</i>	99	91	93	98

Table 13: F1-Score (%) of each model when considering visual scenes containing a single object and querying about its *shape*. Images have 672×672 pixels.

Model	Sheen	
	Matte	Glossy
<i>LLaVA-NeXT 7B</i>	1	67
<i>LLaVA-NeXT 13B</i>	66	61
<i>Molmo-O 7B</i>	31	71
<i>Qwen2-VL 7B</i>	44	70
<i>CLIP</i>	66	0

Table 14: F1-Score (%) of each model when considering visual scenes containing a single object and querying about its *sheen*. Images have 672×672 pixels.

Molmo-O 7B

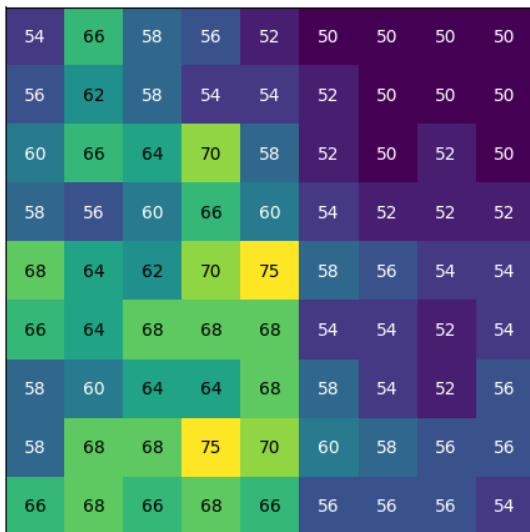


Figure 6: Accuracy (%) of Molmo 7B in each cell of our 9×9 world when queried about the *sheen* of a Single Object. Results are based on 672×672 images.

Model	Top			Center			Bottom		
	Left	Center	Right	Left	Center	Right	Left	Center	Right
<i>LLaVA-NeXT 7B</i>	61	43	56	6	30	4	45	41	53
<i>LLaVA-NeXT 13B</i>	58	44	59	7	43	23	62	42	62
<i>Molmo-O 7B</i>	74	54	74	40	53	46	72	58	76
<i>Qwen2-VL 7B</i>	70	64	75	15	66	19	70	59	67
<i>CLIP</i>	23	14	5	20	0	0	1	4	12

Table 15: F1-Score (%) of each model when considering visual scenes containing a single object and querying about its *absolute position* (w.r.t. the background). Images have 672×672 pixels.

Model	Above		Directly				Below	
	Left	Right	Left	Above	Right	Below	Left	Right
<i>LLaVA-NeXT 7B</i>	34	43	11	13	16	17	18	13
<i>LLaVA-NeXT 13B</i>	55	51	10	2	16	12	37	31
<i>Molmo-O 7B</i>	15	35	16	17	24	15	27	33
<i>Qwen2-VL 7B</i>	54	51	20	15	21	18	52	52
<i>CLIP</i>	33	0	0	0	0	0	0	0

Table 16: F1-Score (%) of each model when considering visual scenes containing two objects and querying about the *relative position* of one object w.r.t. another. Images have 672×672 pixels.

Model	Image Size	Category	Position
<i>Random Baseline</i>		33	11
<i>LLaVA-NeXT 7B</i>	336	55	23
	672	91	35
	1344	91	37
<i>LLaVA-NeXT 13B</i>	336	56	37
	672	90	45
	1344	80	51
<i>Molmo-O 7B</i>	336	34	45
	672	69	59
	1344	70	53
<i>Qwen2-VL 7B</i>	336	60	30
	672	86	49
	1344	97	52
<i>CLIP</i>	336	65	13
	672	67	16
	1344	67	15

Table 17: Accuracy (%) of each model when considering visual scenes containing different objects from COCO (i.e., zebra, giraffe, and elephant) and querying about their *category* and *absolute position* (w.r.t. the background).

Model	Category		
	Giraffe	Elephant	Zebra
<i>LLaVA-NeXT 7B</i>	94	84	92
<i>LLaVA-NeXT 13B</i>	88	59	86
<i>Molmo-O 7B</i>	99	16	69
<i>Qwen2-VL 7B</i>	96	95	99
<i>CLIP</i>	70	0	94

Table 18: F1-Score (%) of each model when considering visual scenes containing different objects from COCO (i.e., zebra, giraffe, and elephant) and querying about their *category*. Images have 1344×1344 pixels.

Guidelines

Motivation

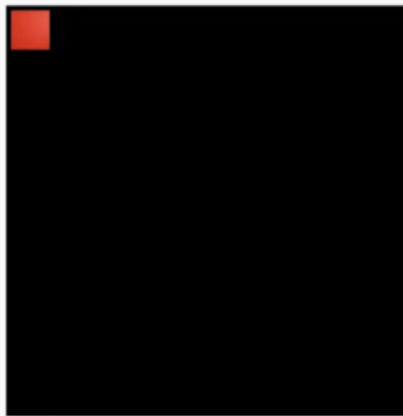
The purpose of this study is to investigate how people visually perceive the position of an object in an image.

Task

In this task, we ask you to annotate a series of images by choosing one option from those proposed. Each image contains an object, and the task is to say where the object is located. This is done by choosing among the provided position categories (e.g., top right, center, bottom left). To answer, select the option that best describes the position of the object by clicking on the corresponding text. The answer can be changed until you decide to proceed to the next image. Once you press "Next" you will not be able to change your answer. Press "Next" to proceed to the next image.

Example

This is an example of an image you will see during the task.



An example of a question with the options follows (the question is **not** related with the image above). To aid the selection, the positions are grouped by: **CENTER**, **BOTTOM**, and **TOP**. For example the group **TOP** contains: "Top Center", "Top Right", and "Top Left".

If you think the answer is "Bottom Left", you can:

1. look for **BOTTOM**
2. find "Bottom Left"
3. press "Next" to go to the next image

Where is the yellow triangle?

CENTER	① →	BOTTOM
<input type="button" value="Center"/>		<input type="button" value="Bottom Center"/>
<input type="button" value="Center Right"/>		<input type="button" value="Bottom Right"/>
<input type="button" value="Center Left"/>		<input type="button" value="Bottom Left"/>

② ↑

TOP
<input type="button" value="Top Center"/>
<input type="button" value="Top Right"/>
<input type="button" value="Top Left"/>

③ →

Figure 7: Guidelines for the proposed human evaluation task.

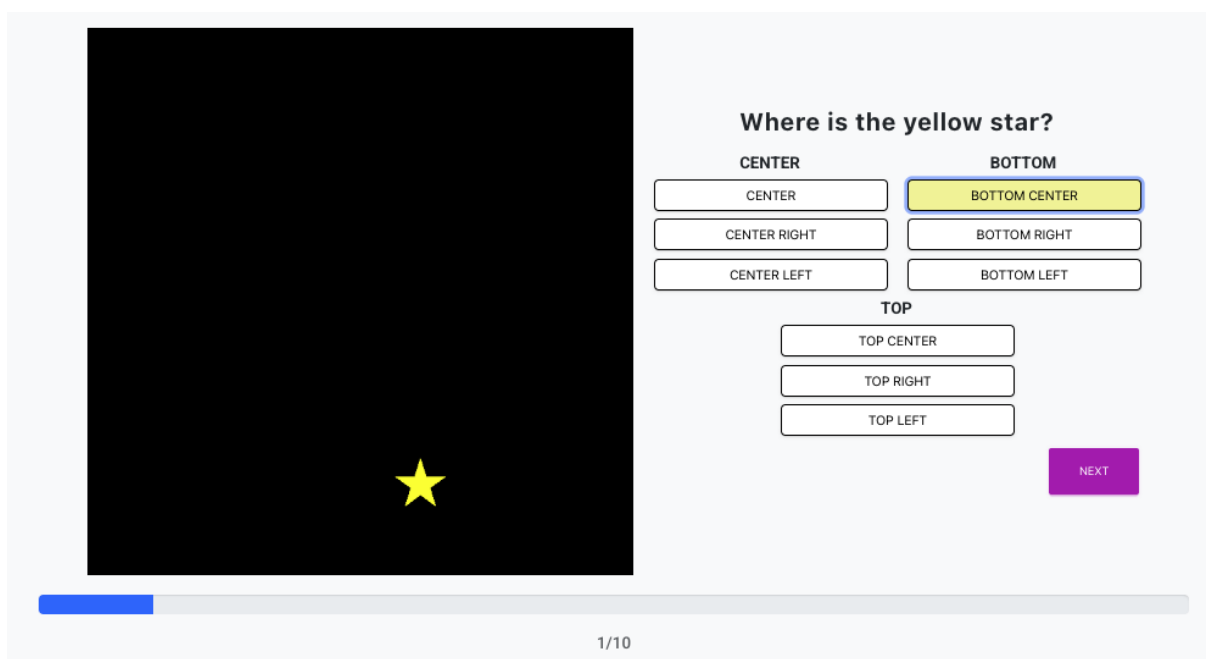


Figure 8: User interface for the proposed human evaluation task.