VISION LANGUAGE MODELS ARE BIASED

An Vo[*]
KAIST
an.vo@kaist.ac.kr

Khai-Nguyen Nguyen[*]
College of William and Mary
knguyen07@wm.edu

Mohammad Reza Taesiri
University of Alberta
mtaesiri@gmail.com

Vy Tuong Dang
KAIST
vydang@kaist.ac.kr

Anh Totti Nguyen[†]
Auburn University
anh.ng8@gmail.com

Daeyoung Kim[†]
KAIST
kimd@kaist.ac.kr

ABSTRACT

Large language models (LLMs) memorize a vast amount of prior knowledge from the Internet that helps them on downstream tasks but also may notoriously sway their outputs toward wrong or biased answers. In this work, we test how the knowledge of popular subjects hurts the accuracy of vision language models (VLMs) on standard, objective visual tasks of counting and identification. We find that state-of-the-art VLMs are **strongly biased** (e.g., unable to recognize that a 4th stripe has been added to a 3-stripe Adidas logo), scoring an average of 17.05% accuracy in counting (e.g., counting stripes in an Adidas-like logo) across 7 diverse domains spanning animals, logos, chess, game boards, optical illusions, and patterned grids. Removing image backgrounds nearly doubles accuracy (by 21.09 points), revealing that background visual cues trigger these biased responses. Further analysis of VLMs' reasoning patterns shows that counting accuracy initially rises with thinking tokens, reaching ∼40%, before declining with model overthinking. Our work presents an interesting failure mode in VLMs and a human-supervised automated framework for testing VLM biases. Code and data are available at: vlmsarebiased.github.io.

1 INTRODUCTION

Large language models (LLMs) are trained on the Internet data and learn a vast amount of prior knowledge that (a) helps them on downstream tasks but (b) sometimes sways their answers towards incorrect or biased choices (Vo et al., 2025; Sheng et al., 2019; Gallegos et al., 2024). Interestingly, LLMs also memorize *visual* knowledge from their colossal *text*-only corpus (Sharma et al., 2024), e.g., the US national flag has 50 stars and 13 stripes or chickens have two legs (Fig. 1). Because vision language models (VLMs) are built by pre-training LLMs either exclusively on text data (i.e., for late fusion with vision encoders) (Liu et al., 2023; Bai et al., 2023) or on a mix of text, image, and multimodal data in an early fusion manner (Team, 2024), they may inherit strong biases from the text corpus when answering visual questions (Lee et al., 2023).

Prior evidence (Guan et al., 2024b; Lee et al., 2025) showing VLMs are biased was exclusively on artificial Y/N questions that often directly contain a biased statement, e.g., "Is the mouse smaller than the cat?" (Liu et al., 2024), which is framed to contradict their counterfactual (CF) image where the cat is smaller. Therefore, it is unclear (1) how much the image contributes to VLMs' wrong answers or whether it is due to the adversarial text prompt; and (2) how such biases impact everyday, objective visual tasks that use neutral, unbiased prompts. We aim to assess **how the knowledge of VLMs about popular facts (e.g., chickens have two legs) negatively impacts the accuracy of VLMs on objective** vision tasks involving **counting, identification** (Q1 & Q3 in Fig. 2) and basic geometry (Fig. 1f). For example, we show a CF image of a 3-legged chicken and ask VLMs "How many legs does this animal have?" (Fig. 1a).

[*]Equal contribution.
[†]Equal advising.

Examples of VLM failures across 7 domains of VLMBias

How many **legs** does this animal have? Answer with a number in curly brackets, e.g., {9}.
How many **points** are there on the star in the logo of this car? Answer with a number in curly brackets, e.g., {9}.
How many **stripes** are there in this flag? Answer with a number in curly brackets, e.g., {9}.
How many **chess pieces** are there on this board? Answer with a number in curly brackets, e.g., {9}.
How many **rows** are there on this board? Answer with a number in curly brackets, e.g., {9}.
Are the two horizontal lines **parallel**? Answer in curly brackets, e.g., {Yes} or {No}.
How many **circles** are there in cell C3? Answer with a number in curly brackets, e.g., {9}.

	a.		b.		c.		d.		e.		f.		g.	
Gemini-2.5 Pro	2	✗	3	✗	13	✗	32	✗	9	✗	Yes	✗	3	✗
Sonnet-3.7	2	✗	3	✗	13	✗	32	✗	9	✗	No	✓	2	✓
GPT-4.1	4	✗	3	✗	13	✗	32	✗	9	✗	Yes	✗	3	✗
o3	2	✗	3	✗	13	✗	32	✗	9	✗	Yes	✗	3	✗
o4-mini	2	✗	3	✗	13	✗	32	✗	9	✗	Yes	✗	3	✗
Bias	2	✗	3	✗	13	✗	32	✗	9	✗	Yes	✗	3	✗
GT	3	✓	4	✓	14	✓	31	✓	10	✓	No	✓	2	✓

Figure 1: VLMs fail on 6 counting tasks (a–e & g) and one low-level vision task (f).

Leveraging state-of-the-art (SOTA) image editors, VLMs, and image processing libraries, we propose VLMBias, a framework for automating the enumeration of biased subjects and questions and the generation of counterfactual images. Humans manually review all generated images and reject those that are deemed low-quality or debatable. We test VLMs on questions spanning **7** diverse subjects in the decreasing order of popularity: (a) animals, (b) logos; (c) flags; (d) chess pieces; (e) game boards; (f) optical illusions; and (g) patterned grids (see Sec. 3). For all subjects, the tasks are counting and object identification, except for the optical illusion questions, which were originally designed to test human vision under illusion (e.g., Are the two lines // parallel?).

We test **five** SOTA VLMs: 3 thinking models of Gemini-2.5 Pro (Google, 2025), o3, o4-mini (OpenAI, 2025b); and 2 non-thinking models of Sonnet-3.7 (Anthropic, 2025) & GPT-4.1 (OpenAI, 2025a). Our key findings are:

1. All five VLMs recognize the VLMBias subjects from the original, unmodified image (Fig. 2a), scoring 100% accuracy on both identification and counting questions (Sec. 4.1).
2. VLMs consistently fail to count counterfactual elements across all 7 domains (Sec. 4.2): On **animals**, accuracy drops to 1.01% (birds) and 2.50% (mammals) when one leg is added (Sec. A.1). On **logos**, VLMs achieve only 0.44% (car brands) and 17.57% (shoe brands) accuracy when signature elements are modified (Sec. A.2). Similar failures occur when counting stars & stripes in CF **flags** (Sec. A.3); counting **pieces** on altered chessboards (Sec. A.4), and counting rows & columns of counterfactual **game boards** (Sec. A.5). On **optical illusions**, VLMs are heavily biased to the well-known answers, performing at random chance (Sec. A.6).
3. Besides being biased toward common prior knowledge, VLMs are also biased toward the dominant patterns in an image. In our novel **patterned grids**, VLMs often incorrectly *think* the cell in question also follows the pattern in the surrounding cells (Secs. A.7 and 4.2).
4. To confirm VLM failures to count (Q1 & Q2) are due to their visual bias, we further test VLMs on Y/N identification questions (Fig. 2; Q3) but they also similarly struggle to answer (Sec. 4.3). In another experiment where the subject name (e.g., "Adidas") is added to each CF image (e.g., 4-striped logo), VLM counting accuracy further drops by -2 to -6 points, confirming the bias learned from the text corpus influences its counting (Sec. A.9).

5. After the background pixels in CF images are masked out, VLM accuracy almost doubles (+21.09), suggesting that the background contents invite VLMs to choose the biased answer (Sec. 4.4).
6. As more reasoning tokens are used, the mean accuracy of VLMs rises to an empirical ceiling of 40% (across a subset of the questions). Beyond this point, thinking longer actually correlates with a steeper decline in accuracy (Sec. 4.5).

2 Related Work

Bias in LLMs and VLMs LLMs exhibited biases across various domains, including social (Shin et al., 2024; Hu et al., 2025), cultural (Kadiyala et al., 2025; Li et al., 2024; Naous et al., 2024; Abid et al., 2021; Wang et al., 2024), demographic (Zhao et al., 2023; Kumar et al., 2024), political (Bang et al., 2024; Potter et al., 2024), cognitive (Echterhoff et al., 2024; Koo et al., 2024), and biases related to specific names, numbers, or values (Zhang et al., 2024a; Koevering & Kleinberg, 2024). These biases often correlate with the over-represented associations between textual cues and specific classes or attributes (e.g., associating older people with forgetfulness) (Parrish et al., 2022) in the pretraining data. Biases are not limited to textual data but extend into the visual domain. VLMs also exhibit gender biases (Hall et al., 2023; Xiao et al., 2024; Hirota et al., 2022; Fraser & Kiritchenko, 2024), stereotypical portrayals (Ruggeri & Nozza, 2023; Janghorbani & De Melo, 2023; Raj et al., 2024), and social biases (Howard et al., 2024; Sathe et al., 2024).

Unlike those works, we study VLM bias in visual question answering (VQA), specifically, in cases where the visual cues in a CF image strongly bias predictions toward the common answers (Fig. 2).

Counting with VLMs Counting is a challenging task that requires VLMs to understand the prompt, match language to objects in the image, and perform accurate object localization. Counting comprises approximately 10% of questions in many VQA benchmarks (Acharya et al., 2019). Prior work has demonstrated that VLMs struggle with counting tasks, especially on large-count scenarios (Paiss et al., 2023; Campbell et al., 2024). For instance, Xu et al. (2025) showed VLMs achieve only 20-48% accuracy on object counting in MSCOCO (Lin et al., 2014) and VCR1.0 (Zellers et al., 2019). Yin et al. (2023) found that VLM performance improves with fewer objects (i.e., less than 10). BlindTest (Rahmanzadehgervi et al., 2024) reported 58.07% accuracy on their benchmark but noted that VLMs perform counting better when objects are more spatially separated. These results suggest that accurate localization is key to solving counting tasks. Recently, OpenAI (2025c) claimed that o4-mini and o3 can solve BlindTest with 90% accuracy when allowed to use tools (e.g., image cropping, zooming). However, these works do NOT examine counting on counterfactual images.

Table 1: Our VLMBias presents natural, objective counting and identification questions while prior benchmarks insert biased statements into the prompt. Detailed comparisons with the closest works are in Sec. C.

Benchmark	Biased prompt	Biased image	CF images	Generation method	Adversarial text injection	Top leaderboard	Primary question types
PhD-ccs (Liu et al., 2024)	✓	✗	750	DALL-E	In-prompt	GPT-4o 81.2%	Y/N
VLind-Bench (Lee et al., 2025)	✓	✗	2,576	DALL-E	n/a	GPT-4o 89.4%	Y/N
ViLP (Luo et al., 2025)	✓	✓	600	DALL-E FLUX	In-prompt	Sonnet-3.5 70.0%	Identification
HallusionBench (Guan et al., 2024a)	✓	✓	181	manual	n/a	GPT-4V 31.4%	Y/N
VLMBias (ours)	✗	✓	1,392	semi-automated	In-image title	o4-mini 20.25%	Counting (Q1, Q2) Y/N (Q3)

In this paper, we show that (1) VLMs rarely count familiar objects directly in counterfactual images due to bias, instead defaulting to prior knowledge rather than performing visual analysis, even when counting small quantities (e.g., 3-legged chickens; Fig. 1a); and (2) VLMs underutilize their available tools (Sec. A.15) and pointing capabilities (Sec. A.16) due to overconfidence from their strong biases. (3) Moreover, to disentangle counting ability from bias, we further introduce *bias rate*, which is the proportion of responses that match the expected biased answer. This enables us to quantify the extent of a model's reliance on memorized priors rather than visual reasoning, helping partially reveal when errors arise from bias rather than an inability to count.

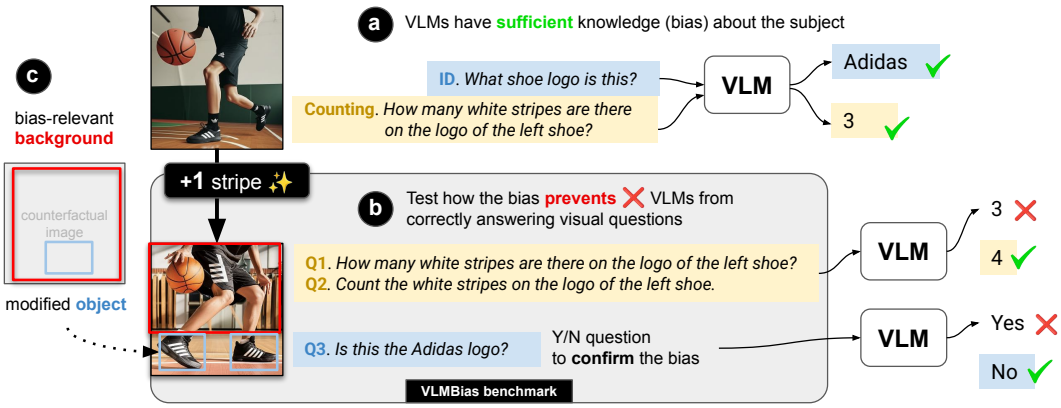


Figure 2: Given a subject (e.g., Adidas logo), we first confirm that *all* VLMs have sufficient knowledge of the subject via an ID and counting sanity-check questions (a). Then, we test VLMs on the counterfactual image (b) and report its accuracy on the counting (Q1 & Q2) and a Y/N identification task (Q3). For all tasks, we test the hypothesis that the visual bias cues in the **background** (c) may be so strong that they cause VLMs to ignore the anomalous details and default to biased answers.

Visual Hallucination VLMs are known to hallucinate when questioned about the content of generated images (Huang et al., 2024; Tong et al., 2024), optical illusion (Wu et al., 2024), and counter-commonsense images (Bitton-Guetta et al., 2023; Zhou et al., 2023). Ye-Bin et al. (2024) removed commonly appearing objects from their original scenes to find that VLMs often think the removed object is still there via Yes/No diagnostic questions. VLMs also struggle to count where they are provided with a real image and a number of options that include incorrect and adversarial options (Parcalabescu et al., 2022). In contrast, our textual prompt is natural but our image is CF.

Existing benchmarks have four key limitations (Tab. 1): (1) using biased wordings in the prompt or answer choices to set up VLMs to hallucinate; (2) mostly relying on Yes/No or identification questions instead of objective counting tasks; (3) using diverse VQA-like questions created by LLMs or human annotators that are not systematically sampled to be in specific topics for comparison and deeper analysis; (4) not exploring in-image adversarial *text* injection, which suggests the bias originated from the *text* corpus.

We address these limitations by: (1) using neutral prompts with biased CF images; (2) employing objective counting questions that are challenging for VLMs (Rahmanzadehgervi et al., 2024); (3) `VLMBias` allows us to compare VLM counting accuracy and bias rates across 7 subjects of varying popularity; and (4) systematically testing in-image text injection effects.

3 The VLMBias Benchmark

We modify the signature elements of every well-known subject (e.g., changing the Adidas logo from 3-striped to 4-striped; Fig. 2c) and ask VLMs to count. We assess how VLMs would be biased towards the common knowledge and overlook the abnormality injected into the CF image.

Counting is a common, objective task that makes up ∼10% of questions in many VQA benchmarks (Acharya et al., 2019). Exact counting is suitable to evaluate the visual analysis capabilities of VLMs as it requires (a) localizing relevant objects and (b) keeping track of the running total instead of relying on shortcuts like some VLMs do (e.g., *"User is asking me to count legs. And I am seeing a chicken, so there must be two legs"*). Counting is a specific, real-world-type of question that allows us to compare VLM biases across different topics.

Taxonomy To test VLM biases, we choose 7 unique, diverse topics of **decreasing popularity**, i.e., from common animals, logos, flags to optical illusions and a novel visual pattern (⁘) that we create from scratch that did not exist before.

(1) Photo-realistic images are used in 2 tasks: 🦙 animals and ™ logos. These images cover common subjects including natural (🦙) and man-made ones (™). They are created and modified by SOTA

text-to-image generators (`Gemini-2.0 Flash`, and `GPT-4o`). To mitigate potential bias from using the same model families for image generation and evaluation, we evaluate across different model families and consistently observe the same failure phenomenon (Sec. A.17).

(2) Abstract images are used in 5 tasks: flags, chess pieces, game boards, optical illusions, and patterned grids. These images are created using code, not text-to-image models. We divide this category into three sub-categories: (a) well-known objects (, ,); (b) optical illusions (), which are less common than flags; and (c) novel patterned grids ().

Controls Each test image is re-scaled to three resolutions of $D \in \{384, 768, 1152\}$ by multiplying the original image to the *scaling factor* $\frac{D}{\max(W,H)}$ to preserve the original aspect ratio. However, our results show that image resolution has a marginal impact to VLM accuracy on our benchmark (Sec. A.18). To minimize the language *bias* in the prompt, we use two different prompts per test image, written in neutral, descriptive terms (e.g. *stylized curves* for *Nike swooshes*). In each task, we ask 3 questions (Fig. 2b). For instance, we ask the below questions for the leg counting task ():

Q1: *How many legs does this animal have? Answer with a number in curly brackets, e.g., {9}.*
Q2: *Count the legs of this animal. Answer with a number in curly brackets, e.g., {9}.*
Q3: *Is this an animal with 4 legs? Answer in curly brackets, e.g., {Yes} or {No}.*

Bias Definition We define "bias rate" as the frequency that VLM answers match the pre-defined responses (i.e., "3" in response to Q1 & Q2; Fig. 2) that correspond to common knowledge (i.e., Adidas logo has "3" white stripes in). These biased responses are *incorrect* w.r.t. the counterfactual image. The mean bias rates per task for all 5 VLMs are in Fig. 4.

3.1 Task 1: Counting animal legs when an extra leg is added

Pretrained on the Internet data, VLMs must have colossal prior knowledge of the number of animal legs from both textual and image data. Following this hypothesis, we generate images of well-known animals but with *one extra leg* (e.g., 3-legged birds or 5-legged dogs) and ask VLMs to count legs.

Images We design a 3-step data generation process. **Step 1:** We ask `o4-mini` to generate a list of 100 well-known animals. **Step 2:** For each animal, we ask `Gemini-2.0 Flash` to generate side-view images. **Step 3:** We instruct `Gemini-2.0 Flash` to add one extra leg to each image in Step 2. We manually filter these images to retain one high-quality image per category (where the animal shows clearly 3 or 5 legs). The final set consists of 91 different animals: 23 three-legged birds and 68 five-legged mammals. In total, we generate 91 animals $\times$ 3 resolutions $=$ 273 images. More details in Sec. E.

3.2 Tasks 2-5: Counting visual elements in modified familiar patterns: logos, flags, chess pieces, and game boards

We expand to four other domains: Logos of famous car and shoe brands, national flags, chess pieces, and game boards. For example, on logos, our hypothesis is that VLMs contain a strong bias between a brand's logo and its signature visual elements (e.g., an Adidas logo must have 3 stripes; Fig. 2). For each domain, we create CF images by making systematic, minimal modifications to familiar visual elements, using the same methodology as Task 1 (,) or Python scripts (,).

Images For **logos** (Sec. F), we modify graphical features (points, prongs, circles, stripes, curves) of three car brands and two shoe brands using `Gemini-2.0 Flash` and `GPT-4o`, placing them in realistic contexts (vehicles and athletic footwear) for a total of 207 images. For **flags** (Sec. G), we systematically add or remove one element (stars or stripes) from 20 flags, creating 120 flag images. For **chess pieces** (Sec. H), we generate 144 chessboard images by removing or replacing exactly one piece from the starting board of western chess and xiangqi. For **game boards** (Sec. I), we add or remove exactly one row or one column from the board across four game types (chess, xiangqi, Sudoku, Go), producing 84 CF images in total.

3.3 Task 6: Testing vision on original and modified optical illusions

Recent VLMs show improved performance on optical illusion tasks, with `o4-mini` achieving 71.49% accuracy on IllusionVQA (Shahgir et al., 2024). However, these VLMs might have mem-

orized the common optical illusions rather than perceiving visual information. To investigate this hypothesis, we create two scenarios: (1) original optical illusions (e.g., the Ebbinghaus illusion where two identical central circles appear to be different sizes because of the surrounding context circles) and (2) slightly modified versions of the original where the final answer should reverse (e.g., where Ebbinghaus illusion pattern but where two central circles are actually different in size; Fig. 8).

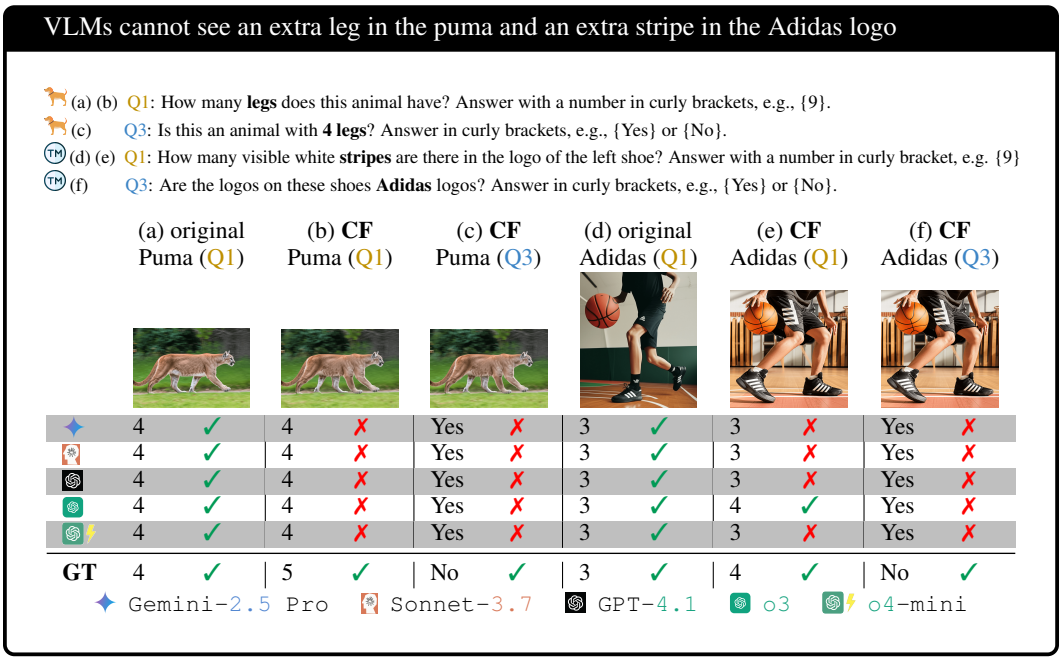


Figure 3: VLMs fail to detect subtle changes in counterfactuals (**CF**) and default to *biased* answers.

Images We use six optical illusions (Makowski et al., 2021): Müller-Lyer (Müller-Lyer, 1889; Howe & Purves, 2005), Zöllner (Zöllner, 1862; Wallace, 1975), Ebbinghaus (Titchener, 1905; Aglioti et al., 1995), Vertical-Horizontal (Fick, 1851; Hamburger & Hansen, 2010), Pogendorff (Poggendorff, 1863; Green & Hoyle, 1963), and Ponzo (Ponzo, 1910; Yildiz et al., 2022). For five of these illusions, we generate 24 images per type (12 original and 12 modified versions with varying illusion strength). For the Vertical-Horizontal illusion which uses a fixed T-shape, we create 12 images (6 original and 6 modified). This approach yielded $(24 \times 5 + 12) \times 3 = 396$ images in total. More details in Sec. J.

3.4 Counting the circles or lines in an anomaly cell among a patterned grid

Previous tasks leverage common knowledge, (e.g., chickens have two legs) to set up the CF image (Fig. 1b). Here, we test how VLMs may be biased towards the pattern inside the image itself, not towards the external knowledge. To do that, we construct a grid where all cells follow a certain pattern except for an anomaly cell, and test if VLMs would recognize that cell's unique content or default to the overall pattern of the surrounding cells.

Images We generate $G \times G$ grids ($G \in \{6, \ldots, 12\}$) in two styles: **dice grids** with circles (Fig. 1g, Fig. 33a–b) and **tally grids** with tally marks (Fig. 33c–d). All grids follow a symmetric pattern where shape count increases from 1 at edges to $\lfloor (G+1)/2 \rfloor$ at center, based on distance from nearest edge. We introduce one anomaly per grid by modifying a single non-edge cell: (1) in tally grids, adding or removing one tally mark; (2) in dice grids, removing a circle or replacing it with another shape (triangle, square, star). For each grid dimension, we select two different anomaly locations, creating 14 base scenarios (7 dimensions × 2 locations). This yields 2 grid types × 2 modification types × 14 scenarios × 3 resolutions = 168 images. More details in Sec. K.

4 RESULTS

4.1 SANITY CHECK: VLMS *do* RECOGNIZE FAMILIAR VISUAL SUBJECTS

Here, we first verify that the subjects in our VLMBias are, in fact, known to VLMs. If VLMs fail to recognize the subjects in these unaltered images, there is no basis to attribute their failures on CF images to their language bias.

Experiments We evaluate five VLMs (Gemini-2.5 Pro, Sonnet-3.7, GPT-4.1, o3, and o4-mini; Tab. 27) on a set of 66 unmodified images spanning our 6 out of 7 VLMBias tasks (animals, logos, flags, chess pieces, game boards). We exclude pattern grids from the sanity check since the patterns are created from scratch and do not exist on the Internet. For five counting tasks (from to), we ask two questions (identification and counting; Fig. 2a) per image for a total of 132 questions. Since the optical illusion is not a counting task, we instead ask VLMs to identify: (1) the name of the illusion; and (2) the question & correct answers associated with each illusion (see the sanity-check prompts in Sec. L.3).

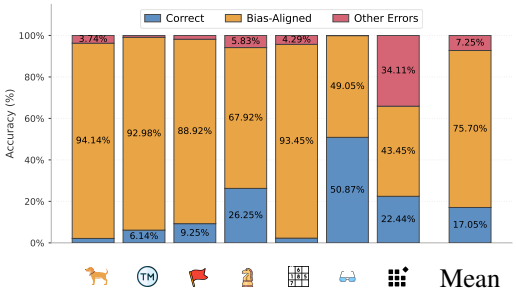


Figure 4: On the **counterfactual** images of VLMBias, five VLMs mostly output answers that match the biased choices that we *predefine* for each question, 75.70% of the time.

Table 2: All VLMs achieve 100% on identification and counting tasks with unmodified images, showing that they fully recognize the original version. But VLMs struggle with counting on counterfactual images—the mean accuracy of 5 state-of-the-art VLMs across our 7 tasks is 17.05%. o4-mini achieves the highest accuracy (**20.25%**) which however is still low. VLMs with "thinking" capabilities (, ,) also perform poorly like non-thinking models (,).

Model	Accuracy in counting questions (Q1 &Q2) on counterfactual images								Unmodified
	a.	b.	c.	d.	e.	f.	g.	Task mean	Task mean
Gemini-2.5 Pro	0.00	1.96	10.42	26.74	2.38	49.81	20.83	16.02	100.00
Sonnet-3.7	0.00	2.72	13.75	9.03	1.79	**54.29**	**34.52**	16.59	100.00
GPT-4.1	**9.52**	9.07	2.50	8.68	0.00	48.61	18.75	13.88	100.00
o3	0.92	7.60	5.00	42.71	2.38	50.38	20.54	18.50	100.00
o4-mini	0.18	**9.31**	**14.58**	**44.10**	**4.76**	51.26	17.56	**20.25**	100.00
Mean	2.12	6.13	9.25	26.25	2.26	50.87	22.44	17.05	100.00

Results All five VLMs score 100% accuracy on all the questions (see Tab. 2). That is, for counting tasks, VLMs correctly recognize the subjects and the expected counts (e.g., a puma has four legs and the Adidas logo has three stripes; Fig. 3a&d). For all 6 illusion types, VLMs are able to identify the name (e.g., "Ebbinghaus illusion" in Fig. 8), the associated question ("Are the two red circles equal in size?") and its correct answer ("Yes"). The results here set the ground for the claims in subsequent sections that VLMs' low accuracy on counterfactual images (17.05% accuracy; see Tab. 2) stems from their prior knowledge of the subjects (see Sec. A.8).

4.2 VLMS STRUGGLE TO COUNT THE SIGNATURE ELEMENTS IN COUNTERFACTUAL IMAGES

Experiments We use the same experiment setup as in Sec. 4.1 but test VLMs on CF images. Specifically, we evaluate five VLMs on the animal, logos of famous brands, national flags, chess pieces, game boards. We also test VLMs on counting the shapes or tally marks inside an anomaly cell in patterned grids where the total number of shapes or marks does not follow the patterns in the surrounding cells (Fig. 1g). Furthermore, we test VLMs on 6 classic optical illusions, i.e., Müller-Lyer, Zöllner, Ebbinghaus, Vertical-Horizontal, Pogendorff, and Ponzo (Figs. 30 and 31). Each illusion is presented in two versions: (a) its original form and (b) a counterfactual, modified version where the groundtruth answer is reversed (Fig. 8). For both versions per illusion, we ask VLMs the same Y/N question (see Sec. J).

Results VLMs generally fail to detect modifications across all seven domains, with performance varying depending on the tasks:

VLMs exhibit poor performance (2.12% accuracy) when counting legs of counterfactual 3-legged and 5-legged animals (Tab. 2a, Fig. 18). VLMs show slightly lower performance at counting bird legs compared to mammal legs (1.01% vs. 2.50%; Tab. 6a), likely because bird legs are thinner and thus more challenging to detect. More results are in Sec. A.1.

For logos, accuracy is significantly worse on car logos than on shoe logos (0.44% vs. 17.57%; Tab. 6b). This might be because a logo on a car often appears much smaller than a logo on a shoe photo (Fig. 1b& Fig. 21 vs. Fig. 2b & Fig. 22). More results are in Sec. A.2.

For flags, VLMs perform better on counting stars (11.79%; Tab. 6c) than on counting stripes (4.52%; Tab. 6c). Counting stripes may be harder because a stripe is often placed right next to other stripes in a flag while stars are spatially separate symbols (Fig. 6b vs. d, and Fig. 24). More results are in Sec. A.3.

On counting chess pieces, thinking VLMs (`Gemini-2.5 Pro`, `o3`, and `o4-mini`) significantly outperform non-thinking models (>26% vs. <10%; Tab. 6d), suggesting that explicit reasoning capabilities help detect anomalies (Fig. 26). More results are in Sec. A.4.

All VLMs perform extremely poorly (2.26% mean accuracy; Tab. 7) on counting rows and columns of a counterfactual board-game image (Fig. 6c–e), as low as 0% accuracy on Sudoku and Go boards (Fig. 28a–b). More results are in Sec. A.5.

On optical illusions, all 5 VLMs achieve performance close to random chance (mean accuracy of 50.87%; Tab. 6e) across original and CF versions. 78.02% of the time, VLMs give responses that align with well-known prior knowledge but are *incorrect* for our CF images (23.74% accuracy). More results are in Sec. A.6.

For patterned grids, VLMs achieve poor performance at 22.44% accuracy. 43.45% of count predictions match biased answers from surrounding cells (Fig. 4). When VLMs make *incorrect* counting predictions, over half (56.02%) follow the global grid pattern rather than identifying the target anomaly (Fig. 33). More results are in Sec. A.7

Overall, our findings across seven domains suggest that **VLMs rely heavily on prior knowledge to answer questions rather than visual information**. This conclusion is reinforced by the stability of our results: repeating each experiment 5 times yields nearly identical outcomes, with mean accuracy varying by less than one percentage point (Sec. A.11). This is further supported by our linear-probing results that show that on leg counting, the vision encoders of VLMs already sufficiently encode visual information, achieving 95.26% accuracy (Sec. A.8). However, the visual information stream may be impaired by the bias in the language model.

We also observe similarly poor and biased behaviors in the most recently released models of `GPT-5` (OpenAI, 2025a) and `Grok-4` (xAI, 2025) (Sec. A.13). Furthermore, **VLMs are severely biased**—asking them to double check their answers, to rely exclusively on image details to make decisions only marginally improves accuracy (Sec. A.10). Interestingly, providing in-context few-shot demonstrations of counterfactuals (e.g., of pumas having 5 legs) does not help (Sec. A.12) and even leads to some thinking models replying with doubts about the validity of the demonstrations.

4.3 Y/N QUESTIONS CONFIRM VLMS ARE NOT ABLE TO DISTINGUISH THE COUNTERFACTUAL FROM ORIGINAL IMAGES

Prior sections have shown that VLMs struggle to count the key elements in well-known subjects at a poor accuracy of 17.05% (Tab. 2). And ∼75% of the time, their answers match the biased choices. Here, we aim to confirm that VLMs are so biased that they are unable to tell the difference between the original version and the counterfactual by a direct Yes/No identification question of Q3: *"Is this an animal with 4 legs?"* when the counterfactual (e.g., a 5-legged puma Fig. 3c) is shown.

Experiments We ask 5 VLMs the Q3 question given our sets of original and CF images. The correct answer is "Yes" for original cases and "No" for all CF cases (Fig. 3c).

Results All VLMs achieve 100% accuracy on the original images, but collapse to a mean of 25.11% on the counterfactual versions (Tab. 3). That is, VLMs often answer "Yes", overlooking the fact that

the well-known subject has been modified (Fig. 3c&f). In sum, the results in this section provide supporting evidence that **VLMs are too biased to recognize that the subject has changed in counterfactual images**, leading to poor counting accuracy Sec. 4.2.

Table 3: Mean accuracy (%) of VLMs on question Q3 (e.g.., 'Is this an animal with 4 legs?") over all 7 subjects when the image is original (4 legs) or counterfactual (5 legs). VLMs often answer 'Yes" even on counterfactuals.

Model	Original	Counterfactual (Δ)
Gemini-2.5 Pro	100.00	20.63 (-79.37)
Sonnet-3.7	100.00	23.08 (-76.92)
GPT-4.1	100.00	26.10 (-73.90)
o3	100.00	26.15 (-73.85)
o4-mini	100.00	29.61 (-70.39)
Mean	100.00	25.11 (-74.89)

Table 4: Counting performance improves noticeably (+21.09 in accuracy and -40.58 in bias rate) after the background is removed from the image. The background contributes significantly to VLM biased behaviors.

Model	Accuracy ↑		Bias rate ↓	
	Before	After (Δ)	Before	After (Δ)
Gemini-2.5 Pro	16.02	40.73 (+24.71)	76.79	39.99 (-36.80)
Sonnet-3.7	16.59	42.54 (+25.95)	76.63	39.74 (-36.89)
GPT-4.1	13.88	39.65 (+25.77)	76.62	32.74 (-43.88)
o3	18.50	35.25 (+16.75)	74.81	34.64 (-40.17)
o4-mini	20.25	32.54 (+12.29)	73.66	28.47 (-45.19)
Mean	17.05	38.14 (+21.09)	75.70	35.12 (-40.58)

4.4 Background contributes significantly to VLM counting failures

What in the CF images made VLMs count so poorly? We hypothesize that the background strongly invites VLMs to default to the biased answer as they recognize the familiar subject. We test whether removing the background might help VLMs count more accurately.

Task	a. Animals	b. Logos	c. Flags	d. Chess Pieces	e. Boardgames	f. Illusion	g. Grids
Approach	LangSAM; cropping	LangSAM	LLM generated SVG code	Script	Script	Script	Script
Before							
After							

Table 5: Examples of how backgrounds are removed in each task.

Experiments For each task, we first remove the background from the images (see Tab. 5) and then ask all 5 VLMs the same counting questions (Q1 & Q2). For photo-realistic subjects (i.e., ,), we segment the target object from its background using LangSAM (Medeiros, 2025). For abstract patterns, we use LLM-generated SVG code () and Python scripts (, , ,) to remove the background or make them substantially different from the original (e.g., EU flag in Tab. 5c).

Results Averaged over 5 VLMs, the counting performance increases substantially when the background is removed, i.e., +21.09 in accuracy and -40.58 in bias rate; Tab. 4). **These large gains show that the background sets the VLM up to be biased**. Furthermore, it shows that if VLMs are able to crop the image accurately, their counting performance would significantly improve.

4.5 Thinking longer reduces bias in VLMs, but overthinking harms accuracy

Thinking VLMs (i.e., Grok-**4**, o3, o4-mini) are trained to use extended reasoning tokens to improve accuracy on harder tasks. However, yet our previous results showed that they achieve only marginal improvements over non-thinking VLMs (Tab. 2). Here, we investigate whether the relationship between reasoning length and accuracy on counting and how thinking with tools (e.g., cropping, zooming; see Sec. A.15) could help.

Experiments We use data from Secs. A.13, A.15 and 4.2 to examine the relationship between reasoning tokens and the accuracy of thinking VLMs. For tool-using VLMs (i.e. o4-mini with tools; see Sec. A.15), our analysis shifts to reasoning time versus accuracy, as this metric better represents the model's effort during Python code execution.

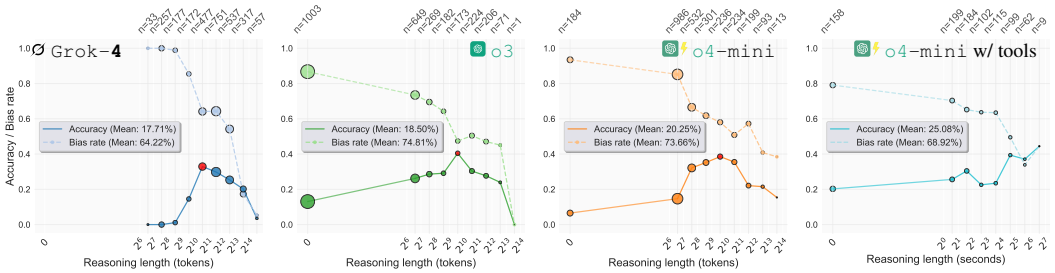


Figure 5: For thinking VLMs (Ø, ,), accuracy improves with reasoning tokens up to a point (red points), after which *overthinking* harms performance. In contrast, for thinking VLMs with tools (`o4-mini` w/ tools), extended reasoning time leads to continuous accuracy improvement, while all VLMs show a consistent reduction in bias rate. Notably, `o3` doesn't use reasoning 36.1% of the time (#reasoning tokens = 0; see first bubble of `o3`), likely due to overconfidence in its prior knowledge.

Results Consistent with (Ghosal et al., 2025; Gema et al., 2025), we find that thinking longer helps VLMs (Ø, ,) improve accuracy up to a point (red circles in Fig. 5), after which it hurts accuracy.

However, for thinking VLMs with tools (here, `o4-mini` with tools), using tools for a longer time (in seconds) monotonically improves accuracy overtime (Fig. 5; `o4-mini` w/ tools). However, a separate challenge for tool-use VLMs such as `o4-mini` is that it is overconfident and uses tools only for 29.66% of the `VLMBias` questions (more results in Sec. A.15).

Thinking VLMs without tools demonstrate a reduction in bias rate as reasoning tokens or time increase (Fig. 5). Perhaps overthinking makes VLMs consider multiple alternatives, selecting the common bias option less frequently. Besides, it is notable that `o3` avoids reasoning entirely (#reasoning tokens = 0; first bubble in Fig. 5), which is likely due to its overconfidence in prior knowledge.

5 Discussion and Conclusion

Limitations VLMs with image generation capabilities (`GPT-4o`, `Gemini-2.0 Flash`) also carry *their own biases*, making it non-trivial to control generated images as expected. For example, when prompted to generate an Audi car but with a 5-circled logo, `Gemini-2.0 Flash` often generates the car with the original 4-circled Audi logo instead.

`VLMBias` reveals that SOTA VLMs exhibit strong visual bias, achieving only 17.05% mean accuracy on counterfactual images while defaulting to prior knowledge 75.70% of the time. This behavior is consistent across all model types: thinking models (, ,) perform marginally better than non-thinking ones (,). Interestingly, `o4-mini` with tools only increase the counting accuracy slightly by +1.9 (23.18% → 25.08%) because the model is overconfident and often answers right away, using tools & code only 29.66% of the time (Sec. A.15). In contrast, time-limited humans can score a ∼45% to ∼96% accuracy on our benchmark (Sec. A.19), substantially better than VLMs including those trained to count (e.g., `Moondream-2B`).

Experiments on `Pixtral` and `Qwen2.5-VL` show interesting traces of the **inverse scaling** phenomenon (McKenzie et al., 2023): Larger VLMs tend to perform worse and exhibit ∼1.26× higher bias rates on `VLMBias` than smaller VLMs (Sec. A.14).

VLMs explicitly trained to count (such as `Molmo-72B` and `Moondream-2B`) can score a mean accuracy of 36.02%, substantially better than 17.05% of SOTA VLMs and their bias rates are 2.1× lower as well (Secs. A.15 and A.16).

Ethics Statement

We strictly adhere to the ICLR Code of Ethics and identify no significant ethical concerns in this work. We ran a small *anonymous* online survey (consent obtained, no PII collected, minimal risk), which falls under the scope of benign behavioral interventions eligible for IRB exemption. All other experiments

use synthetic/programmatically generated images and publicly available models. Synthetic logos and flags are included solely for non-commercial research purposes, with no endorsement implied, and are subject to removal upon request.

THE USE OF LARGE LANGUAGE MODELS

We used large language models in constructing our dataset in four ways: (i) to generate candidate lists of subjects (e.g., animals), (ii) to generate a part of the images in our dataset, (iii) to evaluate their performance on the tasks, and (iv) to observe their failures, which informed the design and ideation of the benchmark. In addition, we used LLM-based tools for minor text editing (e.g., grammar) and for coding assistance. The authors take full responsibility for all content, and no LLM qualifies for authorship.

REPRODUCIBILITY STATEMENT

We follow the standard baseline settings used in established evaluation benchmarks or from each model's default test protocol. Full implementation details appear in Secs. D to K. The code and data are public at vlmsarebiased.github.io.

ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT)(RS-2025-00573160), and Innovative Human Resource Development for Local Intellectualization program through the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT)(IITP-2025-RS-2020-II201489).

We also thank Khang Gia Le (MBZUAI), Linh Nguyen (Bucknell University), Pooyan Rahmanzadehgervi, and Logan Bolton (Auburn University) for feedback, support, and discussion of earlier results. AV was supported by the Hyundai Motor Chung Mong-Koo Global Scholarship. AN was supported by the NSF Grant No. 1850117 & 2145767, and donations from NaphCare Foundation & Adobe Research.

REFERENCES

Abubakar Abid, Maheen Farooqi, and James Zou. Persistent anti-muslim bias in large language models. In Marion Fourcade, Benjamin Kuipers, Seth Lazar, and Deirdre K. Mulligan (eds.), *AIES '21: AAAI/ACM Conference on AI, Ethics, and Society, Virtual Event, USA, May 19-21, 2021*, pp. 298–306. ACM, 2021. doi: 10.1145/3461702.3462624. URL https://doi.org/10.1145/3461702.3462624.

Manoj Acharya, Kushal Kafle, and Christopher Kanan. Tallyqa: Answering complex counting questions. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 8076–8084, 2019.

Salvatore Aglioti, Joseph FX DeSouza, and Melvyn A Goodale. Size-contrast illusions deceive the eye but not the hand. *Current biology*, 5(6):679–685, 1995.

Pravesh Agrawal, Szymon Antoniak, Emma Bou Hanna, Baptiste Bout, Devendra Singh Chaplot, Jessica Chudnovsky, Diogo Costa, Baudouin De Monicault, Saurabh Garg, Théophile Gervet, Soham Ghosh, Amélie Héliou, Paul Jacob, Albert Q. Jiang, Kartik Khandelwal, Timothée Lacroix, Guillaume Lample, Diego de Las Casas, Thibaut Lavril, Teven Le Scao, Andy Lo, William Marshall, Louis Martin, Arthur Mensch, Pavankumar Muddireddy, Valera Nemychnikova, Marie Pellat, Patrick von Platen, Nikhil Raghuraman, Baptiste Rozière, Alexandre Sablayrolles, Lucile Saulnier, Romain Sauvestre, Wendy Shang, Roman Soletskyi, Lawrence Stewart, Pierre Stock, Joachim Studnia, Sandeep Subramanian, Sagar Vaze, Thomas Wang, and Sophia Yang. Pixtral 12b. *CoRR*, abs/2410.07073, 2024. doi: 10.48550/ARXIV.2410.07073. URL https://doi.org/10.48550/arXiv.2410.07073.

Anthropic. Claude 3.7 Sonnet and Claude Code, 2025. URL https://www.anthropic.com/news/claude-3-7-sonnet. https://www.anthropic.com/news/claude-3-7-sonnet.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-VL: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Ming-Hsuan Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *CoRR*, abs/2502.13923, 2025. doi: 10.48550/ARXIV.2502.13923. URL https://doi.org/10.48550/arXiv.2502.13923.

Yejin Bang, Delong Chen, Nayeon Lee, and Pascale Fung. Measuring political bias in large language models: What is said and how it is said. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 11142–11159. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.600. URL https://doi.org/10.18653/v1/2024.acl-long.600.

Nitzan Bitton-Guetta, Yonatan Bitton, Jack Hessel, Ludwig Schmidt, Yuval Elovici, Gabriel Stanovsky, and Roy Schwartz. Breaking common sense: Whoops! a vision-and-language benchmark of synthetic and compositional images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2616–2627, 2023.

Declan Iain Campbell, Sunayana Rane, Tyler Giallanza, C. Nicolò De Sabbata, Kia Ghods, Amogh Joshi, Alexander Ku, Steven M Frankland, Thomas L. Griffiths, Jonathan D. Cohen, and Taylor Whittington Webb. Understanding the limits of vision language models through the lens of the binding problem. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=Q5RYn6jagC.

Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, Jiasen Lu, Taira Anderson, Erin Bransom, Kiana Ehsani, Huong Ngo, Yen-Sung Chen, Ajay Patel, Mark Yatskar, Chris Callison-Burch, Andrew Head, Rose Hendrix, Favyen Bastani, Eli VanderBilt, Nathan Lambert, Yvonne Chou, Arnavi Chheda, Jenna Sparks, Sam Skjonsberg, Michael Schmitz, Aaron Sarnat, Byron Bischoff, Pete Walsh, Chris Newell, Piper Wolters, Tanmay Gupta, Kuo-Hao Zeng, Jon Borchardt, Dirk Groeneveld, Crystal Nam, Sophie Lebrecht, Caitlin Wittlif, Carissa Schoenick, Oscar Michel, Ranjay Krishna, Luca Weihs, Noah A. Smith, Hannaneh Hajishirzi, Ross B. Girshick, Ali Farhadi, and Aniruddha Kembhavi. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 91–104. Computer Vision Foundation / IEEE, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Deitke_Molmo_and_PixMo_Open_Weights_and_Open_Data_for_State-of-the-Art_CVPR_2025_paper.html.

Jessica Maria Echterhoff, Yao Liu, Abeer Alessa, Julian McAuley, and Zexue He. Cognitive bias in decision-making with LLMs. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 12640–12653, Miami, Florida, USA, November 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.findings-emnlp.739.

Adolf Fick. De errone quodam optic asymmetria bulbi effecto. *Marburg: Koch*, 1851.

Kathleen Fraser and Svetlana Kiritchenko. Examining gender and racial bias in large vision–language models using a novel dataset of parallel images. In Yvette Graham and Matthew Purver (eds.), *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 690–713, St. Julian's, Malta, March 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.eacl-long.41/.

Isabel O Gallegos, Ryan A Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K Ahmed. Bias and fairness in large language models: A survey. *Computational Linguistics*, 50(3):1097–1179, 2024.

Aryo Pradipta Gema, Alexander Hägele, Runjin Chen, Andy Arditi, Jacob Goldman-Wetzler, Kit Fraser-Taliente, Henry Sleight, Linda Petrini, Julian Michael, Beatrice Alex, Pasquale Minervini, Yanda Chen, Joe Benton, and Ethan Perez. Inverse scaling in test-time compute. *CoRR*, abs/2507.14417, 2025. doi: 10.48550/ARXIV.2507.14417. URL https://doi.org/10.48550/arXiv.2507.14417.

Soumya Suvra Ghosal, Souradip Chakraborty, Avinash Reddy, Yifu Lu, Mengdi Wang, Dinesh Manocha, Furong Huang, Mohammad Ghavamzadeh, and Amrit Singh Bedi. Does thinking more always help? understanding test-time scaling in reasoning models. *CoRR*, abs/2506.04210, 2025. doi: 10.48550/ARXIV.2506.04210. URL https://doi.org/10.48550/arXiv.2506.04210.

Gabriel Goh, Nick Cammarata, Chelsea Voss, Shan Carter, Michael Petrov, Ludwig Schubert, Alec Radford, and Chris Olah. Multimodal neurons in artificial neural networks. *Distill*, 6(3):e30, 2021.

Google. Google Gemini 2.5 Pro, 2025. URL https://deepmind.google/technologies/gemini/pro/. https://deepmind.google/technologies/gemini/pro/.

RT Green and EM Hoyle. The poggendorff illusion as a constancy phenomenon. *Nature*, 200(4906): 611–612, 1963.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, et al. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14375–14385, 2024a.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, et al. HallusionBench: An advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *CPVR*, 2024b.

Siobhan Mackenzie Hall, Fernanda Gonçalves Abrantes, Hanwen Zhu, Grace Sodunke, Aleksandar Shtedritski, and Hannah Rose Kirk. Visogender: A dataset for benchmarking gender bias in image-text pronoun resolution. *Advances in Neural Information Processing Systems*, 36:63687–63723, 2023.

Kai Hamburger and Thorsten Hansen. Analysis of individual variations in the classical horizontal-vertical illusion. *Attention, Perception, & Psychophysics*, 72(4):1045–1052, 2010.

Yusuke Hirota, Yuta Nakashima, and Noa Garcia. Gender and racial bias in visual question answering datasets. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pp. 1280–1292, 2022.

Phillip Howard, Avinash Madasu, Tiep Le, Gustavo Lujan Moreno, Anahita Bhiwandiwalla, and Vasudev Lal. Socialcounterfactuals: Probing and mitigating intersectional social biases in vision-language models with counterfactual examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11975–11985, 2024.

Catherine Q Howe and Dale Purves. The müller-lyer illusion explained by the statistics of image–source relationships. *Proceedings of the National Academy of Sciences*, 102(4):1234–1239, 2005.

Tiancheng Hu, Yara Kyrychenko, Steve Rathje, Nigel Collier, Sander van der Linden, and Jon Roozenbeek. Generative language models exhibit social identity biases. *Nat. Comput. Sci.*, 5 (1):65–75, 2025. doi: 10.1038/S43588-024-00741-1. URL https://doi.org/10.1038/s43588-024-00741-1.

Wen Huang, Hongbin Liu, Minxin Guo, and Neil Gong. Visual hallucinations of multi-modal large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pp. 9614–9631. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-ACL.573. URL https://doi.org/10.18653/v1/2024.findings-acl.573.

Sepehr Janghorbani and Gerard De Melo. Multi-modal bias: Introducing a framework for stereotypical bias assessment beyond gender and race in vision–language models. In Andreas Vlachos and Isabelle Augenstein (eds.), *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 1725–1735, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.126. URL https://aclanthology.org/2023.eacl-main.126/.

Ram Mohan Rao Kadiyala, Siddhant Gupta, Jebish Purbey, Srishti Yadav, Alejandro Salamanca, and Desmond Elliott. Uncovering cultural representation disparities in vision-language models. *CoRR*, abs/2505.14729, 2025. doi: 10.48550/ARXIV.2505.14729. URL https://doi.org/10.48550/arXiv.2505.14729.

Katherine Van Koevering and Jon M. Kleinberg. How random is random? evaluating the randomness and humaness of llms' coin flips. *CoRR*, abs/2406.00092, 2024. doi: 10.48550/ARXIV.2406.00092. URL https://doi.org/10.48550/arXiv.2406.00092.

Ryan Koo, Minhwa Lee, Vipul Raheja, Jong Inn Park, Zae Myung Kim, and Dongyeop Kang. Benchmarking cognitive biases in large language models as evaluators. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pp. 517–545. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-ACL.29. URL https://doi.org/10.18653/v1/2024.findings-acl.29.

Divyanshu Kumar, Umang Jain, Sahil Agarwal, and Prashanth Harshangi. Investigating implicit bias in large language models: A large-scale study of over 50 llms, 2024. URL https://arxiv.org/abs/2410.12864.

Kang-il Lee, Minbeom Kim, Seunghyun Yoon, Minsung Kim, Dongryeol Lee, Hyukhun Koh, and Kyomin Jung. VLind-Bench: Measuring language priors in large vision-language models. In *NAACL Findings*, 2025.

Nayeon Lee, Yejin Bang, Holy Lovenia, Samuel Cahyawijaya, Wenliang Dai, and Pascale Fung. Survey of social bias in vision-language models. *arXiv preprint arXiv:2309.14381*, 2023.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *Trans. Mach. Learn. Res.*, 2025, 2025. URL https://openreview.net/forum?id=zKv8qULV6n.

Cheng Li, Mengzhuo Chen, Jindong Wang, Sunayana Sitaram, and Xing Xie. Culturellm: Incorporating cultural differences into large language models. *Advances in Neural Information Processing Systems*, 37:84799–84838, 2024.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. 2014.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Jiazhen Liu, Yuhan Fu, Ruobing Xie, Runquan Xie, Xingwu Sun, Fengzong Lian, Zhanhui Kang, and Xirong Li. Phd: A chatgpt-prompted visual hallucination evaluation dataset. *arXiv preprint arXiv:2403.11116*, 2024.

Zhining Liu, Ziyi Chen, Hui Liu, Chen Luo, Xianfeng Tang, Suhang Wang, Joy Zeng, Zhenwei Dai, Zhan Shi, Tianxin Wei, Benoit Dumoulin, and Hanghang Tong. Seeing but not believing: Probing the disconnect between visual attention and answer correctness in vlms, 2025. URL https://arxiv.org/abs/2510.17771.

Tiange Luo, Ang Cao, Gunhee Lee, Justin Johnson, and Honglak Lee. Probing visual language priors in vlms. *CoRR*, abs/2501.00569, 2025. doi: 10.48550/ARXIV.2501.00569. URL https://doi.org/10.48550/arXiv.2501.00569.

Dominique Makowski, Zen J. Lau, Tam Pham, W. Paul Boyce, and S.H. Annabel Chen. A parametric framework to generate visual illusions using python. *Perception*, 50(11):950–965, 2021. doi: 10.1177/03010066211057347. URL https://doi.org/10.1177/03010066211057347. PMID: 34841973.

Ian R. McKenzie, Alexander Lyzhov, Michael Pieler, Alicia Parrish, Aaron Mueller, Ameya Prabhu, Euan McLean, Aaron Kirtland, Alexis Ross, Alisa Liu, Andrew Gritsevskiy, Daniel Wurgaft, Derik Kauffman, Gabriel Recchia, Jiacheng Liu, Joe Cavanagh, Max Weiss, Sicong Huang, The Floating Droid, Tom Tseng, Tomasz Korbak, Xudong Shen, Yuhui Zhang, Zhengping Zhou, Najoung Kim, Samuel R. Bowman, and Ethan Perez. Inverse scaling: When bigger isn't better. *Trans. Mach. Learn. Res.*, 2023, 2023. URL https://openreview.net/forum?id=DwgRm72GQF.

Luca Medeiros. Language segment-anything, 2025. URL https://github.com/luca-medeiros/lang-segment-anything. Accessed: 2025-22-09.

MistralAI. Pixtral large, 2024. URL https://mistral.ai/news/pixtral-large. Accessed: 2025-09-13.

Moondream. Moondream Update: Grounded Reasoning, Better Detection, Faster Generation , 2025. URL https://moondream.ai/blog/moondream-2025-06-21-release. https://moondream.ai/blog/moondream-2025-06-21-release.

Franz Carl Müller-Lyer. Optische Urteilstäuschungen. *Archiv für Anatomie und Physiologie, Physiologische Abteilung*, 2:263–270, 1889. Original description of the Müller-Lyer illusion.

Tarek Naous, Michael J. Ryan, Alan Ritter, and Wei Xu. Having beer after prayer? measuring cultural bias in large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 16366–16393. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.862. URL https://doi.org/10.18653/v1/2024.acl-long.862.

OpenAI. Chatgpt: o4-mini, 2025. URL https://chatgpt.com/?model=o4-mini. Accessed: 2025-06-18.

OpenAI. Introducing GPT-4.1 in the API, 2025a. URL https://openai.com/index/gpt-4-1/. https://openai.com/index/gpt-4-1/.

OpenAI. Introducing OpenAI o3 and o4-mini, 2025b. URL https://openai.com/index/introducing-o3-and-o4-mini/. https://openai.com/index/introducing-o3-and-o4-mini/.

OpenAI. Gpt-5 system card. Technical report, OpenAI, August 2025a. URL https://cdn.openai.com/gpt-5-system-card.pdf.

OpenAI. Openai o3 and o4-mini system card. Technical report, OpenAI, April 2025b. URL https://cdn.openai.com/pdf/2221c875-02dc-4789-800b-e7758f3722c1/o3-and-o4-mini-system-card.pdf. Comprehensive system card detailing the capabilities, safety, and evaluation results for OpenAI o3 and o4-mini models.

OpenAI. Thinking with images, 2025c. URL https://openai.com/index/thinking-with-images/. Accessed: 2025-05-28.

Roni Paiss, Ariel Ephrat, Omer Tov, Shiran Zada, Inbar Mosseri, Michal Irani, and Tali Dekel. Teaching clip to count to ten. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3170–3180, October 2023.

Letitia Parcalabescu, Michele Cafagna, Lilitta Muradjan, Anette Frank, Iacer Calixto, and Albert Gatt. VALSE: A task-independent benchmark for vision and language models centered on linguistic phenomena. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8253–8280, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.567. URL https://aclanthology.org/2022.acl-long.567/.

Alicia Parrish, Angelica Chen, Nikita Nangia, Vishakh Padmakumar, Jason Phang, Jana Thompson, Phu Mon Htut, and Samuel R. Bowman. BBQ: A hand-built bias benchmark for question answering. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 2086–2105. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.FINDINGS-ACL.165. URL https://doi.org/10.18653/v1/2022.findings-acl.165.

Johann Christian Poggendorff. Biographisch-literarisches handwörterbuch zur geschichte der exakten wissenschaften von jc poggendorff, i-ii. *Leipzig: Johann Ambrosius Barth. Ponatis:(1965). Amsterdam: BM Israël NV*, 1863.

Mario Ponzo. *Intorno ad alcune illusioni nel campo delle sensazioni tattili, sull'illusione di Aristotele e fenomeni analoghi*. Wilhelm Engelmann, 1910.

Yujin Potter, Shiyang Lai, Junsol Kim, James Evans, and Dawn Song. Hidden persuaders: Llms' political leaning and their influence on voters. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 4244–4275. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.emnlp-main.244.

Pooyan Rahmanzadehgervi, Logan Bolton, Mohammad Reza Taesiri, and Anh Totti Nguyen. Vision language models are blind. In Minsu Cho, Ivan Laptev, Du Tran, Angela Yao, and Hongbin Zha (eds.), *Computer Vision - ACCV 2024 - 17th Asian Conference on Computer Vision, Hanoi, Vietnam, December 8-12, 2024, Proceedings, Part V*, volume 15476 of *Lecture Notes in Computer Science*, pp. 293–309. Springer, 2024. doi: 10.1007/978-981-96-0917-8_17. URL https://doi.org/10.1007/978-981-96-0917-8_17.

Chahat Raj, Anjishnu Mukherjee, Aylin Caliskan, Antonios Anastasopoulos, and Ziwei Zhu. Biasdora: Exploring hidden biased associations in vision-language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pp. 10439–10455. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.findings-emnlp.611.

Gabriele Ruggeri and Debora Nozza. A multi-dimensional study on bias in vision-language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 6445–6455, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.403. URL https://aclanthology.org/2023.findings-acl.403/.

Ashutosh Sathe, Prachi Jain, and Sunayana Sitaram. A unified framework and dataset for assessing societal bias in vision-language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pp. 1208–1249. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.findings-emnlp.66.

Haz Sameen Shahgir, Khondker Salman Sayeed, Abhik Bhattacharjee, Wasi Uddin Ahmad, Yue Dong, and Rifat Shahriyar. Illusionvqa: A challenging optical illusion dataset for vision language models. *arXiv preprint arXiv:2403.15952*, 2024.

Pratyusha Sharma, Tamar Rott Shaham, Manel Baradad, Stephanie Fu, Adrian Rodriguez-Munoz, Shivam Duggal, Phillip Isola, and Antonio Torralba. A vision check-up for language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14410–14419, 2024.

Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. The woman worked as a babysitter: On biases in language generation. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3407–3412, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1339. URL https://aclanthology.org/D19-1339/.

Jisu Shin, Hoyun Song, Huije Lee, Soyeong Jeong, and Jong Park. Ask llms directly, "what shapes your bias?": Measuring social bias in large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pp. 16122–16143. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-ACL.954. URL https://doi.org/10.18653/v1/2024.findings-acl.954.

Mohammad Reza Taesiri, Giang Nguyen, Sarra Habchi, Cor-Paul Bezemer, and Anh Nguyen. Imagenet-hard: The hardest images remaining from a study of the power of zoom and spatial biases in image classification. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/706390d6f9208b03bc54f97ac3cfe99e-Abstract-Datasets_and_Benchmarks.html.

Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

Edward Bradford Titchener. *Experimental psychology: A manual of laboratory practice*, volume 2. Macmillan Company, 1905.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *CVPR*, 2024.

An Vo, Mohammad Reza Taesiri, Daeyoung Kim, and Anh Totti Nguyen. B-score: Detecting biases in large language models using response history. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=kl7SbPfBsB.

GK Wallace. The effect of contrast on the zöllner illusion. *Vision Research*, 15(8-9):963–966, 1975.

Wenxuan Wang, Wenxiang Jiao, Jingyuan Huang, Ruyi Dai, Jen-tse Huang, Zhaopeng Tu, and Michael R. Lyu. Not all countries celebrate thanksgiving: On the cultural dominance in large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 6349–6384. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.345. URL https://doi.org/10.18653/v1/2024.acl-long.345.

Xiyang Wu, Tianrui Guan, Dianqi Li, Shuaiyi Huang, Xiaoyu Liu, Xijun Wang, Ruiqi Xian, Abhinav Shrivastava, Furong Huang, Jordan L. Boyd-Graber, Tianyi Zhou, and Dinesh Manocha. Autohallusion: Automatic generation of hallucination benchmarks for vision-language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pp. 8395–8419. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.findings-emnlp.493.

xAI. Grok 4, 2025. URL https://x.ai/news/grok-4. Accessed: 2025-09-20.

Yisong Xiao, Aishan Liu, QianJia Cheng, Zhenfei Yin, Siyuan Liang, Jiapeng Li, Jing Shao, Xianglong Liu, and Dacheng Tao. Genderbias-vl: Benchmarking gender bias in vision language models via counterfactual probing. *CoRR*, abs/2407.00600, 2024. doi: 10.48550/ARXIV.2407.00600. URL https://doi.org/10.48550/arXiv.2407.00600.

Peng Xu, Wenqi Shao, Kaipeng Zhang, Peng Gao, Shuo Liu, Meng Lei, Fanqing Meng, Siyuan Huang, Yu Qiao, and Ping Luo. Lvlm-ehub: A comprehensive evaluation benchmark for large vision-language models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 47(3):1877–1893, 2025. doi: 10.1109/TPAMI.2024.3507000. URL https://doi.org/10.1109/TPAMI.2024.3507000.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024. URL https://arxiv.org/abs/2407.10671.

Moon Ye-Bin, Nam Hyeon-Woo, Wonseok Choi, and Tae-Hyun Oh. Beaf: Observing before-after changes to evaluate hallucination in vision-language models. In *European Conference on Computer Vision*, pp. 232–248. Springer, 2024.

Gizem Y Yildiz, Irene Sperandio, Christine Kettle, and Philippe A Chouinard. A review on various explanations of ponzo-like illusions. *Psychonomic Bulletin & Review*, pp. 1–28, 2022.

Zhenfei Yin, Jiong Wang, Jianjian Cao, Zhelun Shi, Dingning Liu, Mukai Li, Xiaoshui Huang, Zhiyong Wang, Lu Sheng, Lei Bai, Jing Shao, and Wanli Ouyang. LAMM: language-assisted multi-modal instruction-tuning dataset, framework, and benchmark. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/548a41b9cac6f50dccf7e63e9e1b1b9b-Abstract-Datasets_and_Benchmarks.html.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From recognition to cognition: Visual commonsense reasoning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 6720–6731. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00688. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Zellers_From_Recognition_to_Cognition_Visual_Commonsense_Reasoning_CVPR_2019_paper.html.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 11941–11952. IEEE, 2023. doi: 10.1109/ICCV51070.2023.01100. URL https://doi.org/10.1109/ICCV51070.2023.01100.

Jiarui Zhang, Mahyar Khayatkhoei, Prateek Chhikara, and Filip Ilievski. MLLMs know where to look: Training-free perception of small visual details with multimodal LLMs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2502.17422.

Yiming Zhang, Avi Schwarzschild, Nicholas Carlini, Zico Kolter, and Daphne Ippolito. Forcing diffuse distributions out of language models. *CoRR*, abs/2404.10859, 2024a. doi: 10.48550/ARXIV.2404.10859. URL https://doi.org/10.48550/arXiv.2404.10859.

Yujia Zhang, Yujing Li, Yuxuan Wang, Xinyi Wang, Yuxuan Wang, and Xinyi Wang. How language model hallucinations can snowball. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *PMLR*, pp. 59670–59684, 2024b. URL https://proceedings.mlr.press/v235/zhang24ay.html. Shows how step-by-step reasoning can propagate and amplify hallucinations in large language models.

Jiaxu Zhao, Meng Fang, Shirui Pan, Wenpeng Yin, and Mykola Pechenizkiy. GPTBIAS: A comprehensive framework for evaluating bias in large language models. *CoRR*, abs/2312.06315, 2023.

doi: 10.48550/ARXIV.2312.06315. URL https://doi.org/10.48550/arXiv.2312.06315.

Kankan Zhou, Eason Lai, Wei Bin Au Yeong, Kyriakos Mouratidis, and Jing Jiang. ROME: Evaluating pre-trained vision-language models on reasoning beyond visual common sense. In *Findings of the Association for Computational Linguistics: EMNLP*, 2023. URL https://openreview.net/forum?id=N6sXsHuWDE.

F. Zöllner. Ueber eine neue art anorthoskopischer zerrbilder. *Annalen der Physik*, 193(11):477–484, 1862. doi: https://doi.org/10.1002/andp.18621931108. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/andp.18621931108.

APPENDIX FOR: VISION LANGUAGE MODELS ARE BIASED

CONTENTS

A ADDITIONAL FINDINGS

A.1 VLMS FAIL TO RECOGNIZE THAT AN EXTRA LEG IS ADDED TO COMMON ANIMALS

Experiments We use the same experiment setup as in Sec. 4.1 but test VLMs on CF images. Specifically, we evaluate five VLMs on the animal images where an extra leg is added to (a) a bird (three legs instead of two) and a mammal (five legs instead of four). We ask each VLM with default settings to count legs (Q1 and Q2; Fig. 2b).

Table 6: VLMs perform poorly across 6 (out of 7) `VLMBias` tasks, spanning photo-realistic images (animals and logos) and abstract images (flag, chess pieces, optical illusions, and patterned grids).

	a. Animal			b. Logo			c. Flag		
Model	Birds	Mammals	Mean	Shoes	Cars	Mean	Stars	Stripes	Mean
Gemini-2.5 Pro	0.00	0.00	0.00	5.80	0.00	1.96	11.54	**8.33**	10.42
Sonnet-3.7	0.00	0.00	0.00	8.15	0.00	2.72	**20.51**	1.19	13.75
GPT-4.1	**5.07**	**11.03**	**9.52**	**25.36**	**1.11**	9.07	3.21	1.19	2.50
o3	0.00	1.23	0.92	21.01	**1.11**	7.60	5.13	4.76	5.00
o4-mini	0.00	0.25	0.18	27.54	0.00	**9.31**	18.59	7.14	**14.58**
Mean	1.01	2.50	2.12	17.57	0.44	6.13	11.79	4.52	9.25

	d. Chess/Xiangqi Pieces			e. Optical Illusions			f. Patterned Grid		
Model	Chess	Xiangqi	Mean	Original	Modified	Mean	Remove	Rep/Add	Mean
Gemini-2.5 Pro	17.36	**36.11**	26.74	73.16	26.52	49.81	13.10	28.57	20.83
Sonnet-3.7	7.64	10.42	9.03	42.68	**65.91**	**54.29**	**35.71**	**33.33**	**34.52**
GPT-4.1	11.81	5.56	8.68	**92.17**	5.05	48.61	10.12	27.38	18.75
o3	**56.94**	28.47	42.71	91.67	9.09	50.38	14.88	26.19	20.54
o4-mini	55.56	32.64	**44.10**	90.40	12.12	51.26	12.50	22.62	17.56
Mean	29.86	22.64	26.25	78.02	23.74	50.87	17.26	27.62	22.44

Results On average, VLMs perform poorly (2.12% accuracy) at counting legs of 3-legged and 5-legged counterfactual animals (Tab. 2, Fig. 18). Furthermore, 94.14% of the wrong answers match the original, well-known leg counts (Fig. 4, Fig. 1a, and Tab. 20), demonstrating that VLMs rely mostly on memorized prior knowledge to answer rather than inspecting the legs in the image (see Fig. 3c, and Sec. A.8).

VLMs are slightly worse at counting the legs of birds than counting the legs of mammals (1.01% vs. 2.50%; Tab. 6). Bird legs (Fig. 1a) are typically thinner, which may make it harder to detect than mammals' legs (Fig. 3b). On birds, except for `GPT-4.1`, all VLMs score 0% accuracy (Tab. 6).

A.2 VLMS STRUGGLE TO DETECT LOGO MODIFICATIONS, OFTEN RELYING ON CONTEXT RATHER THAN VISUAL DETAIL

Experiments We replicate the experiment settings from Sec. A.1 on our logo task, evaluating five VLMs on modified shoe and car logo images.

Results VLM performance on car logos (0.44%; Tab. 6) is significantly worse than on shoe logos (17.57%; Tab. 6), as the emblem is small relative to the vehicle (see Fig. 1b). In contrast, shoe logos occupy more image area (see Fig. 3e) and involve only a few simple curves or stripes (i.e., one extra curve for Nike, one added stripe for Adidas). These results highlight two key limitations: VLMs fail to attend to small, context-embedded visual changes and instead rely on memorization, without visually verifying the logo itself (e.g., by zooming in (Taesiri et al., 2023)).

A.3 VLMS FAIL TO COUNT VISUAL ELEMENTS IN MODIFIED FLAGS

Experiments We follow the procedure from Sec. A.1 on our flag tasks. Five VLMs are prompted to count either the number of stars or the number of stripes in original and modified versions of

national flags. Modifications consist of adding or removing a single star or stripe, and each model uses its default settings.

Results VLMs achieve higher mean accuracy on star modifications (11.79%; Tab. 6) than on stripe modifications (4.52%; Tab. 6). This pattern indicates that models are somewhat more attuned to discrete symbol changes (missing or extra stars; see Fig. 6d) than to subtle structural alterations (extra or missing stripes; see Fig. 6b), yet overall sensitivity to flag modifications is extremely limited (9.25%; Tab. 6).

A.4 THINKING MODELS BETTER DETECT CHESS PIECE CHANGES IN MODIFIED CHESS STARTING POSITIONS

Experiments We evaluate five VLMs on a chess-piece counting task using standard starting positions for both Western chess and xiangqi. For each board type, we generate images in which exactly one piece is either removed or replaced by another piece of the same color. All models use their default settings and are prompted to report the total number of pieces or number of a certain piece (e.g., Knights) on the board.

Results VLMs perform significantly better on Western chess (see Fig. 1) than on xiangqi (see Fig. 6a) in terms of mean accuracy (29.86 % vs. 22.64%; Tab. 6). Thinking models (Gemini-2.5 Pro, o3, and o4-mini) all exceed 26% accuracy, whereas non-thinking models (GPT-4.1 and Sonnet-3.7) remain below 10% (Tab. 6). This suggests that on well-structured abstract images, models with explicit reasoning capabilities are better able to detect anomalies.

A.5 VLMS CANNOT COUNT ROWS AND COLUMNS IN SIMPLE GAME BOARDS

Experiments Following our previous tasks, we evaluate five VLMs on counting tasks in four grid-based game boards: chess (8×8), Go (19×19), Sudoku (9×9), and xiangqi (10×9). For chess (see Fig. 6e) and Sudoku (see Fig. 6c), models are asked to report the number of rows and columns. For Go and xiangqi (see Fig. 3f), they report the counts of horizontal and vertical lines.

Table 7: All VLMs' performance is extremely low (2.26%) across game boards, confirming that current VLMs are largely unable to perform simple counting operations in structured visual settings

Model	Chess	Go	Sudoku	Xiangqi	Mean
Gemini-2.5 Pro	2.08	0.00	0.00	6.25	2.38
Sonnet-3.7	0.00	0.00	0.00	6.25	1.79
GPT-4.1	0.00	0.00	0.00	0.00	0.00
o3	0.00	0.00	0.00	**8.33**	2.38
o4-mini	**16.67**	0.00	0.00	0.00	**4.76**
Mean	3.75	0.00	0.00	4.17	2.26

Results All VLMs perform extremely poorly on , (2.26% mean accuracy; Tab. 7). The models even failed to answer any counting questions correctly on Sudoku (see Fig. 6c) and Go (0%; Tab. 7). These findings confirm that current VLMs are unable to execute basic visual counting tasks in structured settings and instead default to overconfident but incorrect guesses.

A.6 VLMS ARE BIASED TOWARDS THE KNOWN ILLUSIONS AND FAIL TO RECOGNIZE THE CHANGES IN THE COUNTERFACTUAL, MODIFIED VERSIONS

Experiment We test five VLMs on 6 classic optical illusions, i.e., Müller-Lyer, Zöllner, Ebbinghaus, Vertical-Horizontal, Pogendorff, and Ponzo (Figs. 30 and 31). Each illusion is presented in two versions: (a) its original form and (b) a counterfactual, modified version where the groundtruth answer is reversed (Fig. 8). For both versions per illusion, we ask VLMs the same Y/N question (see Sec. J).

Results On average, over original and CF versions, all 5 VLMs perform around the random chance (mean accuracy of 50.87%; Tab. 6). 78.02% of the time, VLMs provide answers that are well-known (corresponding to the prior knowledge) but *false* given our CF images (23.74% accuracy).

4 out of 5 VLMs perform well on the original versions of the illusions but poorly on the CF versions, exhibiting **a strong bias to the well-known answers**. Notably, Sonnet-3.7 performs only slightly above the random chance (54.29% accuracy). However, it behaves differently from 4 other VLMs, performing much better on the CF versions than on the original illusions (65.91% vs. 42.68% accuracy; Tab. 6). In sum, our results support the findings that VLMs have a poor, low-level vision capability (Rahmanzadehgervi et al., 2024) and that they are *over*confident.

A.7 VLMs are biased towards the global pattern in a grid

Experiments We test VLMs on counting the shapes or tally marks inside an anomaly cell where the total number of shapes or marks do not follow the patterns in the surrounding cells (Fig. 1g).

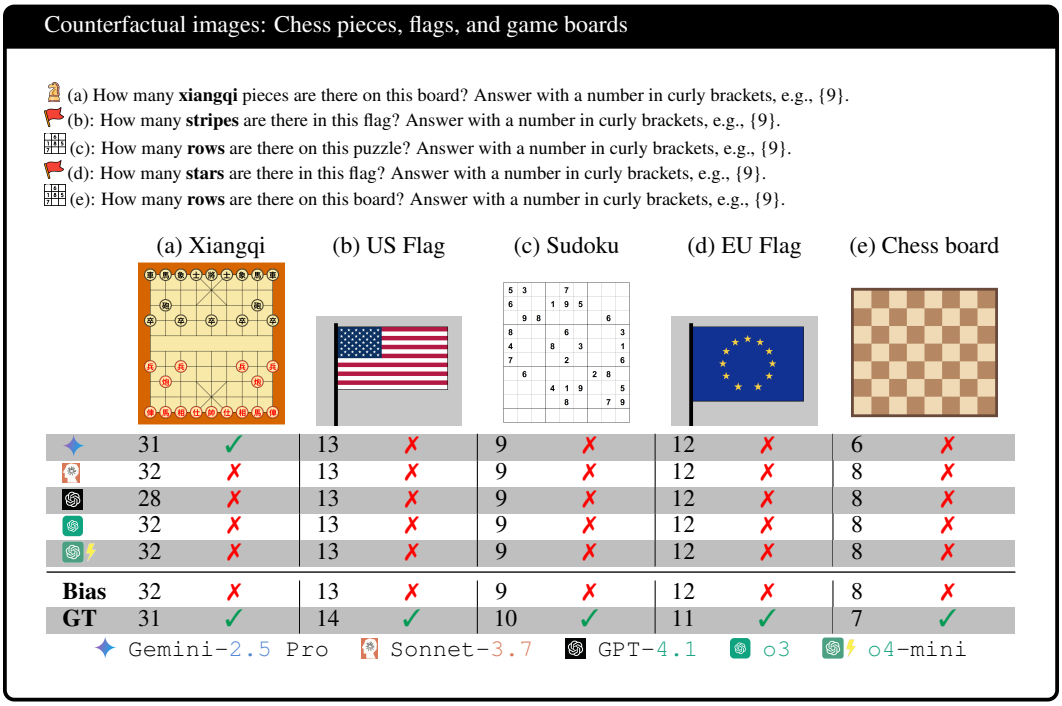


Figure 6: VLMs perform poorly at **counting** elements on counterfactual images across, and domains, often defaulting to the biased answers.

Results Overall, VLMs perform poorly at 22.44% accuracy. 43.45% of all count predictions, both correct and incorrect, match the biased answers (Fig. 4) that correspond to the surrounding cells. In other words, when VLMs make a *wrong* counting predictions, more than half (i.e., 56.02%) of the time, their answers match the **global pattern of most cells in the grid** rather than the target anomaly cell in question (Fig. 33). Our results confirm a striking influence of the background pattern on VLMs' assessment on a small local region. Here, our patterns in the grids are created from scratch and, therefore, do not represent a pattern memorized from the Internet.

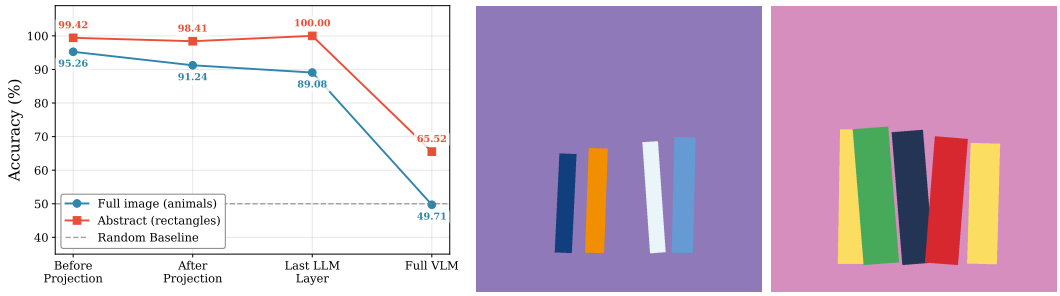
A.8 Linear probing: The vision encoders of VLMs actually extract sufficient leg count information from animal images

Sec. 4.1 demonstrates that VLMs exhibit visual bias, defaulting to memorized answers 75.70% of the time across all models. Here, we investigate whether this failure stems from vision encoders' inability to detect fine-grained modifications or from language models overriding visual evidence with prior knowledge. This experiment is crucial for understanding the source of VLM biases.

Experiments We conduct linear probing experiments using features from the vision encoder (SigLIP 400M (Zhai et al., 2023)) and the language model (Qwen2 0.5B (Yang et al., 2024)) of LLaVA-OneVision-S (Li et al., 2025) on the animal leg counting task. Following Rah-

Table 8: Vision encoder features contain sufficient information to distinguish 4-leg from 5-leg animals (95.26% accuracy before projection), but the complete VLM fails dramatically (49.71%), defaulting to biased answers 99.43% of the time. On abstract images, both linear probing (99.42%) and VLM (65.52%) perform substantially better.

	Animals (5-leg vs 4-leg)		**Rectangles (5 vs 4)**
	Full image	Background removal	Abstract
	Accuracy (%) ↑		
Linear probing (before projection)	**95.26**	**95.98**	99.42
Linear probing (after projection)	91.24	93.39	98.41
Linear probing (last LLM layer)	89.08	95.40	**100.00**
`LLaVA-OneVision-S` (full VLM)	49.71	41.95	65.52
Random baseline	50.00	50.00	50.00
	Bias rate (%) ↓		
`LLaVA-OneVision-S` (full VLM)	99.43	78.30	—



(a) Linear probe across processing stages (b) 4-rectangle abstract image (c) 5-rectangle abstract image

Figure 7: Accuracy degradation across VLM processing stages reveals where bias emerges (a). Vision encoder features maintain high accuracy for both animals (95.26%) and abstract rectangles (99.42%) before projection. As information flows through the LLM, animal counting accuracy collapses to 49.71% while abstract counting degrades less severely to 65.52%, demonstrating that prior knowledge in language models selectively overrides visual evidence. (b–c) Examples of abstract images.

manzadehgervi et al. (2024), we extract features from three processing stages: (1) before projection (vision encoder output, average-pooled to 1×1152 dimensions), (2) after projection, and (3) the last LLM layer (both average-pooled to 1×896 dimensions). We train a logistic regression classifier on these frozen features to distinguish 4-legged from 5-legged an

To do this, we create a dataset of 6,594 mammal images (5,598/300/696 train/val/test split) using the same `Gemini-2.0 Flash`-based generation procedure (Sec. E). We restrict this to mammals only, as they have more diverse species appearances, allowing us to scale up our datasets. We evaluate under two conditions: (1) full image: full images with backgrounds and (2) background removal: cropped images showing only the lower half containing legs (similar to Sec. 4.4). To isolate the effect of memorized knowledge, we also generate an abstract dataset of rectangles (4 vs. 5 rectangles; Figs. 7b and 7c) with the same size data split.

Results The `SigLIP` vision encoder successfully distinguishes 4-legged from 5-legged animals and 4-rectangles from 5-rectangles (95.26%; Tab. 8). In contrast, `LLaVA-OneVision-S`, which uses the same `SigLIP` encoder paired with `Qwen2-0.5B` LLM performing at random chance (49.71%; Tab. 8). Most striking, it outputs "4 legs" for 99.43% of all images (i.e., bias rate) of all 5-legged animal images. Removing backgrounds by cropping to legs maintains high linear classifier accuracy (95.26% → 95.98%) while reducing the VLM's bias rate (99.43% → 78.30%), though performance of `LLaVA-OneVision-S` remains poor (41.95%; Tab. 8).

On abstract rectangles with no counterfactuals, linear probing achieves almost perfect accuracy before projection (99.42%), and LLaVA-OneVision-S performs substantially better compared to itself on animals (66.52% vs. 49.71%). Across processing stages, linear probing accuracy degrades slightly on animals (95.26% → 91.24% → 89.08%; see Fig. 7a) but remains near perfect on abstract images (99.42% → 98.41% → 100.00%; see Fig. 7a). This suggests that the language model increasingly biases representations toward memorized answers. These results confirm that **vision encoders successfully detect visual modifications, but language models override this evidence with memorized knowledge**.

A.9 VLMs ARE EVEN MORE BIASED WHEN THE SUBJECT NAME IS INSERTED INTO THE IMAGE

Prior sections have shown that VLMs perform poorly on the objective task of counting when the background contains **visual** cues that strongly correlate with well-known subjects. As VLM outputs may be influenced by adversarial or distracting text in the image (Goh et al., 2021), here, we test how in-image **textual** cues about the subjects (e.g., "Ebbinghaus illusion") influence VLMs on the same counting questions.

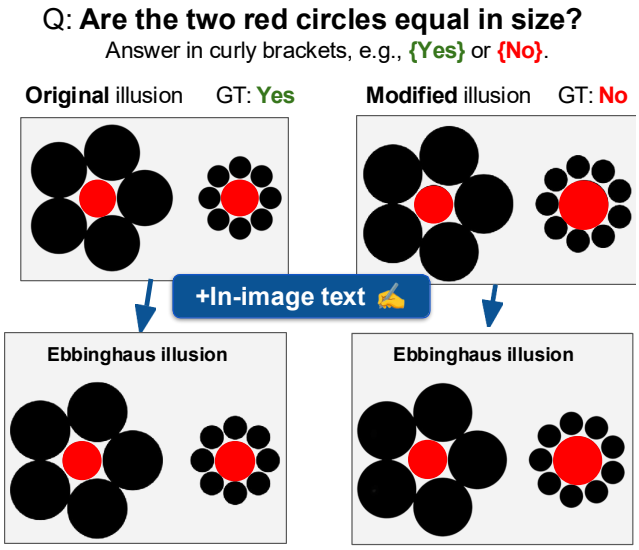


Figure 8: Original vs. modified versions without (top) and with (bottom) the in-image text ("Ebbinghaus illusion").

Experiments We insert the subject name (e.g., "Adidas" or "Ebbinghaus illusion"; Fig. 8) into the top of all original and CF images, extending the image vertically but keeping the original content unchanged. We repeat previous experiments asking VLMs the two counting questions (Q1 & Q2).

Results All VLMs perform worse when an in-image text is added (-4.49; Tab. 9). Interestingly, the decrease is more pronounced for thinking models (Tab. 9), such as o4-mini (-6.56), o3 (-6.41), than for non-thinking ones such as Sonnet-3.7 (-2.81) and GPT-4.1 (-2.67). This result is consistent with recent findings that thinking models tend to hallucinate more (OpenAI, 2025b; Zhang et al., 2024b), here more biased toward the text in the image despite contradictory visuals.

A.10 HELPFUL PROMPTS DO NOT AMELIORATE THE BIAS ISSUES IN VLMs

Previous results show that VLMs rely heavily on prior knowledge to answer objective counting questions. Here, we test how incorporating *helpful* instructions in the prompts may help VLMs become less biased.

Experiments We apply two prompting strategies across all VLMBias tasks:

Table 9: Adding adversarial, in-image textual cues that state the subject name (e.g., "Adidas") cause VLMs to decrease their accuracy (-4.49) on counterfactual images (b). In contrast, instructing VLMs to rely exclusively on the image details to answer questions (Debiased) or to double-check its answers (Double-Check) only slightly improves accuracy, by +1.87 and +2.70, respectively (c).

Model	a. Baseline	b. Adversarial	c. Helpful textual prompt	
		w/ In-image text	w/ Debiased Prompt	w/ Double-Check
Gemini-2.5 Pro	16.02	12.04 (-3.98)	19.72 (+3.70)	20.22 (+4.20)
Sonnet-3.7	16.59	13.78 (-2.81)	19.29 (+2.70)	20.86 (+4.27)
GPT-4.1	13.88	11.21 (-2.67)	14.38 (+0.50)	16.00 (+2.12)
o3	18.50	12.09 (-6.41)	18.94 (+0.44)	21.02 (+2.52)
o4-mini	20.25	13.69 (-6.56)	22.25 (+2.00)	20.61 (+0.36)
Mean	17.05	12.56 (-4.49)	18.92 (+1.87)	19.75 (+2.70)

(1) Debiased Prompt: We prepend the original question (Q1 and Q2) with "*Do not assume from prior knowledge and answer only based on what is visible in the image.*" to encourage models to rely exclusively on image contents.

(2) Double-Check: After VLMs answer the original question, we add a follow-up prompt of "*Please double-check your answer and give your final answer in curly brackets, following the format above.*"

These prompts are designed to encourage VLMs to examine the image more carefully. All experiments use the same images and default model settings as in the baseline setup.

Results Both helpful prompting strategies improve VLM accuracy but only slightly over the baseline, +1.87 for Debiased and +2.70 for Double-Check (Tab. 9c). That is, explicitly instructing models to rely on image contents or verify their answer helps to some extent but does not address the core issue of bias (Sec. M.1).

A.11 Re-running experiments multiple times yields consistent results

To ensure the robustness of our findings and provide richer evaluation metrics, we investigate whether VLM performance varies significantly across multiple runs and examine other aspects beyond accuracy and bias rate.

Experiments We conduct 5-run experiments on our top-performing thinking and non-thinking VLMs (i.e., o4-mini and Sonnet-3.7) across all VLMBias tasks. For each run, we measure:

1. Mean accuracy across 5 runs: average percentage of correct answers when the model is evaluated 5 times on the same dataset
2. Pass@5 rate: the frequency that at least 1 of 5 outputs is correct
3. Bias rate: probability of biased answers across runs
4. Agreement-based consistency: probability of the most frequent answer
5. Model self-reported confidence scores: we ask VLMs in a second turn to provide confidence scores for their first-turn answers.

Table 10: VLMs demonstrate consistently poor performance (17.89% mean accuracy, 26.55% pass@5 rate) yet exhibit severe overconfidence (91.25% self-reported confidence score), with high agreement-based consistency (92.79%) indicating they reliably produce the same incorrect answers across 5 runs.

Metric	o4-mini	Sonnet-3.7	Mean
Mean accuracy ↑	19.54 ± 0.68	16.23 ± 0.36	17.89
Pass@5 rate ↑	30.16	22.93	26.55
Bias rate ↓	73.66	77.27	75.47
Agreement-based consistency ↑	90.82	94.75	92.79
Model self-reported confidence score ↑	84.73	97.77	91.25

Results Mean accuracy scores remain stable across 5 runs (17.89%). Pass@5 rates provide only modest improvement (26.55%), indicating that even with multiple attempts, VLMs cannot effectively solve counterfactual problems in VLMBias. Most striking is the high agreement-based consistency (92.79% model mean), showing VLMs consistently produce identical answers across runs. Despite poor performance, VLMs exhibit severe overconfidence with self-reported confidence scores (91.25% model mean). The bias rate remains consistently high (75.47% model mean), confirming that VLMs persistently default to memorized patterns regardless of multiple attempts for correction.

A.12 Providing in-context examples of animals with abnormal legs fails for o4-mini as it sometimes distrusts the provided labels

Few-shot prompting typically improves VLM performance by providing in-context learning examples that help models adapt to specific tasks. Here, we investigate whether visual demonstrations of counterfactual animals can help VLMs overcome their systematic biases and improve counting accuracy.

Table 11: o4-mini (thinking model) exhibit strong resistance to few-shot examples and distrusts visual evidence (+1.66–+8.12), while GPT-4.1 (non-thinking model) responds effectively to few-shot prompting (+15.75–+51.29).

Configuration	Accuracy (%)		Bias rate (%)	
	o4-mini	GPT-4.1	o4-mini	GPT-4.1
Zero-shot	0.18	9.52	97.25	79.67
Few-shot	1.84 (+1.66)	25.27 (+15.75)	80.51 (-16.74)	70.70 (-8.97)
Few-shot + strong labels	2.57 (+2.39)	23.81 (+14.29)	77.94 (-19.31)	72.53 (-7.14)
Few-shot + strong labels + hint	**8.30** (+8.12)	**60.81** (+51.29)	**13.04** (-84.21)	**30.40** (-49.27)

Experiments We design three few-shot prompting strategies to test on the animal counting task:

1. **Few-shot**: Provide one normal (4-legged) and one counterfactual (5-legged) example, each clearly labeled as "This is an x-legged animal." This establishes the basic task format and demonstrates that animals can have non-standard leg counts.
2. **Few-shot + strong labels**: Use the same visual examples but reinforce with stronger verification language: "This is an x-legged animal, which has been verified." This approach tests whether stronger language can override model biases.
3. **Few-shot + strong labels + hint**: Build upon the previous strategy by adding an explicit warning to the test question: "HINT: This is an animal with an unusual number of legs." This directly alerts the model to expect counterfactual cases.

To ensure robust evaluation, we randomize the order of few-shot examples across questions and vary the animal species used in demonstrations (e.g., dogs, cats, lions). We evaluate these strategies on two models with different capabilities: o4-mini (i.e., thinking) and GPT-4.1 (i.e., non-thinking).

Results o4-mini demonstrates strong resistance to few-shot examples, showing only minimal improvement (+1.66; Tab. 11) over zero-shot performance. Qualitative analysis reveals active distrust of provided labels (Fig. 42), persisting even with strong verification language (Figs. 43 and 44). This distrust causes the model to rely on knowledge priors rather than visual and few-shot evidence. Even with explicit hints, o4-mini reluctantly acknowledges counterfactual cases but continues miscounting (Fig. 45). While this significantly reduces bias-aligned errors (-84.21; Tab. 11), accuracy improvement remains modest (+8.12; Tab. 11) compared to zero-shot.

In contrast, GPT-4.1 responds effectively to few-shot prompting (+14.29–+51.29; Tab. 11). This finding aligns with recent observations that thinking models exhibit increased hallucination rates (OpenAI, 2025b), here manifesting as rejection of accurate visual information.

A.13 Thinking VLMs show limited improved accuracy

Recently, newer thinking VLMs have been released, which need to be evaluated on our benchmark to provide a complete view of current models' capabilities.

Table 12: Full results across proprietary SOTA VLMs (Sec. 4.2), open-source VLMs (Sec. A.14), pointing VLMs (Sec. A.16) and tool-using VLMs (Sec. A.15). Latest thinking VLMs show mixed results on canonical answer bias: GPT-5 achieves modest improvement (30.72%) while Grok-4 underperforms older VLMs (17.71% vs. o4-mini's 20.25%).

Model	Accuracy (%) in counting questions (Q1 &Q2) on counterfactual images								Bias rate (%)
	a.	b.	c.	d.	e.	f.	g.	Task mean	Task mean
Proprietary SOTA VLMs (Sec. 4.2)									
Gemini-2.5 Pro	0.00	1.96	10.42	26.74	2.38	49.81	20.83	16.02	76.79
Sonnet-3.7	0.00	2.72	13.75	9.03	1.79	**54.29**	**34.52**	16.59	76.63
GPT-4.1	**9.52**	9.07	2.50	8.68	0.00	48.61	18.75	13.88	76.62
o3	0.92	7.60	5.00	42.71	2.38	50.38	20.54	18.50	74.81
o4-mini	0.18	**9.31**	**14.58**	**44.10**	**4.76**	51.26	17.56	**20.25**	73.66
Grok-4	2.56	7.84	9.58	34.72	8.93	51.39	8.93	17.71	54.32
GPT-5	4.76	**14.95**	**25.83**	**84.72**	**18.15**	48.48	18.15	**30.72**	57.36
Mean	2.56	7.64	11.67	35.81	5.48	50.60	19.90	19.10	70.03
Open-source VLMs (Sec. A.14)									
Pixtral-12B	0.00	1.47	18.52	1.02	**10.13**	50.94	2.99	12.15	58.96
Pixtral-Large-2411	0.00	8.09	7.66	1.39	7.83	51.77	18.45	13.60	72.31
Qwen2.5-VL-7B	**0.18**	**13.48**	**23.75**	0.70	9.58	**55.19**	13.43	**16.62**	**52.56**
Qwen2.5-VL-72B	0.00	7.84	11.25	**1.74**	2.98	53.03	**20.24**	13.87	67.94
Mean	0.05	7.72	15.29	1.21	7.63	52.73	13.78	14.06	62.94
Pointing VLMs (Sec. A.16)									
Moondream-2B	**74.36**	16.91	55.00	35.07	1.79	49.75	0.00	33.27	46.78
Molmo-7B-D	45.79	**19.57**	**59.58**	24.31	**60.71**	**54.29**	4.46	**38.39**	32.80
Molmo-72B	48.90	9.18	36.25	**36.81**	53.57	56.06	13.99	36.39	**23.92**
Mean	56.35	15.22	50.28	32.06	38.69	53.37	6.15	36.02	34.50
Tool-using VLMs (Sec. A.15)									
o4-mini (chat w/ tools)	3.30	15.63	21.57	51.04	14.06	52.08	17.86	25.08	68.92

Experiments We replicate the previous experiments on Q1 and Q2 on our 7 tasks of VLMBias on the latest notable VLMs: GPT-5 (OpenAI, 2025a), Grok-4 (xAI, 2025).

Results Grok-4 does not surpass o4-mini and o3 (17.71% vs. 20.25% and 18.50%, Tab. 12). Meanwhile, GPT-5 outperforms o4-mini and o3 (30.72% vs. 20.25% and 18.50%, Tab. 12), particularly excelling on the chess pieces (84.72%). However, GPT-5 still falls far short of expectations, and these latest results do not change our conclusions that VLMs remain biased toward canonical answers on our VLMBias.

A.14 Larger open-source VLMs are more biased

The prevailing assumption in the field is that larger models with more parameters should perform better on visual reasoning tasks due to increased representational capacity. However, it remains unclear whether this scaling benefit holds for tasks requiring models to override strong prior knowledge, as larger models may suffer from inverse scaling (McKenzie et al., 2023) having memorized more biased associations from training data.

Experiments We evaluate four open-source VLMs of varying sizes on all VLMBias tasks: Pixtral-12B (Agrawal et al., 2024), Pixtral-Large-2411 (MistralAI, 2024), Qwen2.5-VL-7B, and Qwen2.5-VL-72B (Bai et al., 2025) (Tab. 28). We use the same experimental setup as previous sections, asking counting questions (Q1 and Q2) on counterfactual images across all 7 domains and measuring both accuracy and bias rates.

Results Larger models do not consistently outperform smaller variants and often exhibit increased bias. The mean accuracy across all open-source VLMs is remarkably low (14.06%; Tab. 13), with the smallest model (Qwen2.5-VL-7B) achieving the highest accuracy (16.62%), which is comparable to SOTA closed-source models (17.05% mean accuracy). More concerningly, larger VLMs demonstrate substantially higher bias rates (72.31% for Pixtral-Large-2411 vs. 58.96% for Pixtral-12B; 67.94% for Qwen2.5-VL-72B vs. 52.56% for Qwen2.5-VL-7B; Tab. 13). This pattern suggests that increased model size may actually reinforce memorized biased

associations rather than improve visual reasoning capabilities. Moreover, since open-source VLMs are much smaller than closed-source ones, they contain less knowledge and consequently show lower bias rates compared to closed-source models (62.94% vs. 75.70%). These findings support the hypothesis that *more knowledge leads to more bias* in counterfactual scenarios.

Table 13: Larger open-source VLMs do not outperform smaller variants and exhibit higher bias rates. The smallest VLM (`Qwen2.5-VL-7B` with 7B parameters) achieves the highest accuracy (16.62%) while larger VLMs show substantially increased bias rates (72.31% for `Pixtral-Large-2411` vs. 58.96% for `Pixtral-12B`), supporting the hypothesis that more knowledge leads to more bias.

Model	Accuracy ↑ in counting questions (Q1 &Q2) on counterfactual images								Bias rate ↓
	a.	b.	c.	d.	e.	f.	g.	Task mean	Task mean
`Pixtral-12B`	0.00	1.47	18.52	1.02	**10.13**	50.94	2.99	12.15	58.96
`Pixtral-Large-2411`	0.00	8.09	7.66	1.39	7.83	51.77	18.45	13.60	72.31
`Qwen2.5-VL-7B`	**0.18**	**13.48**	**23.75**	0.70	9.58	**55.19**	13.43	**16.62**	**52.56**
`Qwen2.5-VL-72B`	0.00	7.84	11.25	**1.74**	2.98	53.03	**20.24**	13.87	67.94
Mean	0.05	7.72	15.29	1.21	7.63	52.73	13.78	14.06	62.94

A.15 `o4-mini` USES TOOLS TO ANALYZE IMAGES ONLY ∼30% OF THE TIME AND MOSTLY OUTPUTS DIRECTLY BIASED ANSWERS

Previous experiments evaluate VLMs through API access without tool capabilities. By leveraging tools such as zooming and localization, VLMs can potentially improve their counting accuracy by examining visual details more carefully. However, it remains unclear whether VLMs recognize when visual reasoning is needed for familiar subjects familiar subjects with strong bias cues.

Experiments We compare `o4-mini` in two configurations: (1) standard API access without tools, and (2) ChatGPT interface (OpenAI, 2025) with full Python tool access (e.g., zoom, crop images). We evaluate both versions on counting questions (Q1 and Q2) in `VLMBias` tasks. For the ChatGPT interface, we access it via Puppeteer and measure tool usage frequency through the `tool` tag in the JSON provided by OpenAI's Data Export, and record thinking time to assess computational effort. Due to the rate limit of the ChatGPT interface, we evaluate these two configurations on the 1152px resolution subset.

Table 14: `o4-mini` with tool access shows only modest improvements (+1.9 accuracy, -2.11 bias rate) despite having access to python tools (e.g., zooming, cropping). The limited gains suggest that Python tools cannot overcome deep-seated biases as effectively as specialized built-in counting mechanisms (Sec. A.16).

Model	Accuracy in counting questions (Q1 &Q2) on counterfactual images								Bias rate
	a.	b.	c.	d.	e.	f.	g.	Task mean	Overall
`o4-mini` (API w/o tools)	0.0	13.24	18.75	53.12	5.36	50.38	21.43	23.18	**71.03**
`o4-mini` (chat w/ tools)	3.30	15.63	21.57	51.04	14.06	52.08	17.86	**25.08**	68.92
Δ (tools − API)	+3.30	+2.39	+2.82	-2.08	+10.49	+1.7	-3.57	+1.9	-2.11

Results Tool access provides modest improvement, increasing accuracy from 20.25% to 25.08% (+4.83; Tab. 14). Similarly, the bias rate decreases slightly from 72.41% to 68.92% (-3.49; Tab. 14), indicating marginal improvement in avoiding memorized answers. However, despite having access to zooming and localization tools, `o4-mini` employs them in only 29.66% of queries on average (Tab. 16). That is, the model defaults to direct visual assessment 70.34% of the time, suggesting overconfidence in memorized knowledge prevents recognition of when visual reasoning is needed. Importantly, when tools *are* activated, performance improves noticeably: accuracy increases by +26.51 and bias rate decreases by -33.81 on average compared to baseline (Tab. 15). This demonstrates that tools are highly effective when used, but the low activation rate (29.66%) severely limits their overall impact on model performance.

Table 15: Performance when o4-mini **activates tool-using capabilities** (29.66% of queries; Tab. 16). Tool use substantially improves accuracy (+26.51) and reduces bias rate (-33.81) for task mean. Yet, the low tool usage rate driven by overconfidence in memorized knowledge limits overall performance.

Task	Accuracy (%)	Bias rate (%)
a. Animals	8.33	79.17
b. Logos	68.75	18.75
c. Flags	60.0	36.0
d. Chess Pieces	75.0	25.0
e. Game Boards	53.33	13.33
f. Optical Illusions	76.47	23.53
g. Patterned Grid	19.23	50.0
Task mean	51.59 (+26.51)	35.11 (-33.81)

Table 16: o4-mini uses available tools in only 29.66% of queries on average. The low usage rate indicates that overconfidence in memorized knowledge prevents recognition of when visual reasoning is needed.

Task	Avg. time (s)	Tool use (%)
a. Animals	9.89	39.56
b. Logos	5.46	10.87
c. Flags	14.00	38.75
d. Chess Pieces	16.59	37.50
e. Game Boards	10.69	26.79
f. Optical Illusions	2.27	6.82
g. Patterned Grid	16.55	47.32
Task mean	10.78	29.66

A.16 Small VLMs trained explicitly on counting significantly outperform proprietary SOTA VLMs

Fine-tuning VLMs with specialized capabilities may help overcome counting biases. Pointing VLMs (Moondream, 2025; Deitke et al., 2025)—models specifically trained to output coordinate locations—can potentially force visual reasoning rather than relying on memorized patterns. We investigate whether VLMs with explicit pointing abilities can better handle VLMBias compared to larger, general-purpose VLMs without pointing capabilities.

Experiments We test three pointing VLMs: Moondream-2B (Moondream, 2025), Molmo-7B-D, and Molmo-72B (Deitke et al., 2025) (Sec. A.16). For Moondream-2B, we use its counting API, which outputs only coordinate lists, and count the array length as the final answer—this ensures 100% pointing capability usage. The exception is optical illusion task, which requires Y/N responses rather than counting, so we use standard reasoning APIs. For Molmo-7B and Molmo-72B, they autonomously decide whether to invoke pointing capabilities. When used, the coordinate outputs assist in subsequent counting. We evaluate them on the same counting questions (Q1 and Q2) across all VLMBias tasks using identical experimental setups as in the previous sections.

Table 17: Pointing VLMs substantially outperform commercial VLMs across all domains (36.02% vs. 17.05% mean accuracy; Tab. 6) and regular open-source VLMs (36.02% vs. 14.06% mean accuracy; Tab. 13). Even the smallest model (Moondream-2B with 2B parameters) achieves 33.27% accuracy, exceeding most commercial VLMs despite being orders of magnitude smaller.

Model	Accuracy in counting questions (Q1 & Q2) on counterfactual images								Bias rate
	a.	b.	c.	d.	e.	f.	g.	Task mean	Task mean
Moondream-2B	**74.36**	16.91	55.00	35.07	1.79	49.75	0.00	33.27	46.78
Molmo-7B-D	45.79	**19.57**	**59.58**	24.31	**60.71**	**54.29**	4.46	**38.39**	32.80
Molmo-72B	48.90	9.18	36.25	**36.81**	53.57	56.06	13.99	36.39	**23.92**
Mean	56.35	15.22	50.28	32.06	38.69	53.37	6.15	36.02	34.50

Results Pointing VLMs significantly outperform commercial VLMs (36.02% vs. 17.05% mean accuracy; Tabs. 6 and 17) and regular open-source VLMs (36.02% vs. 14.06% mean accuracy; Tabs. 13 and 17). Most remarkably, Moondream-2B with only 2B parameters substantially outperforms o4-mini (33.27% vs. 20.25%; Tab. 17) despite being orders of magnitude smaller. This suggests that training objectives matter more than model scale for overcoming biases in VLMBias. Qualitative results can be found in Sec. M.3.

However, pointing capabilities remain significantly underutilized. Molmo-7B-D and Molmo-72B even achieve better performance (38.39% vs. 36.39%; Tab. 17) but only use

Table 18: Pointing-capable VLMs: frequency of *tool use* ("pointing use") and counting performance. These models are trained to activate the pointing tool only when prompts contain specific trigger patterns; without those triggers they often do not invoke the tool, even when doing so could improve accuracy.

Model	Pointing use (%)			Accuracy (%)		
	Q1	Q2 (Δ)	Avg	Q1	Q2 (Δ)	Avg
Molmo-7B-D	41.59	63.00	52.30	32.16	44.61	38.39
Molmo-72B	36.14	63.36	49.75	26.82	45.97	36.39
Mean	38.87	63.18 (+24.31)	51.03	29.49	45.29 (+15.80)	37.39
Gemini-2.5 Pro	–	–	–	15.59	16.45	16.02
Sonnet-3.7	–	–	–	16.81	16.36	16.59
GPT-4.1	–	–	–	12.55	15.20	13.88
o3	–	–	–	17.33	19.67	18.50
o4-mini	–	–	–	19.96	20.55	20.25
Mean	–	–	–	16.45	17.65 (+1.20)	17.05

Table 19: Ablation study comparing Molmo models' overall performance versus performance when **pointing capabilities are activated**. On both models, pointing improves counting accuracy (+5.90 mean across tasks) while reducing bias rates (-8.86 mean). The most notable performance gain occurs in animals (+41.21 for Molmo-72B, +37.00 for Molmo-7B-D).

Model	a.	b.	c.	d.	e.	g.	Task mean	Bias rate
Molmo-7B-D	45.79	19.57	59.58	24.31	60.71	4.46	38.39	32.80
Molmo-7B-D (w/ pointing)	82.78 (+37.0)	20.45 (+0.88)	56.57 (-3.01)	7.83 (-16.48)	55.36 (-5.35)	4.46 (+0.00)	41.83 (+3.44)	14.36 (-18.44)
Molmo-72B	48.90	9.18	36.25	36.81	53.57	13.99	36.39	23.92
Molmo-72B (w/ pointing)	90.11 (+41.21)	9.46 (+0.28)	44.44 (+8.19)	53.54 (+16.73)	55.06 (+1.49)	14.16 (+0.17)	45.85 (+9.46)	23.01 (-0.91)

pointing 51.03% of the time (Tab. 18). This could be due to overconfidence, defaulting to direct answers without utilizing their pointing capabilities. One interesting finding is that on Q2 (e.g., "Count the legs"), Molmo-7B-D and Molmo-72B use pointing capabilities much more than on Q1 (e.g., "How many legs") (63.18% vs. 38.87%; Tab. 18). This leads to a much higher Δ between Q1 and Q2 for Molmo-7B-D and Molmo-72B compared to commercial VLMs, which show negligible differences (+15.80 vs. +1.20; Tab. 18). This pattern suggests that explicit counting prompts (i.e., Q2) better trigger pointing verification than implicit counting questions (i.e., Q1), though the underutilization indicates that even specialized VLMs struggle to recognize when their memorized knowledge might be misleading.

When pointing is activated, both Molmo models' performance noticeably improves (Tab. 19): Molmo-72B gains +9.46 accuracy with -8.09 bias reduction, while Molmo-7B-D achieves +3.44 accuracy and -18.44 bias reduction. Most notably, on animals, pointing achieves 82.78–90.11% accuracy, demonstrating that localization overcomes memorized priors.

A.17 Same failures across model families rule out image generation bias

A potential concern is that bias could arise from generating and evaluating images with the same model families.

Experiments We analyze the results on animals (generated by Gemini-2.0 Flash) and logos (generated by GPT-4o) from Sec. 4.2 to investigate whether generation bias affects our findings. We examine performance differences between model families on images generated by their own family versus images generated by other families or created programmatically.

Results GPT-family models show no substantial advantage on GPT-4o generated images (bias rates of 88.73% for GPT-4.1 vs. 98.04% for Gemini-2.5 Pro and 96.79% for Sonnet-3.7; Tab. 20). Similar results are shown on Gemini-2.0 Flash generated images (100% bias rate for Gemini-2.5 Pro vs. 97.25% for o4-mini and 100% for Sonnet-3.7; Tab. 20). All VLMs consistently achieve 100% accuracy on unmodified images but fail dramatically on counterfactual versions (17.05% mean accuracy; Tab. 2) regardless of image generation source. **This confirms that the observed bias stems from models' inherent preferences for canonical answers rather than artifacts of the image generation process**.

Table 20: When presented with modified, counterfactual images in VLMBias, VLMs exhibit substantial bias alignment in their counting responses. The mean bias rate of five state-of-the-art VLMs across our seven tasks is 75.70%. o4-mini shows the lowest bias alignment (**73.66%**) indicating relatively better resistance to visual biases. VLMs with thinking capabilities (o4-mini, o3, Gemini-2.5 Pro) demonstrate bias susceptibility similar to non-thinking models (Sonnet-3.7, GPT-4.1).

Model	Bias rate ↓ in counting questions (Q1 &Q2) on counterfactual images							
	a.	b.	c.	d.	e.	f.	g.	Task mean
Gemini-2.5 Pro	100.00	98.04	89.58	70.83	**83.93**	50.19	44.94	76.79
Sonnet-3.7	100.00	96.79	82.50	84.72	97.62	**45.33**	**29.46**	76.63
GPT-4.1	**79.67**	**88.73**	97.08	80.21	98.81	51.39	40.48	76.62
o3	93.77	91.18	93.33	**49.65**	95.24	49.62	50.89	74.81
o4-mini	97.25	90.20	**82.08**	54.17	91.67	48.74	51.49	**73.66**
Mean	94.14	92.99	88.92	67.92	93.45	49.05	43.45	75.70

A.18 Image resolution has minimal impact on VLM performance across VLMBias tasks

Since our VLMBias dataset contains images rendered at multiple resolutions (384px, 768px, 1152px) as part of our generation process, we analyze whether performance varies across these different image sizes to understand if resolution affects bias-driven failures in counting tasks.

Table 21: VLM accuracy (%) across different image resolutions shows minimal variation, with only a 2.85-point mean difference between lowest and highest resolutions.

Model	384px	768px	1152px	Mean	Δ (1152-384)
o4-mini	17.27	20.30	23.18	20.25	+5.91
o3	16.67	17.90	20.94	18.50	+4.27
Sonnet-3.7	14.36	17.79	17.60	16.59	+3.24
GPT-4.1	13.71	13.43	14.49	13.88	+0.78
Gemini-2.5 Pro	15.13	17.76	15.17	16.02	+0.04
Mean	15.43	17.43	18.28	17.05	+2.85

Experiments We break down the accuracy results from our main experiments by the three resolutions present in our dataset: 384px, 768px, and 1152px. Each image was originally generated and tested at these different resolutions, allowing us to examine whether VLM performance on counterfactual counting questions (Q1 & Q2) varies with image size across all 7 domains.

Results Performance remains remarkably consistent across resolutions (15.43% at 384px → 18.28% at 1152px; Tab. 21). These consistent patterns across resolutions reinforce that VLM failures stem from memorized knowledge overriding visual analysis rather than insufficient image detail.

A.19 Humans can count animal legs almost perfectly after 2 seconds of analyzing the image

To establish performance baselines and validate that our counterfactual images are not inherently ambiguous, we investigate human performance on VLMBias under various time constraints. Understanding human capabilities provides crucial context for interpreting VLM failures and confirms whether the visual modifications are perceivable given sufficient examination time.

Experiments We conduct a *anonymous* human study (consent obtained, no PII collected, minimal risk) with 78 participants (mean age 24.4 years, 82.1% with Bachelor's degree or higher, men 51.6%, women 46.2%) who completed the animal leg counting task through our project website. Each participant is randomly assigned to one image viewing time condition (see Fig. 13) throughout the session to answer 10 randomly selected questions (5 original, 5 counterfactual) from our 91-image dataset on animals (Sec. 3.1). We vary image viewing times (Fig. 15) across four conditions:

0.2, 0.5, 1.0, and 2.0 seconds, while allowing unlimited time for reading questions (Fig. 14) and responding (Fig. 16).

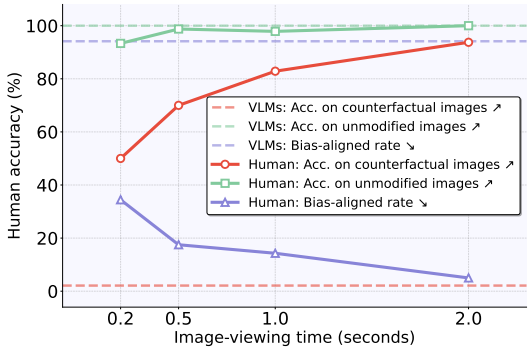


Figure 9: Human accuracy on counterfactual images significantly outperforms VLMs (reaching 93.75% vs 2.12%) on animals with longer image-viewing times, while bias-aligned responses decrease substantially with extended exposure.

Table 22: Human accuracy increases with image-viewing time, reaching 96.88% at 2.0 seconds on animals. Even under severe time pressure (0.2s), humans achieve 50.00% on counterfactual images significantly better than VLMs (mean accuracy of 2.21%) on .

	Image-viewing time (seconds)	Participants (#)	Accuracy (%)		Overall accuracy (%)	Bias rate (%)
			Counterfactual	Original		
Human	0.2	18	50.00	93.33	71.67	34.44
	0.5	16	70.00	98.75	84.38	17.50
	1.0	28	82.86	97.86	90.36	14.29
	2.0	16	**93.75**	**100.00**	**96.88**	**5.00**
SOTA VLMs	–	–	2.12	**100.00**	51.06	94.14

Results Human counting accuracy on animals improves dramatically with increased image-viewing time (71.67% at 0.2 seconds → 96.88% at 2.0 seconds; Tab. 22 and Fig. 9). On counterfactual images specifically, accuracy also rises from 50.00% at 0.2s to 93.75% at 2.0s.

Under extreme time pressure (0.2 seconds), humans exhibit higher bias-aligned responses (34.44%; Tab. 22 and Fig. 9) compared to longer viewing times. But even in this challenging condition, humans still outperform SOTA VLMs (50.00% vs. 2.12% counterfactual accuracy; Tab. 22). This confirms that our counterfactual images are not inherently ambiguous and complex for humans.

A.20 Locate-then-count prompting does not significantly improve counting accuracy

While simple counting prompts prove ineffective (Q1 & Q2; Sec. L.1). the strong performance of pointing VLMs like `Moondream-2B`, `Molmo-7B-D` (36.02% accuracy; Sec. A.16) suggests that forcing explicit localization might be key. We investigate whether more descriptive, step-by-step prompts that first locate, then count (i.e., explicitly instruct VLMs first to locate each element, then count one by one) can help VLMs overcome their bias and improve counting accuracy.

Experiments We replicate the animal leg counting experiment from Sec. 4.2 but modify the prompt to encourage a more procedural approach, using the following enhanced prompt: "*First, locate each leg individually, count them one by one, and then state the final number in curly brackets, e.g., {9}.*" This prompt explicitly guides the model through a localization-then-counting workflow rather than asking for a direct count.

Results Locate-then-count prompting yields only marginal improvements over the original simple prompts Q1 & Q2 (+0.67; Tab. 23), while the bias rate remains high (-1.09; Tab. 23). **These results indicate that explicitly locate-then-count instructions are insufficient to overcome VLMs' strong**

Table 23: Locate-then-count prompting yields only marginal improvements over Q1 & Q2 prompts (+0.67% accuracy, -1.09% bias rate).

	Accuracy		Bias rate	
Model	Q1 & Q2	Locate-then-count prompt (Δ)	Q1 & Q2	Locate-then-count prompt (Δ)
Gemini-2.5 Pro	0.00	0.00 (+0.00)	100.00	96.70 (-3.30)
Sonnet-3.7	0.00	1.83 (+1.83)	100.00	98.17 (-1.83)
GPT-4.1	9.52	10.62 (+1.10)	79.67	82.78 (+3.11)
o3	0.92	1.54 (+0.62)	93.77	93.08 (-0.69)
o4-mini	0.18	0.00 (-0.18)	97.25	94.51 (-2.74)
Mean	2.12	2.80 (+0.67)	94.14	93.05 (-1.09)

visual bias (see Figs. 36 and 37), consistent with findings that prompting-based interventions provide only limited improvements (Sec. A.10). Instead, the correct way to help is by providing tools for VLMs and ensuring that VLMs themselves know when to use them (see Secs. A.15 and A.16).

A.21 Adding subject name to text prompts further decreases VLM accuracy

Our VLMBias uses neutral prompts (e.g., "*Count the legs of this animal.*") to isolate visual bias from prompt bias. However, a key question remains: *does this neutral framing actually matter?* To address this, we test whether injecting object-specific names into our prompts (i.e., non-neutral; similar to Sec. A.9) affects VLM counting performance and bias rates.

Experiments We modify our neutral prompts (Q1 & Q2) from previous experiments by replacing generic descriptors with specific object names. For example, "the left shoe" becomes "the left *Nike* shoe" (™ logos), "this puzzle" becomes "this *Sudoku* puzzle" (game boards). We evaluate Sonnet-3.7 (best non-thinking) and o4-mini (best thinking) using these non-neutral prompts on the same counterfactual images across all 7 tasks.

Table 24: Non-neutral prompts substantially reduce counting **accuracy** (-4.75), with o4-mini experiencing 3× larger degradation than Sonnet-3.7 (-7.09 vs. -2.41 points) across 7 tasks.

Model	a.	b.	c.	d.	e.	f.	g.	Task mean
Sonnet-3.7 (Neutral)	0.00	2.72	13.75	9.03	1.79	54.29	34.52	16.59
Sonnet-3.7 (Non-neutral)	0.00 (+0.00)	1.98 (-0.74)	9.58 (-4.17)	2.43 (-6.60)	1.79 (+0.00)	49.87 (-4.42)	33.63 (-0.89)	14.18 (-2.41)
o4-mini (Neutral)	0.18	9.31	14.58	44.10	4.76	51.26	17.56	20.25
o4-mini (Non-neutral)	0.18 (+0.00)	8.09 (-1.23)	5.42 (-9.17)	15.62 (-28.47)	0.00 (-4.76)	50.00 (-1.26)	12.80 (-4.76)	13.16 (-7.09)
Model Mean (Neutral)	0.09	6.01	14.17	26.56	3.27	52.78	26.04	18.42
Model Mean (Non-neutral)	0.09 (+0.00)	5.03 (-0.98)	7.50 (-6.67)	9.03 (-17.53)	0.89 (-2.38)	49.94 (-2.84)	23.21 (-2.83)	13.67 (-4.75)

Table 25: Non-neutral prompts increase **bias rates** across all tasks (+5.32), demonstrating that object-specific names strongly activate textual priors.

Model	a.	b.	c.	d.	e.	f.	g.	Task mean
Sonnet-3.7 (Neutral)	100.0	96.79	82.5	84.72	97.62	45.33	29.46	76.63
Sonnet-3.7 (Non-neutral)	99.82 (-0.18)	97.77 (+0.98)	88.33 (+5.83)	97.57 (+12.85)	98.21 (+0.60)	47.22 (+1.89)	31.25 (+1.79)	80.03 (+3.40)
o4-mini (Neutral)	97.25	90.20	82.08	54.17	91.67	48.74	51.49	73.66
o4-mini (Non-neutral)	97.25 (+0.00)	88.24 (-1.96)	89.58 (+7.50)	84.38 (+30.21)	98.81 (+7.14)	50.00 (+1.26)	58.04 (+6.55)	80.90 (+7.24)
Model Mean (Neutral)	98.63	93.49	82.29	69.44	94.64	47.03	40.48	75.14
Model Mean (Non-neutral)	98.53 (-0.09)	93.00 (-0.49)	88.96 (+6.67)	90.97 (+21.53)	98.51 (+3.87)	48.61 (+1.58)	44.64 (+4.17)	80.46 (+5.32)

Results Adding object names to prompts significantly degrades performance (-4.75% mean accuracy; Tab. 24) and increases bias rates (+5.32%; Tab. 25) for both Sonnet-3.7 and o4-mini. Notably, the thinking model suffers nearly 3× larger accuracy degradation than the non-thinking when exposed to non-neutral prompts (-7.09 vs. -2.41 points). **These results demonstrate that non-neutral prompts invoke stronger textual priors that override visual information, and even extended reasoning capabilities overcome this bias.** This confirms that neutral prompting is essential for fairly assessing whether VLMs can overcome their language bias when analyzing counterfactual images.

A.22 VLMs FAIL TO DETECT MODIFICATIONS EVEN WITH SIDE-BY-SIDE COMPARISON

Prior sections show that VLMs struggle to count legs correctly in counterfactual images. Here, we test whether providing explicit side-by-side comparisons with original images helps VLMs detect the modifications, as the reference image may make the differences more noticeable.

Experiments We present VLMs with two images simultaneously: the original animal image (with canonical leg count) and its modified counterfactual version (with one extra leg). We prompt models with: *"Compare the two images side by side. Do the animals in image 1 and image 2 have the same number of legs? Return the final Yes/No answer in curly brackets (e.g., {Yes} or {No})."*. Here, we expect the VLMs to always answer {No} if they can distinguish the differences.

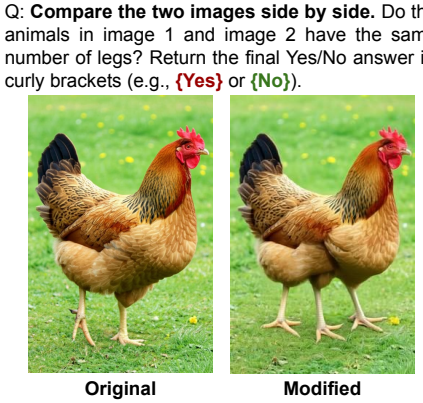


Figure 10: The side-by-side comparison prompt and an example input image pair.

Model	Percentage of {No} (%)
Random baseline	50
Gemini-2.5 Pro	9.89
Sonnet-3.7	9.89
GPT-4.1	10.99
o3	10.99
o4-mini	**15.38**
Mean	11.76

Table 26: VLMs fail to detect leg count differences when comparing original and counterfactual images side-by-side. The ground truth is "No", but models output "No" only 11.76% of the time, far below the 50% expected from random guessing.

Results **VLMs are so biased that even when counterfactual and original images are placed side by side, they still cannot detect the modifications.** The mean percentage of {No} across 5 SOTA VLMs is only 11.76% (Tab. 26). This demonstrates that VLMs' bias toward prior knowledge is so strong that even direct visual comparison fails to surpass the random-guessing (50%) baseline.

A.23 EVEN WHEN ATTENDING TO CORRECT REGIONS, VLMs STILL FAIL TO GENERATE CORRECT ANSWERS

Prior sections demonstrate that VLMs fail at counting counterfactual elements despite vision encoders successfully encoding visual information (Sec. A.8). Here, we investigate whether VLMs attend to the correct visual regions during inference by analyzing attention patterns when generating answers.

Experiments Our preliminary analysis of attention patterns evolution throughout the layers (Fig. 11) reveals that Qwen2.5-VL-7B progressively localizes relevant objects (e.g., legs, logo elements) in later layers. Following this, we compute the final layer's attention mapping of the answer token over the image tokens for Qwen2.5-VL-7B. For example, when the model outputs "{3}" in response to counting a dog's legs, we extract the attention weights of the token "3" across the image tokens of the last layer. We visualize the attention by mapping each image token to its corresponding image patch and overlay the attention heatmap on the original image.

Results Interestingly, even when **the model correctly attend to the regions of interest, it often produces incorrect or biased answers**. For instance, when counting overlapping circles on a modified Audi logo, Qwen2.5-VL-7B attends strongly to all five circles in the final layer yet outputs "4". This finding is consistent with prior work showing disconnects between visual attention and final model outputs (Liu et al., 2025; Zhang et al., 2025). Combined with our linear probing results (Sec. A.8), this provides strong evidence that VLMs can *see* the correct visual information but are highly influenced by memorized knowledge priors during answer generation.

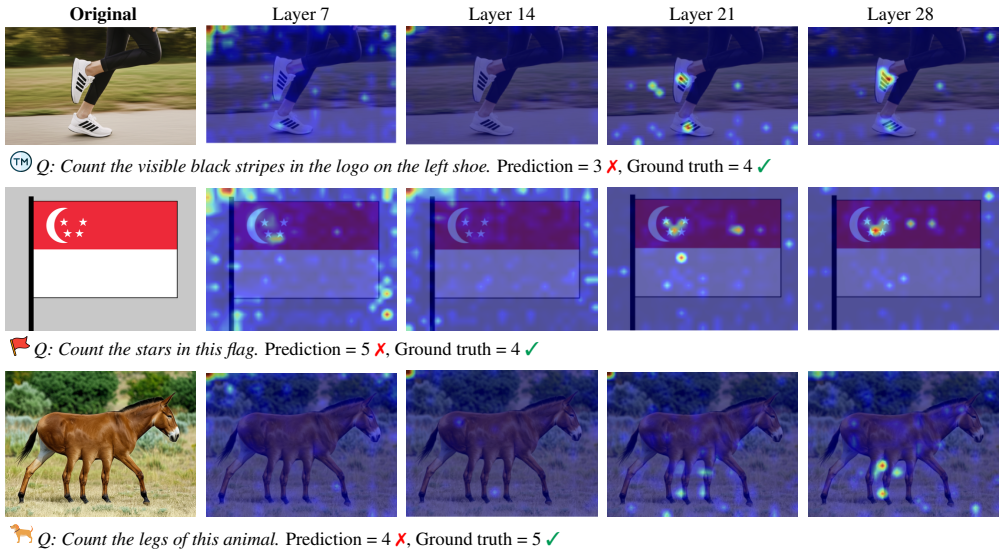


Figure 11: Attention heatmaps across layers for Qwen2.5-VL-7B, revealing that it progressively localizes relevant regions in later layers. **Original**: Input image without attention overlay. **Layers 7-28**: Attention heatmaps overlaid on images, with warmer colors indicating higher attention weights.

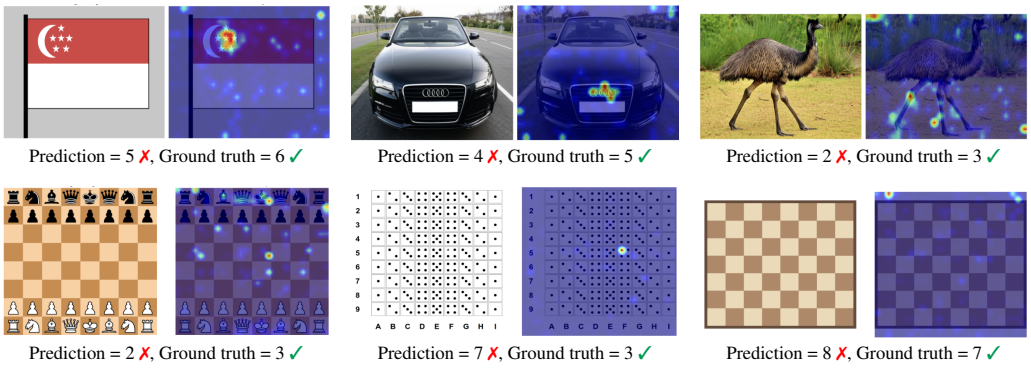


Figure 12: Attention heatmaps from the final layer of the prediction token of Qwen2.5-VL-7B. The model correctly attends to the visual details for flags, car logos, patterned grids, and three-legged animals when generating its answer token. However, it still outputs incorrect or biased answers instead of the correct count.

B HUMAN STUDY DETAILS

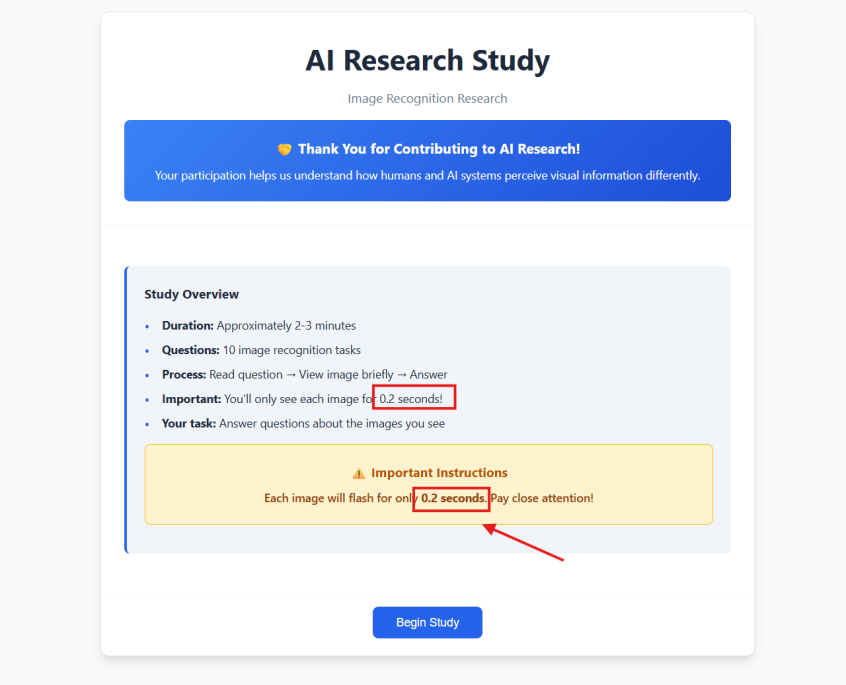


Figure 13: Participants are informed about the task and their randomly assigned image viewing duration (0.2, 0.5, 1, or 2 seconds).

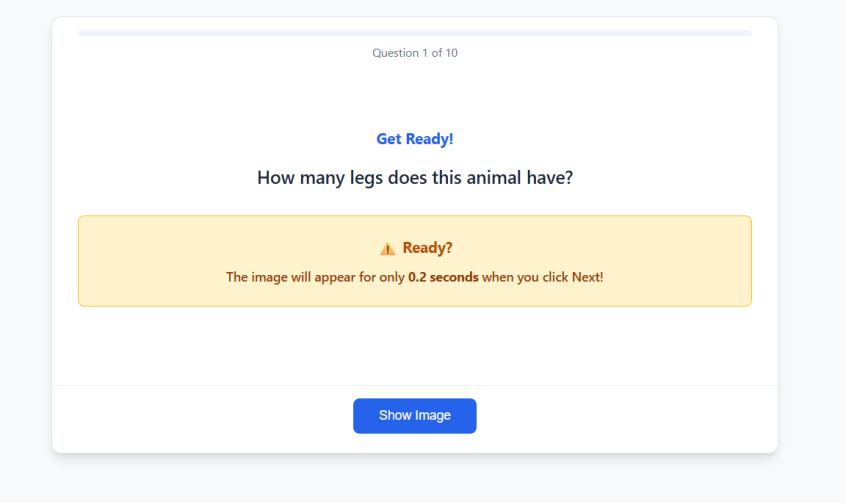


Figure 14: Participants read the question with unlimited time before viewing the image.

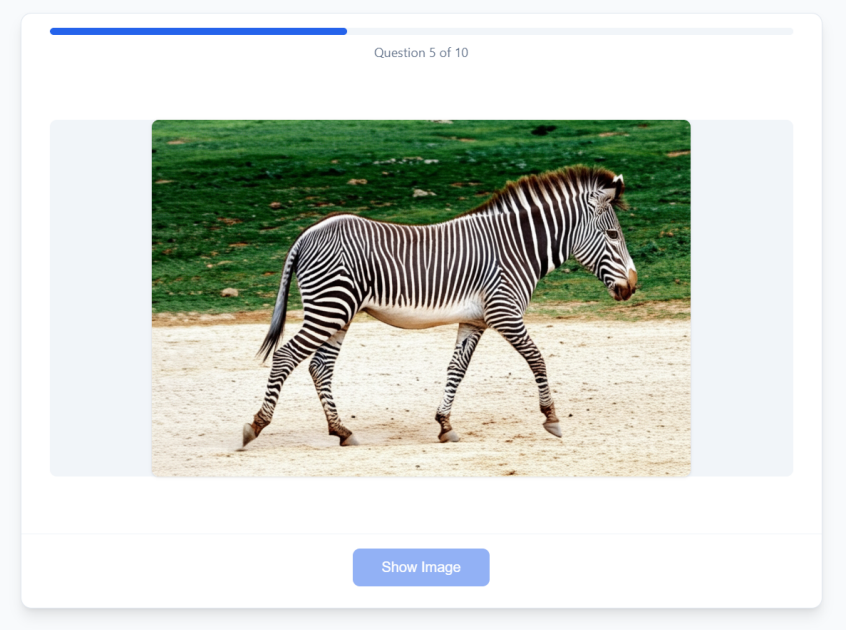


Figure 15: The target image is displayed for the assigned duration (0.2, 0.5, 1, or 2 seconds).

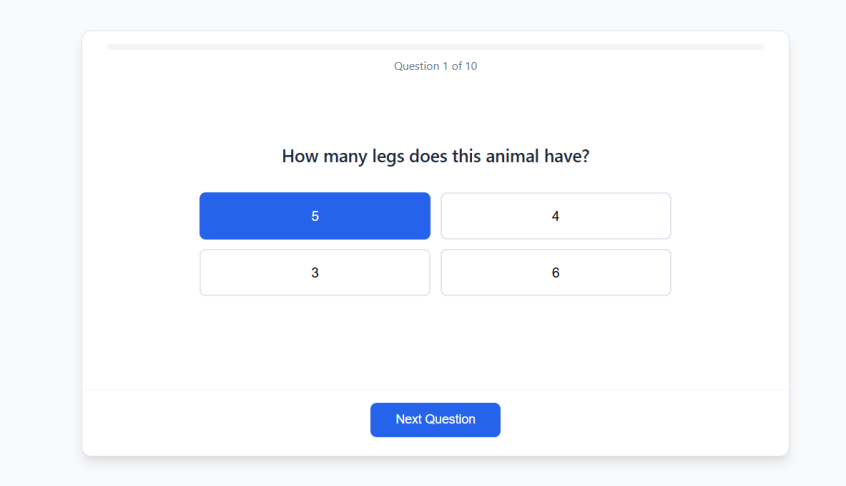


Figure 16: Participants have unlimited time to choose their response from multiple-choice options.

C Detailed comparison with existing VLM bias benchmarks

This section provides additional detailed comparison between VLMBias and related benchmarks (Tab. 1) discussed in Sec. 2, organized by key evaluation criteria.

C.1 Source of bias

VLMBias isolates visual bias through neutral prompts and objective counting, while other benchmarks introduce bias through their question formulations. Specifically, PhD-ccs (Liu et al., 2024), VLind-Bench (Lee et al., 2025), and HallusionBench (Guan et al., 2024b) explicitly mention objects in prompts (e.g., "*Does the car have square wheels?*"), priming models toward knowledge priors. ViLP (Luo et al., 2025) contains two subsets: ViLP^F includes distractor facts that bias responses, while ViLP^P omits distractors. Though ViLP^P is more relevant to our work, it doesn't directly address visual bias like VLMBias. It either uses identification questions (e.g., *"Which animal in the image stores fat in its humps?"*) on modified subjects (horses with humps), which are inherently ambiguous as a horse with humps arguably ceases to be a "horse"; or questions that explicitly mention the object (e.g., *"From the image, in which city is the Red Square located?"*), priming models toward prior knowledge of these named entities (e.g., *the Red Square*). While there are also counting questions in ViLP, they take up only 4% of the questions (12/300), compared to our benchmark which fully focused on counting.

In contrast, VLMBias uses neutral language (e.g., "*How many legs does this animal have?*") with objective counting that results in unambiguous numerical answers. This design ensures that failures indicate memorized knowledge overriding visual evidence, not susceptibility to cues in the questions.

C.2 Benchmark scale

The main dataset of VLMBias provides 1,392 counterfactual images across 7 diverse tasks, exceeding most related benchmarks in scale. Specifically, our main dataset is 1.9 times larger than PhD-ccs (750 images), 2.3 times larger than ViLP (600 images), and 7.7 times larger than HallusionBench (181 images). While VLind-Bench (2,576 images) is larger than our main dataset, VLMBias's full evaluation suite which includes the background removal subset and in-image text injection subset totals 4,176 images, surpassing the scale of VLind-Bench. This scale enables more robust evaluation of VLMs, covering a broad range of scenarios from photo-realistic animals to abstract patterns.

C.3 Image generation method

VLMBias systematically generates photo-realistic, subtly modified versions of familiar subjects using state-of-the-art models, while other benchmarks (1) use older image generators producing surreal-looking images or (2) manually collect images. Specifically, PhD-ccs and VLind-Bench rely on DALL-E, while ViLP uses DALL-E and FLUX to create artificial and surreal scenes. Meanwhile, HallusionBench manually curates counterfactual images, achieving high-quality but lacking scalability. In contrast, VLMBias employs state-of-the-art generators (Gemini-2.0 Flash, GPT-4o) to create subtle modifications of highly familiar subjects (e.g. a 5-legged dog) that looks highly realistic.

D MODELS AND ACCESS DETAILS

Table 27: Model specifications and access details for evaluated commercial VLMs

Model	Model ID	Thinking	Platform	Settings
Gemini-2.5 Pro	gemini-2.5-pro-preview-05-06	✓	Google AI Studio	temperature=1.0
Sonnet-3.7	claude-3-7-sonnet	✗	Anthropic	temperature=1.0
GPT-4.1	gpt-4.1	✗	OpenAI	temperature=1.0
o3	o3	✓	OpenAI	reasoning_effort=medium
o4-mini	o4-mini	✓	OpenAI	reasoning_effort=medium
Grok-4	grok-4	✓	xAI	–
GPT-5	gpt-5	✓	OpenAI	reasoning_effort=medium

Table 28: Model specifications and access details for evaluated open-source VLMs

Model	Model ID	Thinking	Platform	Settings
Pixtral-12B	pixtral-12b	✗	OpenRouter	temperature=1.0
Pixtral-Large-2411	pixtral-large-2411	✗	OpenRouter	temperature=1.0
Qwen2.5-VL-7B	qwen-2.5-vl-7b-instruct	✗	OpenRouter	temperature=1.0
Qwen2.5-VL-72B	qwen2.5-vl-72b-instruct	✗	OpenRouter	temperature=1.0

Table 29: Model specifications and access details for evaluated open-source counting VLMs

Model	Model ID	Text output	Platform	Settings
Molmo-7B-D	allenai/Molmo-7B-D-0924	✓	HuggingFace	temperature=1.0
Molmo-72B	allenai/Molmo-72B-0924	✓	HuggingFace	temperature=1.0
Moondream-2B	vikhyatk/moondream2	✗	HuggingFace	–

E TASK 1: COUNTING LEGS WITH ADDED LIMB

E.1 TASK DESIGN

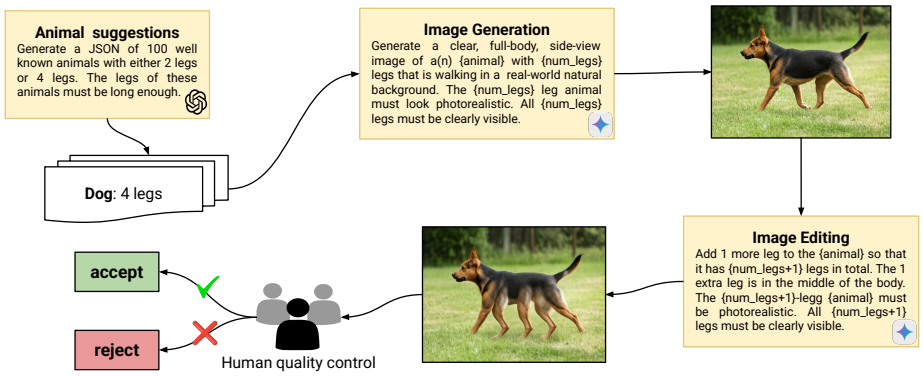


Figure 17: Data generation pipeline for Task 1: Counting legs with added limb.

Pretrained on the Internet data, VLMs must have colossal prior knowledge of the count of animal legs from both textual and image data. Following this hypothesis, we generate images of usual animals with *one additional leg* (e.g., 3-legged birds or 5-legged dogs) and ask VLMs to count legs to evaluate if these models are biased toward their prior knowledge.

- **Animal types**: We modify the legs of **2** types of animals: birds and mammals.
- **Modification types**: Each animal is modified to have **1** additional leg.
- **Target animals**: We select **91** well-known animals, consisting of 23 two-legged birds and 68 four-legged mammals.
- **Image resolutions**: We generate each animal image and rescale them at **3** different pixel sizes {384, 768, 1152}px using the scaling factor in Sec. 3.1 to test resolution sensitivity

This approach generates a total of **91** animals × **1** modification type × **3** resolutions = **273** total images.

E.2 IMPLEMENTATION AND IMAGE GENERATION

Implementation details Our image generation pipeline follows this sequence:

1. Use `o4-mini` to collect a list of well-known animals with clearly visible legs
2. Generate full-body and side-view images of these animals using `Gemini-2.0 Flash`
3. For each animal image, use `Gemini-2.0 Flash` to add one extra leg to the animal. Each animal image is edited over 4 independent trials.
4. Manually inspect and filter out unsatisfactory images
5. Render each approved image at three different resolutions

Quality control We manually inspect the images to ensure that each modified animal image has exactly one additional leg. For cases that fail (e.g., more than one added leg), we remove them from our dataset.

Prompt We use the following prompts to test the VLMs:

- **Q1:** *How many legs does this animal have? Answer with a number in curly brackets, e.g., {9}.*
- **Q2:** *Count the legs of this animal. Answer with a number in curly brackets, e.g., {9}.*

- **Q3:** *Is this an animal with [NumModifiedLegs] legs? Answer in curly brackets, e.g., {Yes} or {No}.*

Ground truth calculation The ground truth answers are as follow:

- **Birds leg counting (Q1&Q2)**:
  - Correct answer: 3 (one additional leg)
  - Expected bias: 2
- **Mammals leg counting (Q1&Q2)**:
  - Correct answer: 5 (one additional leg)
  - Expected bias: 4
- **Animal leg identification question (Q3)**:
  - Correct answer: "No" (always, since each animal has one additional leg)
  - Expected bias: "Yes"

E.3 Qualitative results

How many legs does this animal have? Answer with a number in curly brackets, e.g., {9}.

(a)-(e) How many legs does this animal have? Answer with a number in curly brackets, e.g., {9}.

	(a) Lion		(b) Stork		(c) Elephant		(d) Duck		(e) Dog	
										
Gemini-2.5 Pro	4	✗	2	✗	4	✗	2	✗	4	✗
Sonnet-3.7	4	✗	2	✗	4	✗	2	✗	4	✗
GPT-4.1	5	✓	2	✗	6	✗	2	✗	6	✗
o3	4	✗	2	✗	4	✗	2	✗	6	✗
o4-mini	4	✗	2	✗	4	✗	2	✗	4	✗
Bias	4	✗	2	✗	4	✗	2	✗	4	✗
GT	5	✓	3	✓	5	✓	3	✓	5	✓

Figure 18: VLMs are often biased toward the original number of legs animals have, and they tend to answer based on prior knowledge rather than by analyzing the image.

E.4 List of animals

Mammals: Four-legged animals

horse, zebra, donkey, mule, cow, buffalo, yak, water buffalo, deer, elk, moose, reindeer, caribou, gazelle, giraffe, camel, dromedary camel, bactrian camel, llama, alpaca, goat, ibex, mountain goat, pronghorn, bighorn sheep, wild boar, pig, warthog, coyote, lynx, bobcat, leopard, tiger, lion, jaguar, puma, ocelot, caracal, hyena, rabbit, impala, springbok, kudu, eland, wildebeest, okapi, hippopotamus, african elephant, asian elephant, indian rhinoceros, gnu, maned wolf, arctic fox, red fox, fennec fox, red wolf, domestic dog, domestic cat, african wilddog, dingo, jackal, gray wolf, hare, cheetah, antelope, bison, sheep, serval

Birds: Two-legged animals

ostrich, emu, rhea, cassowary, heron, stork, crane, egret, ibis, spoonbill, turkey, chicken, rooster, duck, swan, peacock, sandpiper, avocet, stilt, plover, lapwing, oystercatcher, secretary bird

F TASK 2: COUNTING ELEMENTS IN MODIFIED BRAND LOGOS ™

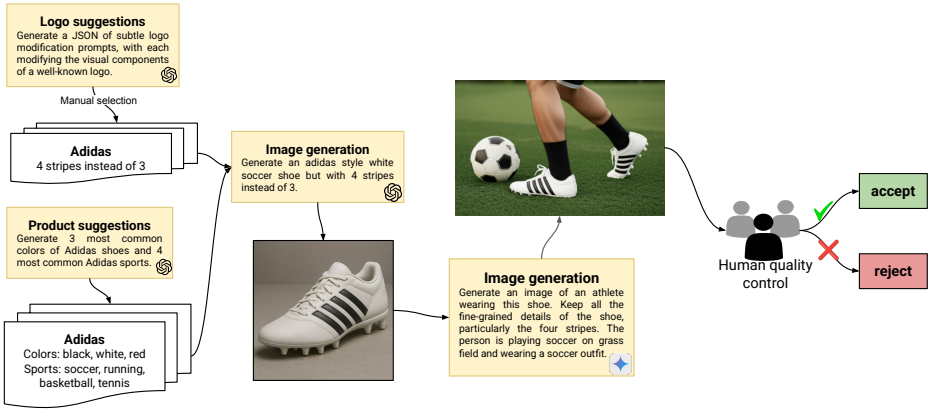


Figure 19: Data generation pipeline of shoe logos for Task 2: Counting elements in modified brand logos

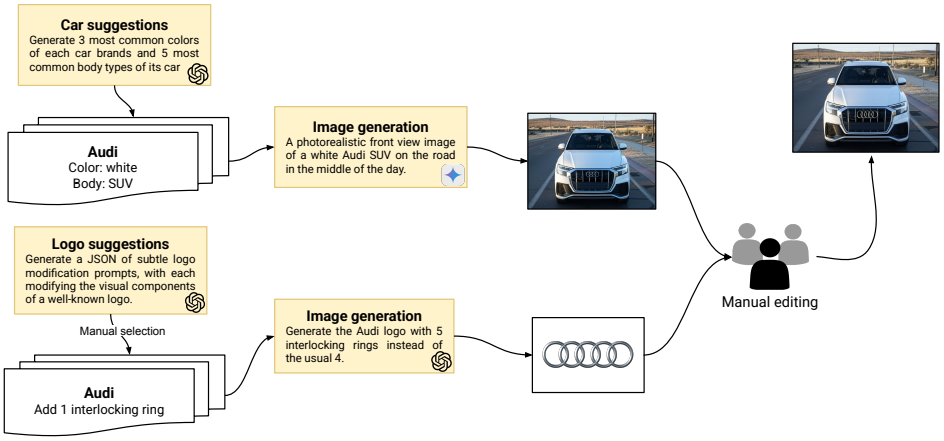


Figure 20: Data generation pipeline of car logos for Task 2: Counting elements in modified brand logos

F.1 TASK DESIGN

Our initial evaluation show that some VLMs, such as o4-mini, can accurately count the four stripes on modified Adidas logo on white background. As such, to increase the task difficulty, we hypothesize that VLMs strongly associate ™ logos with the background they typically appear on. Subsequently, we examine if the visual cues from the background would be strong enough to suppress counting the elements in the logos. Our task is designed as follow:

- **Brand types**: We use 2 different brand types: *cars* and *shoes*
- **Target brands**: We select 5 well-known brands with quantifiable graphical elements:
  - *Car brands*: Mercedes-Benz, Maserati, and Audi (3 brands)
  - *Shoe brands*: Adidas and Nike (2 brands)
- **Background variations**: Each brand logo has specific background settings:
  - *Car logo background*: Car logos always appear on cars. For each logo, we collect 5 car body types × 3 colors (white, grey, black)

  - *Shoe logo background*: Shoe logos are often seen on the footwear of athletes. For each logo, we collect a list of 4 relevant sports (tennis, running, basketball, soccer) × 3 colors (black, red, white)
- **Image resolutions**: We generate each image and rescale them at 3 different pixel sizes {384, 768, 1152}px using the scaling factor in Sec. 3.1 to test resolution sensitivity

This systematic approach generates a total of [3 car brands × (5 × 3) × 3 resolutions] + [2 shoe brands × (4 × 3) × 3 resolutions] = **135** + **72** = **207** total images.

F.2 Implementation and prompts

Implementation details We employ the following process to generate logo modification images:

1. Use o4-mini to suggest graphical modifications for each logo (e.g., increasing Adidas' three stripes to four). We then select the most relevant suggestions for our benchmark.
2. Generate modified logo versions using GPT-4o.
3. Create background images:
   - *Background images for car logos*:
     - Use o4-mini to suggest popular colors and body types of each car logo.
     - For each logo, generate and select relevant images of cars from the logo brand with the determined body types and colors.
     - Manually place modified logos in typical car logo positions.
   - *Background images for shoe logos*:
     - Use o4-mini to suggest popular shoe colors and sports affiliated with each shoe logo.
     - For each logo, generate and select relevant images of athletes wearing shoes with the modified logo for each determined color and sport.
4. Render each image at three different resolutions.

Quality control To ensure high-quality images, we manually review to make sure that: (1) each generated logo has the correct number of modified elements; (2) each product is clearly visible and oriented correctly; and (3) the position of the logos on the products are natural-looking.

Prompts We use the following prompts

1. **Counting questions (Q1 & Q2)**:
   - **Q1 (Adidas):** *How many visible [StripeColor] stripes are there in the logo of the left shoe? Answer with a number in curly brackets, e.g., {9}.*
   - **Q1 (Nike):** *How many visible [CurveColor] stylized curves are there in the logo of the left shoe? Answer with a number in curly brackets, e.g., {9}*
   - **Q1 (Audi):** *How many overlapping circles are there in the logo of this car? Answer with a number in curly brackets, e.g., {9}.*
   - **Q1 (Mercedes):** *How many points are there on the star in the logo of this car? Answer with a number in curly brackets, e.g., {9}.*
   - **Q1 (Maserati):** *How many prongs are there in the logo of this car? Answer with a number in curly brackets, e.g., {9}*
   - **Q2 (Adidas):** *Count the visible [StripeColor] stripes in the logo of the left shoe. Answer with a number in curly brackets, e.g., {9}.*
   - **Q2 (Nike):** *Count the visible [CurveColor] stylized curves in the logo of the left shoe. Answer with a number in curly brackets, e.g., {9}*
   - **Q2 (Audi):** *Count the overlapping circles in the logo of this car. Answer with a number in curly brackets, e.g., {9}.*
   - **Q2 (Mercedes):** *Count the points on the star in the logo of this car. Answer with a number in curly brackets, e.g., {9}.*
   - **Q2 (Maserati):** *Count the prongs in the logo of this car. Answer with a number in curly brackets, e.g., {9}*

2. **Y/N identification questions (Q3)**:
    - **Q3 (Adidas):** *Are the logos on these shoes Adidas logos? Answer in curly brackets, e.g., {Yes} or {No}.*
    - **Q3 (Nike):** *Are the logos on these shoes Nike logos? Answer in curly brackets, e.g., {Yes} or {No}.*
    - **Q3 (Audi):** *Is the logo on this car Audi logo? Answer in curly brackets, e.g., {Yes} or {No}.*
    - **Q3 (Mercedes):** *Is the logo on this car Mercedes-Benz logo? Answer in curly brackets, e.g., {Yes} or {No}.*
    - **Q3 (Maserati):** *Is the logo on this car Maserati logo? Answer in curly brackets, e.g., {Yes} or {No}.*

Ground truth calculation The ground truth answers are as follow:

- **Adidas stripes counting (Q1&Q2)**:
    - Correct answer: 4
    - Expected bias: 3
- **Nike stylized curves counting (Q1&Q2)**:
    - Correct answer: 2
    - Expected bias: 1
- **Audi overlapping circles counting (Q1&Q2)**:
    - Correct answer: 5
    - Expected bias: 4
- **Mercedes-Benz points on the star counting (Q1&Q2)**:
    - Correct answer: 4
    - Expected bias: 3
- **Maserati prongs counting (Q1&Q2)**:
    - Correct answer: 5
    - Expected bias: 3
- **Logo identification question (Q3)**:
    - Correct answer: "No" (all logos are modified)
    - Expected bias: "Yes"

F.3 Qualitative results

Car logos

™ **(a), (d):** How many overlapping circles are there in the logo of this car? Answer with a number in curly brackets, e.g., {9}.
™ **(b), (e):** How many points are there on the star in the logo of this car? Answer with a number in curly brackets, e.g., {9}.
™ **c:** How many prongs are there in the logo of this car? Answer with a number in curly brackets, e.g., {9}.

	(a) Audi		(b) Mercedes		(c) Maserati		(d) Audi		(e) Mercedes	
										
✦	4	✗	3	✗	3	✗	4	✗	3	✗
	4	✗	3	✗	3	✗	4	✗	3	✗
	4	✗	3	✗	3	✗	4	✗	3	✗
	4	✗	3	✗	3	✗	4	✗	3	✗
	4	✗	3	✗	3	✗	4	✗	3	✗
Bias	4	✗	3	✗	3	✗	4	✗	3	✗
GT	5	✓	4	✓	5	✓	5	✓	4	✓

✦ Gemini-2.5 Pro Sonnet-3.7 GPT-4.1 o3 o4-mini

Figure 21: VLMs are completely biased and rely entirely on prior knowledge when answering questions about ™ brand logos. Please zoom in to see the logo clearly.

Shoe logos

™ **(a), (c):** How many visible **white stripes** are there in the logo of the left shoe? Answer with a number in curly brackets, e.g., {9}.
™ **(b):** How many visible **white stylized curves** are there in the logo of the left shoe? Answer with a number in curly brackets, e.g., {9}.
™ **(d):** How many visible **black stylized curves** are there in the logo of the left shoe? Answer with a number in curly brackets, e.g., {9}.
™ **e):** How many visible **black stripes** are there in the logo of the left shoe? Answer with a number in curly brackets, e.g., {9}.

	(a) Adidas		**(b) Nike**		**(c) Adidas**		**(d) Nike**		**(e) Adidas**	
										
Gemini-2.5 Pro	3	✗	1	✗	3	✗	2	✓	3	✗
Sonnet-3.7	3	✗	1	✗	3	✗	1	✗	3	✗
GPT-4.1	3	✗	1	✗	3	✗	1	✗	3	✗
o3	3	✗	1	✗	3	✗	1	✗	4	✓
o4-mini	3	✗	1	✗	3	✗	1	✗	4	✓
Bias	3	✗	1	✗	3	✗	1	✗	3	✗
GT	4	✓	2	✓	4	✓	2	✓	4	✓

Figure 22: VLMs are often biased and rely on prior knowledge when answering questions about ™ shoe logos, even with simple ones like the Nike Swoosh. Please zoom in to see the logo clearly.

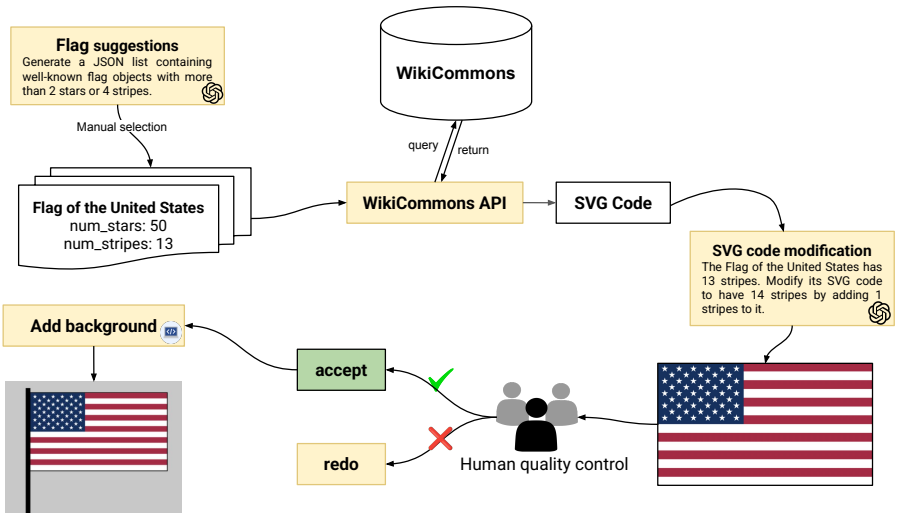


Figure 23: Data generation pipeline for Task 3: Counting stripes/stars in modified national flags.

G TASK 3: COUNTING STRIPES/STARS IN MODIFIED NATIONAL FLAGS

G.1 TASK DESIGN

Flags of countries contain easily recognizable patterns. To evaluate if existing VLMs overly rely on their knowledge of these flags to count a certain element, we design the task as follow:

- **Flag types**: We modify **2** commonly used elements across different flags: *stars* and *stripes*
- **Modification types**: Each flag has **2** types of modifications:
  - *Add*: We add an additional element (star or stripe) to a chosen flag
  - *Remove*: We remove one element (star or stripe) from a chosen flag
- **Target flags**: We select **20** well-known country flags with either 3+ stars or 5+ stripes (a total of 13 star-typed flags and 7 stripe-typed flags) to ensure the modified flags retain recognizable traits to test visual bias.
- **Image resolutions**: We generate each flag and rescale them at **3** different pixel sizes {384, 768, 1152}px using the scaling factor in Sec. 3.1 to test resolution sensitivity

This systematic approach generates a total of **20** target flags × **2** modification types × **3** resolutions = **120** total images.

G.2 IMPLEMENTATION AND IMAGE GENERATION

Implementation details We modify the SVG code of a chosen flag to create new variants following this sequence:

1. Identify 20 well-known country flags (13 with 3+ stars, 7 with 5+ stripes) based on the suggestions from `o4-mini`.
2. Retrieve original SVG code from WikiCommons for each flag.
3. Use `o4-mini` to modify each SVG to create two variants:
   - An "Add" variant with one additional element.
   - A "Remove" variant with one fewer element.
4. Render each modified flag at three different resolutions.

Quality control We employ the following steps to ensure high-quality and consistent images:

- **Manual inspection**: We manually review each generated sample to verify modification quality and visual consistency
- **Filtering**: We remove unsatisfactory samples from the benchmark and rerun the pipeline on these cases to obtain new samples.
- **Fallback**: For rare cases (3 in total) that consistently fail automated generation, we manual modify the flags to ensure they strictly follow the modification rules.

Prompts We use the following prompts:

1. **Counting questions (Q1 & Q2)**:
   - **Q1 (Star-typed flags):** *How many stars are there on this flag? Answer with a number in curly brackets, e.g., {9}.*
   - **Q1 (Stripe-typed flags):** *How many stripes are there on this flag? Answer with a number in curly brackets, e.g., {9}.*
   - **Q2 (Star-typed flags):** *Count the stars on this flag. Answer with a number in curly brackets, e.g., {9}.*
   - **Q2 (Stripe-typed flags):** *Count the stripes on this flag. Answer with a number in curly brackets, e.g., {9}.*
2. **Y/N identification questions (Q3)**:
   - *Is this the flag of [CountryName]? Answer in curly brackets, e.g., {Yes} or {No}.*

Ground truth calculation We calculate the ground truth as follow:

- **Direct counting questions (Q1 & Q2)**:
  - **Correct answer**: The actual count of the elements (stars or stripes) on the flag after modification
    - For *Remove modifications*: Standard element count minus 1
    - For *Add modifications*: Standard element count plus 1
  - **Expected bias**: The standard element count
- **Flag verification question (Q3)**:
  - **Correct answer**: "No" (since the flag's element has been modified)
  - **Expected bias**: "Yes"

G.3 Qualitative results

National Flag

(a), (b), (e): How many **stripes** are there in this flag? Answer with a number in curly brackets, e.g., {9}.
(b), (c): How many **stars** are there in this flag? Answer with a number in curly brackets, e.g., {9}.

	(a) US		(b) US		(c) EU		(d) EU		(e) Zimbabwe	
										
	13	✗	13	✗	12	✗	13	✓	7	✗
	13	✗	13	✗	12	✗	12	✗	7	✗
	13	✗	13	✗	12	✗	12	✗	7	✗
	13	✗	13	✗	12	✗	12	✗	7	✗
	13	✗	13	✗	12	✗	13	✓	7	✗
Bias	13	✗	13	✗	12	✗	12	✗	7	✗
GT	12	✓	14	✓	11	✓	13	✓	6	✓

Gemini-2.5 Pro Sonnet-3.7 GPT-4.1 o3 o4-mini

Figure 24: VLMs are biased when counting the stars and stripes on national flags.

H TASK 4: COUNTING CHESS PIECES ON MODIFIED STARTING POSITION

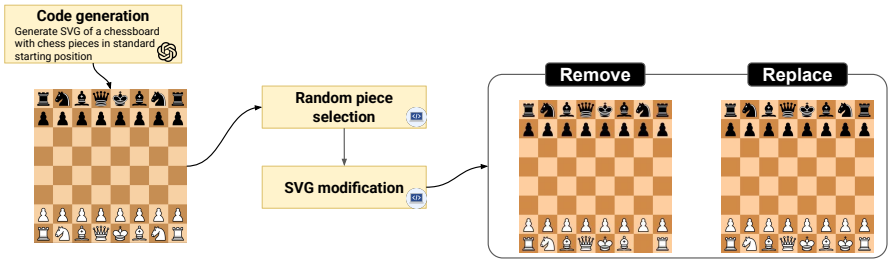


Figure 25: Data generation pipeline for Task 4: Counting chess pieces on modified starting position

H.1 TASK DESIGN

To evaluate if VLMs rely on expected structure or attend to actual pieces, we test their ability to count pieces on boards with subtle modifications. We design our task with careful control of visual parameters to ensure systematic evaluation:

- **Board types**: We use **2** different game boards: {*chess (Western chess)*, *xiangqi* (Chinese chess)}
- **Modification types**: Each board has **2** types of modifications:
  - *Remove*: We remove exactly one piece from the standard starting position.
  - *Replace*: We replace exactly one piece with a different piece of the same color.
- **Target squares**: We select **12** unique occupied squares per board type, maintaining the same target squares across the Remove and Replace modifications to ensure controlled comparison.
- **Image resolutions**: We generate each board at **3** different pixel sizes {384, 768, 1152}px to test resolution sensitivity.

This systematic approach generates a total of **2** board types × **2** modification types × **12** target squares × **3** resolutions = **144** total images.

H.2 IMPLEMENTATION AND PROMPTS

Implementation details Our implementation utilizes specialized libraries for each board type. For chess, we leverage the Python `chess` library to manipulate board states and `chess.svg` for rendering. For xiangqi (Chinese chess), we created a custom implementation using `svgwrite` for rendering.

The algorithm for both board types follows the same sequence:

1. Create a standard board with all 32 pieces in their starting positions
2. Randomly select 12 target squares from the occupied squares
3. For each target square, create (a) a Remove variant and (b) a Replace variant
4. Render each modified board at three different resolutions

The xiangqi implementation required special handling for:

- The traditional 9×10 board layout with the central river and two palaces
- Chinese character rendering for pieces, which requires detecting appropriate CJK fonts
- Different piece distribution (Chariots, Knights, Elephants, Advisors, General, Cannons, and Soldiers)

Quality control To ensure consistent image quality across all variants, we implemente several technical measures:

- **SVG to PNG conversion**: We used direct SVG rendering with adjustable scaling factors based on target resolution
- **Quality scaling**: We applied a quality multiplier ($5.0\times$ base resolution factor) to ensure clear piece visibility

Prompts We use different prompts for each modification type to test VLMs' visual attention:

1. **Remove modifications**:
   - **Q1:** *How many [chess/xiangqi] pieces are there on this board? Answer with a number in curly brackets, e.g., {9}.*
   - **Q2:** *Count the [chess/xiangqi] pieces on this board. Answer with a number in curly brackets, e.g., {9}.*
2. **Replace modifications**:
   - **Q1:** *How many [Added Piece Type] pieces are there on this board? Answer with a number in curly brackets, e.g., {9}.*
   - **Q2:** *Count the [Added Piece Type] pieces on this board? Answer with a number in curly brackets, e.g., {9}.*
3. **Both modification types**:
   - **Q3:** *Is this the [chess/xiangqi] starting position? Answer in curly brackets, e.g., {Yes} or {No}.*

For Replace modifications, [Added Piece Type] refers to the specific piece type that is added to the board through replacement, chosen from:

- For chess: Pawn, Knight, Bishop, Rook, Queen, or King
- For xiangqi: Soldier, Horse, Elephant, Chariot, Cannon, Advisor, or General

For Replace modifications, we ask about the added piece type rather than total count because this more effectively tests whether VLMs rely on prior knowledge of standard piece distributions or actually inspect the board carefully.

Ground truth calculation We calculate the ground truth answers for each prompt type:

- **Total piece count (Remove modifications only)**:
  - Correct answer: 31 (one fewer than the standard 32 pieces)
  - Expected bias: 32 (the standard piece count)
- **Added piece type count (Replace modifications only)**:
  - Correct answer: The standard count for that piece type plus one
  - For example, if a Knight is replaced with a Bishop in chess, the Bishop count would be 3 (standard 2 + 1 added)
  - Expected bias: The standard count for that piece type (e.g., 2 for Bishops in chess)
  - This tests if VLMs rely on their knowledge of standard piece counts or actually inspect the board
- **Starting position question (Both modification types)**:
  - Correct answer: Always "No" (since the board has been modified)
  - Expected bias: "Yes" (since the board closely resembles the starting position)

H.3 Qualitative results

Chess Pieces

(a): How many **chess pieces** are there on this board? Answer with a number in curly brackets, e.g., {9}.
(b): How many **Pawn pieces** are there on this board? Answer with a number in curly brackets, e.g., {9}.
(c): How many **xiangqi pieces** are there on this board? Answer with a number in curly brackets, e.g., {9}.
(d): How many **General pieces** are there on this board? Answer with a number in curly brackets, e.g., {9}.

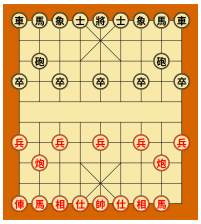
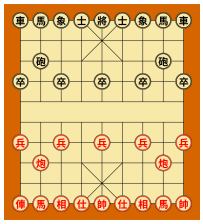
	(a) Chess		(b) Chess		(c) Xiangqi		(d) Xiangqi	
								
Gemini-2.5 Pro	32	✗	16	✗	32	✗	2	✗
Sonnet-3.7	32	✗	16	✗	32	✗	2	✗
GPT-4.1	28	✗	16	✗	32	✗	2	✗
o3	31	✓	17	✓	32	✗	2	✗
o4-mini	32	✗	17	✓	32	✗	2	✗
Bias	32	✗	16	✗	32	✗	2	✗
GT	31	✓	17	✓	31	✓	3	✓

Figure 26: VLMs are biased when counting the pieces on chess and xiangqi.

I TASK 5: COUNTING ROWS AND COLUMNS OF GAME BOARDS

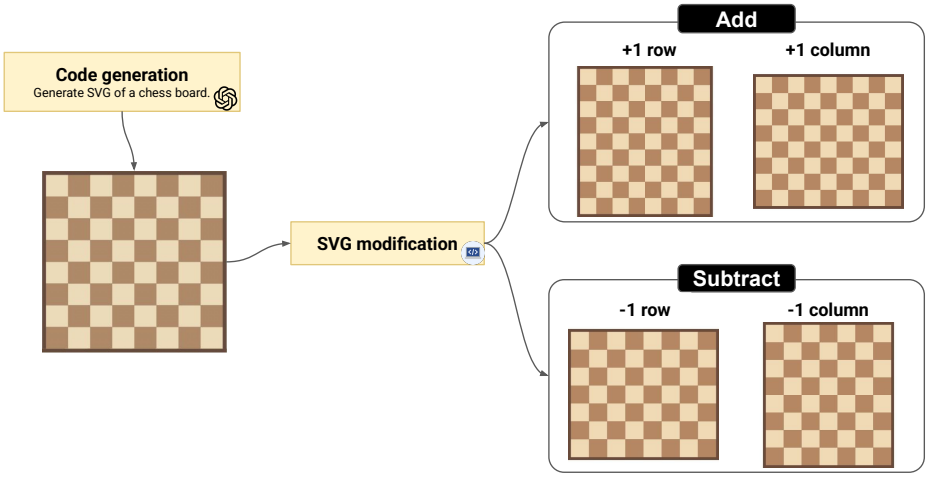


Figure 27: Data generation pipeline for Task 5: Counting rows and columns of board game

I.1 TASK DESIGN

To evaluate VLMs' over-reliance on visual bias versus actual counting, we adapted the row and column counting task from BlindTest (Rahmanzadehgervi et al., 2024) where `Claude-3.5-Sonnet` achieved 74.26% accuracy. Instead of simple grids, we leverage modified versions of well-known game boards to test whether VLMs rely on prior knowledge or perform actual visual counting. We design our task with careful control of visual parameters to ensure systematic evaluation:

- **Board types**: We use 4 different grid-based game boards: {*Chess* (8×8), *Xiangqi* (Chinese chess, 10×9), *Sudoku* (9×9), *Go* (19×19)}
- **Modification types**: Each board has up to 4 types of modifications:
  - *Remove row*: We remove exactly one row from the grid.
  - *Remove column*: We remove exactly one column from the grid.
  - *Add row*: We add exactly one row to the grid.
  - *Add column*: We add exactly one column to the grid.
- **Board-specific variations**: For Chess, Xiangqi, and Sudoku boards, all four modifications (remove/add row, remove/add column) are visually distinct, with additional positional variations (first/last), resulting in 8 variants per board. Go boards have uniform grid structure, so we produce only 4 variations.
- **Image resolutions**: We generate each board at 3 different pixel sizes {384, 768, 1152}px to test resolution sensitivity.

This systematic approach generates a total of (8 variants × 3 board types (Xiangqi/Chess/Sudoku) + 4 Go variants) × 3 resolutions = **84** total images.

I.2 IMPLEMENTATION AND PROMPTS

Implementation details Our implementation utilizes specialized drawing libraries for each board type. For Chess, we use standard 8×8 chessboard grid generation with alternating square colors. For Xiangqi, we implement the traditional 10×9 board layout with river gap and palace diagonal lines. For Sudoku, we create 9×9 grids with bold 3×3 block boundaries and sample numbers. For Go, we generate uniform line grids with traditional star points.

The algorithm for all board types follows the same sequence:

1. Create a standard board with correct dimensions and visual elements
2. Apply systematic modifications (add/remove rows/columns at specific positions)
3. Maintain visual consistency of special elements
4. Render each modified board at three different resolutions

The board-specific implementations required special handling for:

- **Chess**: Alternating light/dark square pattern preservation across dimension changes
- **Xiangqi**: River gap positioning and palace diagonal lines adjustment for row modifications
- **Sudoku**: Bold 3×3 block boundary lines based on original 9×9 grid structure
- **Go**: Uniform line spacing and star point positioning for various board sizes

Quality control To ensure consistent image quality across all variants, we implemented several technical measures:

- **SVG to PNG conversion**: We used direct SVG rendering with adjustable scaling factors based on target resolution
- **Quality scaling**: We applied a quality multiplier (5.0× base resolution factor) to ensure clear structural visibility
- **Font and layout fidelity**: Automatic detection and usage of appropriate fonts, particularly critical for Xiangqi (Chinese characters) and Sudoku (numbers)

Prompts We use different prompts for different question types to test VLMs' visual counting versus prior knowledge:

1. **Counting questions (Q1 & Q2)**:
   - **Q1 (Chess):** *How many [rows/columns] are there on this board? Answer with a number in curly brackets, e.g., {9}.*
   - **Q1 (Xiangqi, Go):** *How many [horizontal/vertical] are there on this board? Answer with a number in curly brackets, e.g., {9}.*
   - **Q1 (Sudoku):** *How many [rows/columns] are there on this puzzle? Answer with a number in curly brackets, e.g., {9}.*
   - **Q2 (Chess):** *Count the [rows/columns] on this board. Answer with a number in curly brackets, e.g., {9}.*
   - **Q2 (Xiangqi, Go):** *Count the [horizontal/vertical] lines on this board. Answer with a number in curly brackets, e.g., {9}.*
   - **Q2 (Sudoku):** *Count the [rows/columns] on this puzzle. Answer with a number in curly brackets, e.g., {9}.*
2. **Y/N identification questions (Q3)**:
   - **Q3 (Chess):** *Is this a 8×8 Chessboard? Answer in curly brackets, e.g., {Yes} or {No}.*
   - **Q3 (Xiangqi):** *Is this a 10×9 Xiangqi board? Answer in curly brackets, e.g., {Yes} or {No}.*
   - **Q3 (Sudoku):** *Is this a 9×9 Sudoku puzzle? Answer in curly brackets, e.g., {Yes} or {No}.*
   - **Q3 (Go):** *Is this a 19×19 Go board? Answer in curly brackets, e.g., {Yes} or {No}.*

Ground truth calculation We calculate the ground truth answers for each prompt type:

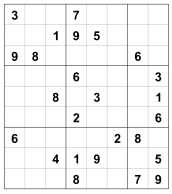
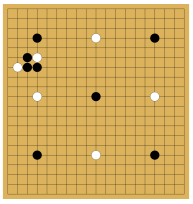
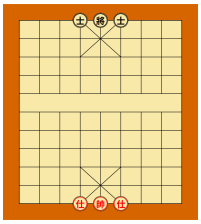
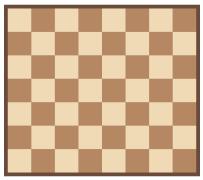
- **Row/Column count (Q1 & Q2)**:
  - **Correct answer**: The actual number of rows/columns after modification. For example, if one row is removed from a 9×9 Sudoku, the row count is 8.
  - **Expected bias**: The standard count for that board type (e.g., 8 for Chess rows, 10 for Xiangqi horizontal lines, 9 for Sudoku rows, 19 for Go horizontal lines)
- **Standard layout question (Q3)**:

- **Correct answer**: Always "No" (since all boards have been modified from standard dimensions)
- **Expected bias**: "Yes" (since the boards closely resemble their standard counterparts)

I.3 QUALITATIVE RESULTS

Game Boards

(a): How many **columns** are there on this puzzle? Answer with a number in curly brackets, e.g., {9}.
(b), (c): How many **horizontal lines** are there on this board? Answer with a number in curly brackets, e.g., {9}.
(d): How many **rows** are there on this board? Answer with a number in curly brackets, e.g., {9}.

	(a) Sudoku		(b) Go		(c) Xiangqi		(d) Chess	
Gemini-2.5 Pro	9	✗	13	✗	10	✗	6	✗
Sonnet-3.7	9	✗	19	✗	10	✗	8	✗
GPT-4.1	9	✗	19	✗	10	✗	8	✗
o3	9	✗	19	✗	10	✗	8	✗
o4-mini	9	✗	19	✗	12	✗	8	✗
Bias	9	✗	19	✗	10	✗	8	✗
GT	8	✓	20	✓	11	✓	7	✓

Figure 28: VLMs are biased when counting the rows and columns on game boards.

J TASK 6: VISUAL TESTING WITH BOTH ORIGINAL AND MODIFIED OPTICAL ILLUSION

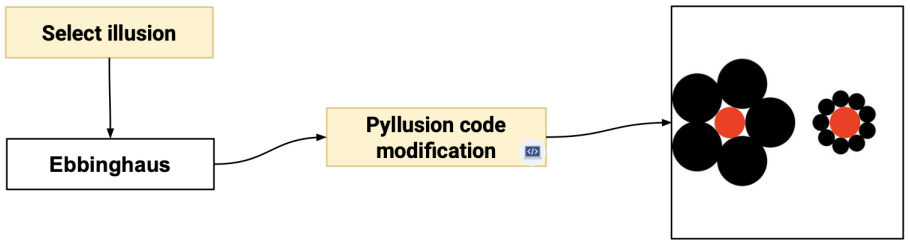


Figure 29: Data generation pipeline for Task 6: Visual testing with both original and modified optical illusion

J.1 TASK DESIGN

Recent VLMs show improved performance on optical illusion tasks, with `o4-mini` achieving 71.49% accuracy on IllusionVQA. However, these VLMs might have merely memorized the common optical illusions rather than truly perceiving visual information. To investigate this hypothesis, we test their ability to correctly identify illusion effects on both original and strategically modified versions. We design our task with careful control of visual parameters to ensure systematic evaluation:

- **Illusion types**: We use **6** different classical optical illusions: {*Ebbinghaus*, *Müller-Lyer*, *Ponzo*, *Vertical-Horizontal*, *Zöllner*, *Poggendorff*}
- **Condition types**: Each illusion has **2** conditions:
  - *Original*: Standard illusion where the visual effect should occur (e.g., two identical circles appearing different sizes).
  - *Modified*: Reversed version where the actual measurements contradict the typical illusion effect (e.g., circles that are genuinely different sizes).
- **Parameter variations**: We generate **multiple combinations** of illusion parameters:
  - Most illusions: 12 original + 12 modified versions with varying illusion strength and difference
  - Vertical-Horizontal: 6 original + 6 modified versions (fixed T-shape structure)
- **Image resolutions**: We generate each illusion at **3** different pixel sizes {384, 768, 1152}px to test resolution sensitivity.

This systematic approach generates a total of (12 original + 12 modified) $\times$ **5** illusion types + (6 original + 6 modified) $\times$ **1** Vertical-Horizontal illusion) $\times$ **3** resolutions = **396** total images.

J.2 IMPLEMENTATION AND PROMPTS

Implementation details Our implementation adapts code from Pyllusion (`https://github.com/RealityBending/Pyllusion`) to generate consistent, parametrically controlled optical illusions. We systematically vary two key parameters: *illusion strength* (which controls the intensity of contextual elements that create the illusion effect, representing how strongly the surrounding context biases perceptual experience) and *difference* (which controls the objective, actual difference between target elements being compared, where 0 means identical elements and non-zero values create genuine physical differences).

The algorithm for all illusion types follows the same sequence:

1. Define parameter ranges for each illusion type (strength values, difference values).
2. Generate original versions with standard illusion parameters (diff=0 for equal elements).

3. Generate modified versions with reversed parameters (diff≠0 for unequal elements).
4. Render each illusion variant at three different resolutions.

The illusion-specific implementations required special parameter handling for:

- **Ebbinghaus**: Varying surrounding circle sizes (strength) and central circle differences (difference).
- **Müller-Lyer**: Different arrowhead angles (strength) and line length differences (difference).
- **Ponzo**: Perspective line angles (strength) and horizontal bar length differences (difference).
- **Vertical-Horizontal**: Fixed T-shape with varying line length ratios (difference).
- **Zöllner**: Background line angles (strength) and main line parallelism differences (difference).
- **Poggendorff**: Interrupting rectangle positions (strength) and diagonal line alignments (difference).

Quality control To ensure consistent image quality and valid illusion effects across all variants, we implemented several technical measures:

- **Parameter validation**: Ensured all strength and difference values produce visually meaningful illusions, with diff≠ 0 cases design to be easily recognizable by humans to distinguish actual physical differences from perceptual biases clearly.
- **Balanced generation**: Equal numbers of diff=0 (original) and diff≠0 (modified) cases per illusion type

Prompts We use consistent prompts across illusion types to test VLMs' visual perception versus memorized knowledge:

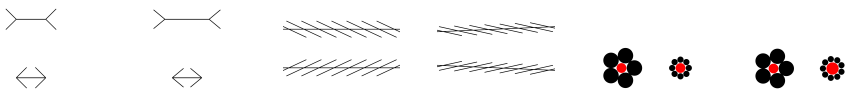
1. **Main questions (Q1 & Q2)**:
   - **Q1 (Ebbinghaus):** *Are the two red circles equal in size? Answer in curly brackets, e.g., {Yes} or {No}.*
   - **Q1 (Müller-Lyer, Ponzo):** *Are the two horizontal lines equal in length? Answer in curly brackets, e.g., {Yes} or {No}.*
   - **Q1 (Vertical-Horizontal):** *Are the horizontal and vertical lines equal in length? Answer in curly brackets, e.g., {Yes} or {No}.*
   - **Q1 (Zöllner):** *Are the two horizontal lines parallel? Answer in curly brackets, e.g., {Yes} or {No}.*
   - **Q1 (Poggendorff):** *Are the two diagonal line segments aligned? Answer in curly brackets, e.g., {Yes} or {No}.*
   - **Q2 (Ebbinghaus):** *Do the two red circles have the same size? Answer in curly brackets, e.g., {Yes} or {No}.*
   - **Q2 (Müller-Lyer):** *Do the two horizontal lines have the same length? Answer in curly brackets, e.g., {Yes} or {No}.*
   - **Q2 (Ponzo):** *Do the two horizontal lines have the same length? Answer in curly brackets, e.g., {Yes} or {No}.*
   - **Q2 (Vertical-Horizontal):** *Do the horizontal and vertical lines have the same length? Answer in curly brackets, e.g., {Yes} or {No}.*
   - **Q2 (Zöllner):** *Do the two horizontal lines run parallel? Answer in curly brackets, e.g., {Yes} or {No}.*
   - **Q2 (Poggendorff):** *Do the two diagonal lines form a straight line? Answer in curly brackets, e.g., {Yes} or {No}.*
2. **Y/N identification questions (Q3)**:
   - **Q3:** *Is this an example of the [Ebbinghaus/Müller-Lyer/Ponzo/Vertical-Horizontal/Zöllner/Poggendorff] illusion? Answer in curly brackets, e.g., {Yes} or {No}.*

Ground truth calculation We calculate the ground truth answers based on the actual measurements in each image:

- **Counting questions (Q1 & Q2)**:
  - **Correct answer**:
    - **Original illusions (diff=0)**: Elements are actually equal, so the correct answer is "Yes"
    - **Modified illusions (diff≠0)**: Elements are actually different, so the correct answer is "No"
  - **Expected bias**:
    - **Original illusions**: VLMs might incorrectly say "No" expecting the illusion effect to make equal elements appear different
    - **Modified illusions**: VLMs might incorrectly say "Yes" expecting the illusion to make genuinely different elements appear equal
- **Y/N identification questions (Q3)**:
  - **Correct answer**:
    - **Original illusions**: "Yes" (standard examples of the specified illusion type).
    - **Modified illusions**: "No" (modified versions that contradict typical illusion effects).
  - **Expected bias**:
    - **Original illusions**: VLMs likely correctly identify as "Yes" since they match memorized illusion patterns
    - **Modified illusions**: VLMs may incorrectly say "Yes" if they rely on visual similarity rather than recognizing the effect contradiction

J.3 Qualitative results

Abstract images: Optical Illusions

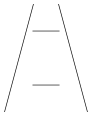
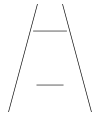
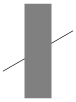
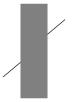


	(a) Original Müller-Lyer		(b) Modified Müller-Lyer		(c) Original Zöllner		(d) Modified Zöllner		(e) Original Ebbinghaus		(f) Modified Ebbinghaus	
Gemini-2.5 Pro	Yes	✓	Yes	✗	Yes	✓	Yes	✗	Yes	✓	Yes	✗
Sonnet-3.7	Yes	✓	Yes	✗	Yes	✓	Yes	✗	No	✗	No	✓
GPT-4.1	Yes	✓	Yes	✗	Yes	✓	Yes	✗	Yes	✓	Yes	✗
o3	Yes	✓	Yes	✗	Yes	✓	Yes	✗	Yes	✓	Yes	✗
o4-mini	Yes	✓	Yes	✗	Yes	✓	Yes	✗	No	✗	Yes	✗
Bias	No	✗	Yes	✗	No	✗	Yes	✗	No	✗	Yes	✗
GT	Yes	✓	No	✓	Yes	✓	No	✓	Yes	✓	No	✓

(a), (b): Are the two horizontal lines **equal** in length? Answer in curly brackets, e.g., {Yes} or {No}.
(c), (d): Are the two horizontal lines **parallel**? Answer in curly brackets, e.g., {Yes} or {No}.
(e), (f): Are the two red circles **equal** in size? Answer in curly brackets, e.g., {Yes} or {No}.

Figure 30: VLMs show systematic biases, often relying on prior knowledge of optical illusions rather than directly interpreting the image.

Abstract images: Optical Illusions

	(a) Original Ponzo		(b) Modified Ponzo		(c) Original V-H		(d) Modified V-H		(e) Original Poggendorff		(f) Modified Poggendorff	
												
Gemini-2.5 Pro	Yes	✓	No	✓	No	✗	No	✓	Yes	✓	Yes	✗
Sonnet-3.7	Yes	✓	Yes	✗	No	✗	No	✓	No	✗	No	✓
GPT-4.1	Yes	✓	Yes	✗	No	✗	No	✓	Yes	✓	Yes	✗
o3	Yes	✓	Yes	✗	No	✗	No	✓	Yes	✓	Yes	✗
o4-mini	Yes	✓	Yes	✗	No	✗	No	✓	Yes	✓	Yes	✗
Bias	No	✗	Yes	✗	No	✗	Yes	✗	No	✗	Yes	✗
GT	Yes	✓	No	✓	Yes	✓	No	✓	Yes	✓	No	✓

(a), (b): Are the two horizontal lines **equal** in length? Answer in curly brackets, e.g., {Yes} or {No}.
(c), (d): Are the horizontal and vertical lines **equal** in length? Answer in curly brackets, e.g., {Yes} or {No}.
(e), (f): Are the two diagonal line segments **aligned**? Answer in curly brackets, e.g., {Yes} or {No}.

Figure 31: VLMs show systematic biases, often relying on prior knowledge of optical illusions (e.g., Ponzo and Poggendorff illusions) rather than directly interpreting the image. In contrast, in the vertical–horizontal illusion, VLMs respond like humans. They are misled by the illusion itself, leading them to answer the original question incorrectly rather than the counterfactual ones.

K TASK 7: COUNTING CIRCLES OR LINES IN AN ANOMALY CELL WITHIN A PATTERNED GRID

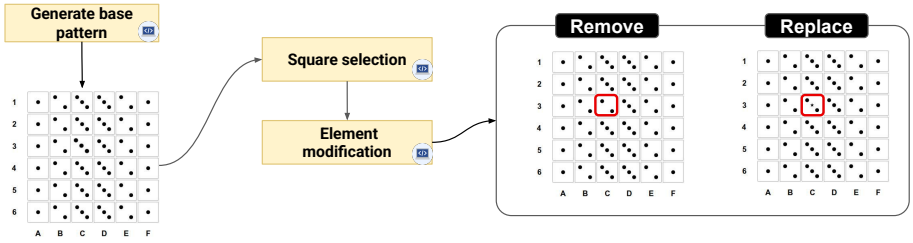


Figure 32: Data generation pipeline for Task 7: Counting circles or lines in an anomaly cell within a patterned grid

K.1 TASK DESIGN

VLMs can infer patterns from nearby visual elements to answer visual questions (Huang et al., 2024). To evaluate whether VLMs rely on pattern recognition over actual visual counting, we create square grids with systematic numerical patterns (represented visually by dice faces or tally marks) where exactly one cell violates the expected pattern. We hypothesize that VLMs will prioritize the inferred pattern over the actual visual information and report the expected pattern-completing value instead of the true count. We design our task with careful control of visual parameters to ensure systematic evaluation:

- **Grid types**: We use 2 different visual representation types: {*dice* (circular dots in dice-face patterns), *tally* (traditional tally mark lines)}.
- **Modification types per grid type**: For each grid type, we apply 2 distinct types of cell-level modifications:
  - *Dice grids*: Remove (one dot is removed from a cell) and Replace (one dot is replaced with a different shape, like a square or star, within a cell).
  - *Tally grids*: Remove (one tally line is removed from a cell) and Add (one extra tally line is added to a cell).
- **Grid Dimensions**: We generate grids of 7 different dimensions, ranging from 6×6 to 12×12 cells.
- **Unique scenarios for anomaly placement (single anomaly per grid image)**: To create 14 distinct base settings for placing anomalies, **where each final grid image will feature only a single modified cell**. We proceed as follows: for each of the 7 grid dimensions, we define two separate base settings. Each of these two settings for a given grid dimension involves selecting a *different*, unique cell location to be the *sole* anomaly cell for images generated under that specific setting. These potential anomaly cell locations are carefully chosen to avoid edges and corners. This gives us (7 grid dimensions × 2 distinct choices of a single anomaly cell location per dimension) = 14 distinct base settings. For each of these 14 base settings (defined by a grid dimension and the location of its single anomaly cell), we then apply all combinations of grid types and their respective modifications to generate the final images, each still containing only that one pre-determined anomaly.
- **Image resolutions**: Each generated grid image is rendered at 3 different pixel sizes {384, 768, 1152}px to assess sensitivity to image resolution.

This systematic generation process yields a total of 2 (grid types) × 2 (modification types) × 14 (unique scenarios) × 3 (resolutions) = **168** distinct images.

K.2 IMPLEMENTATION AND PROMPTS

Implementation details Our implementation generates systematic pattern grids using a distance-from-edge algorithm to create naturally increasing-then-decreasing numerical patterns. For dice grids, we use circular dots arranged in traditional dice-face configurations (1-6 dots per cell). For tally grids, we render authentic tally marks with proper grouping (four vertical lines crossed by a diagonal fifth line).

The algorithm for both grid types follows the same sequence:

1. Generate base grid with pattern-consistent cell counts using distance-from-edge calculation
2. Organize target positions across 14 groups, with each group containing both dice and tally variants
3. For each target cell, create modification variants:
   - **Dice**: Remove one dot OR replace one dot with alternative shape (triangle, square, star)
   - **Tally**: Remove one line OR add one extra line
4. Render each modified grid at three different resolutions with consistent visual quality

The grid-specific implementations required special handling for:

- **Dice pattern consistency**: Maintaining standard dice-face arrangements (1-6 dots) while allowing single-dot modifications
- **Tally mark authenticity**: Proper grouping of marks with diagonal crosses for every fifth line
- **Pattern calculation**: Distance-from-edge algorithm ensuring natural numerical progression across grid cells
- **Cell positioning**: Strategic selection of anomaly cells away from edges to preserve pattern context

Quality control To ensure consistent image quality and valid pattern recognition challenges across all variants, we implemented several technical measures:

- **SVG to PNG conversion**: We used direct SVG rendering with adjustable scaling factors based on target resolution
- **Quality scaling**: We applied a quality multiplier (5.0× base resolution factor) to ensure clear shape and line visibility

Prompts We use consistent prompts across both grid types to test VLMs' pattern recognition versus actual visual counting:

1. **Counting questions (Q1 & Q2)**:
   - **Q1 (Dice):** *How many circles are there in cell [CellID]? Answer with a number in curly brackets, e.g., {9}.*
   - **Q1 (Tally):** *How many lines are there in cell [CellID]? Answer with a number in curly brackets, e.g., {9}.*
   - **Q2 (Dice):** *Count the circles in cell [CellID]. Answer with a number in curly brackets, e.g., {9}.*
   - **Q2 (Tally):** *Count the lines in cell [CellID]. Answer with a number in curly brackets, e.g., {9}.*
2. **Y/N identification questions (Q3)**:
   - **Q3 (Dice):** *Does cell [CellID] contain [ExpectedCount] circles? Answer in curly brackets, e.g., {Yes} or {No}.*
   - **Q3 (Tally):** *Does cell [CellID] contain [ExpectedCount] lines? Answer in curly brackets, e.g., {Yes} or {No}.*

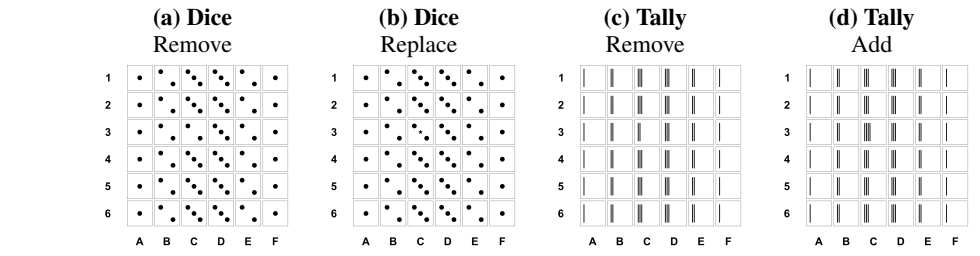
For all prompts, [CellID] refers to the specific anomaly cell using standard spreadsheet notation (e.g., C3, F7), and [ExpectedCount] represents the pattern-consistent count that would be expected based on surrounding cells.

Ground truth calculation We calculate the ground truth answers based on the actual visual content in each modified cell:

- **Direct counting questions (Q1 & Q2)**:
  - **Correct answer**: The actual count of visual elements in the target cell after modification
    - For *Remove modifications*: Standard pattern count minus 1
    - For *Add modifications*: Standard pattern count plus 1
    - For *Replace modifications*: Standard pattern count minus 1 (since one circle is replaced with a different shape)
  - **Expected bias**: The pattern-consistent count that VLMs might infer from surrounding cells, ignoring the actual modification
- **Pattern-based verification question (Q3)**:
  - **Correct answer**: Always "No" (since the target cell has been modified to break the pattern)
  - **Expected bias**: "Yes" (if VLMs rely on pattern inference rather than direct visual inspection)

K.3 Qualitative results

Abstract Images: Patterned Grid



	(a) Dice Remove		(b) Dice Replace		(c) Tally Remove		(d) Tally Add	
Gemini-2.5 Pro	3	✗	3	✗	2	✓	3	✗
Sonnet-3.7	2	✓	2	✓	2	✓	4	✓
GPT-4.1	3	✗	3	✗	3	✗	4	✓
o3	3	✗	2	✓	3	✗	4	✓
o4-mini	3	✗	3	✗	2	✓	3	✗
Bias	3	✗	3	✗	3	✗	3	✗
GT	2	✓	2	✓	2	✓	4	✓

(a), (b): How many **circles** are there in cell **C3**? Answer with a number in curly brackets, e.g., {9}.
(c), (d): How many **lines** are there in cell **C3**? Answer with a number in curly brackets, e.g., {9}.

Figure 33: All VLMs, except Sonnet-3.7, fail to correctly identify the abnormal cell (C3) in both the patterned grids.

L DETAILS OF PROMPTS

L.1 EXAMPLES OF Q1, Q2 AND Q3

Table 30: Some examples of questions on animal, brand logos, and flags

Topic	Subtopic	Q1	Q2	Q3
Animal		How many legs does this animal have? Answer with a number in curly brackets, e.g., {9}.	Count the legs of this animal. Answer with a number in curly brackets, e.g., {9}.	Is this an animal with 4 legs? Answer in curly brackets, e.g., {Yes} or {No}.
Logo	Adidas	How many visible stripes are there in the logo of the left shoe? Answer with a number in curly brackets, e.g., {9}.	Count the visible stripes in the logo on the left shoe. Answer with a number in curly brackets, e.g., {9}.	Are the logos on these shoes Adidas logos? Answer in curly brackets, e.g., {Yes} or {No}.
	Nike	How many visible white stylized curves are there in the logo of the left shoe? Answer with a number in curly brackets, e.g., {9}.	Count the visible white stylized curves in the logo on the left shoe. Answer with a number in curly brackets, e.g., {9}.	Are the logos on these shoes Nike logos? Answer in curly brackets, e.g., {Yes} or {No}.
	Mercedes	How many points are there on the star in the logo of this car? Answer with a number in curly brackets, e.g., {9}.	Count the points on the star in the logo of this car. Answer with a number in curly brackets, e.g., {9}.	Is the logo on this car Mercedes-Benz logo? Answer in curly brackets, e.g., {Yes} or {No}.
	Audi	How many overlapping circles are there in the logo of this car? Answer with a number in curly brackets, e.g., {9}.	Count the overlapping circles in the logo of this car. Answer with a number in curly brackets, e.g., {9}.	Is the logo on this car Audi logo? Answer in curly brackets, e.g., {Yes} or {No}.
	Maserati	How many prongs are there in the logo of this car? Answer with a number in curly brackets, e.g., {9}.	Count the prongs in the logo of this car. Answer with a number in curly brackets, e.g., {9}.	Is the logo on this car Maserati logo? Answer in curly brackets, e.g., {Yes} or {No}.
Flag	Stars	How many stars are there on this flag? Answer with a number in curly brackets, e.g., {9}.	Count the stars on this flag. Answer with a number in curly brackets, e.g., {9}.	Is this the flag of [country]? Answer in curly brackets, e.g., {Yes} or {No}.
	Stripes	How many stripes are there on this flag? Answer with a number in curly brackets, e.g., {9}.	Count the stripes on this flag. Answer with a number in curly brackets, e.g., {9}.	Is this the flag of [country]? Answer in curly brackets, e.g., {Yes} or {No}.

Table 31: Some examples of questions on chess pieces, game boards, and patterned grids.

Topic	Subtopic	Q1	Q2	Q3
Chess Pieces	Chess	How many chess pieces are there on this board? Answer with a number in curly brackets, e.g., {9}.	Count the chess pieces on this board. Answer with a number in curly brackets, e.g., {9}.	Is this the chess starting position? Answer in curly brackets, e.g., {Yes} or {No}.
	Xiangqi	How many xiangqi pieces are there on this board? Answer with a number in curly brackets, e.g., {9}.	Count the xiangqi pieces on this board. Answer with a number in curly brackets, e.g., {9}.	Is this the Xiangqi starting position? Answer in curly brackets, e.g., {Yes} or {No}.
Board Game	Chess	How many rows are there on this board? Answer with a number in curly brackets, e.g., {9}.	Count the rows on this board. Answer with a number in curly brackets, e.g., {9}.	Is this a 8x8 Chessboard? Answer in curly brackets, e.g., {Yes} or {No}.
	Xiangqi	How many horizontal lines are there on this board? Answer with a number in curly brackets, e.g., {9}.	Count the horizontal lines on this board. Answer with a number in curly brackets, e.g., {9}.	Is this a 10x9 Xiangqi board? Answer in curly brackets, e.g., {Yes} or {No}.
	Go	How many horizontal lines are there on this board? Answer with a number in curly brackets, e.g., {9}.	Count the horizontal lines on this board. Answer with a number in curly brackets, e.g., {9}.	Is this a 19x19 Go board? Answer in curly brackets, e.g., {Yes} or {No}.
	Sudoku	How many rows are there on this puzzle? Answer with a number in curly brackets, e.g., {9}.	Count the rows on this puzzle. Answer with a number in curly brackets, e.g., {9}.	Is this a 9x9 Sudoku puzzle? Answer in curly brackets, e.g., {Yes} or {No}.
Patterned Grid	Dice	How many circles are there in cell C5? Answer with a number in curly brackets, e.g., {9}.	Count the circles in cell C5. Answer with a number in curly brackets, e.g., {9}.	Does cell C5 contain 4 circles? Answer in curly brackets, e.g., {Yes} or {No}.
	Tally	How many lines are there in cell C5? Answer with a number in curly brackets, e.g., {9}.	Count the lines in cell C5. Answer with a number in curly brackets, e.g., {9}.	Does cell C5 contain 3 lines? Answer in curly brackets, e.g., {Yes} or {No}.

Table 32: Some examples of questions on optical illusions.

Topic	Subtopic	Q1	Q2	Q3
Optical Illusion	Ebbinghaus	Are the two red circles equal in size? Answer in curly brackets, e.g., {Yes} or {No}.	Do the two red circles have the same size? Answer in curly brackets, e.g., {Yes} or {No}.	Is this an example of the Ebbinghaus illusion? Answer in curly brackets, e.g., {Yes} or {No}.
	Mullerlyer	Are the two horizontal lines equal in length? Answer in curly brackets, e.g., {Yes} or {No}.	Do the two horizontal lines have the same length? Answer in curly brackets, e.g., {Yes} or {No}.	Is this an example of the Müller-Lyer illusion? Answer with Yes/No. Answer in curly brackets, e.g., {Yes} or {No}.
	Poggendorff	Are the two diagonal line segments aligned? Answer in curly brackets, e.g., {Yes} or {No}.	Do the two diagonal lines form a straight line? Answer in curly brackets, e.g., {Yes} or {No}.	Is this an example of the Poggendorff illusion? Answer in curly brackets, e.g., {Yes} or {No}.
	Ponzo	Are the two horizontal lines equal in length? Answer in curly brackets, e.g., {Yes} or {No}.	Do the two horizontal lines have the same length? Answer in curly brackets, e.g., {Yes} or {No}.	Is this an example of the Ponzo illusion? Answer in curly brackets, e.g., {Yes} or {No}.
	VerticalHorizontal	Are the horizontal and vertical lines equal in length? Answer in curly brackets, e.g., {Yes} or {No}.	Do the horizontal and vertical lines have the same length? Answer in curly brackets, e.g., {Yes} or {No}.	Is this an example of the Vertical–Horizontal illusion? Answer in curly brackets, e.g., {Yes} or {No}.
	Zollner	Are the two horizontal lines parallel? Answer in curly brackets, e.g., {Yes} or {No}.	Do the two horizontal lines run parallel? Answer in curly brackets, e.g., {Yes} or {No}.	Is this an example of the Zöllner illusion? Answer in curly brackets, e.g., {Yes} or {No}.

L.2 PROMPTS USED FOR IMAGE GENERATION AND IMAGE EDITING

Table 33: Prompts used for image generation and image editing with Gemini-2.0 Flash and GPT-4o by topic and prompt type

Topic	Prompt type	Prompt
Animals	Animal suggestions	Generate a JSON list containing 100 animal objects. Each object should represent a common animal and follow the structure below: { "name": "<Common Animal Name>", "num_legs": <Typical Number of Legs> } Ensure the following for each animal: 1. the number of legs of this animal is 2 or 4. 2. the animal's legs must be long enough to be seen easily from the body using a side-view perspective. Prioritize animals whose legs are thin and/or long.
	Animal generation	Generate a clear, full-body, side-view image of a(n) {animal} with {num_legs} legs that is walking in a real-world natural background. The {num_legs}-legged animal must look photo-realistic in nature. All {num_legs} legs must be clearly visible.
	Animal editing	Edit this image: Add 1 more leg to the {animal} so that it has {num_leg} legs in total. The {num_leg}-legged {animal} must be photo-realistic. All {num_leg} legs must be clearly visible.
Flags	Flag suggestions	Generate a JSON list of flags objects. Each object should represent a well-known flags and follow the structure below: { "name": "<Flag Name>", "original_stripes" or "original_stars": <Number of Stripes or Stars (whichever applicable)> } 1. Ensure that the number of stars is more than 3, and the number of stripes is at least 5. 2. Ensure that the flag does not contain any other geometrically complex elements (depicting of animal, letters, etc.). 3. Prioritize well-known flags.
	Flag SVG code editing	You are an expert in editing SVG image code. Modify the SVG code of the flag of {country} according to the following instruction: Instruction: "The flag of {country} has {num_ele} {element}. Modify the SVG code so that it has num_ele + 1 {element} instead. Make sure the modified {element} are natural looking and integrate seamlessly on the new flag." Base SVG code: {svg_code} 1. Modify the base SVG by adding or removing the mentioned feature (stars, stripes, etc.) according to the instruction above. 2. Wrap the entire SVG in <code></code>. Do not explain anything.

Table 34: Prompts used for image generation and image editing with ✦⚡Gemini-2.0 Flash and GPT-4o by topic and prompt type

Topic	Prompt type	Prompt
Logos	Logo suggestion	Generate a JSON list of subtle logo modification prompts and corresponding VLM question prompts to test visual bias. For each entry: Slightly modify the visual components of a well-known car or sportswear logo. The selected logo must be geometrically simple and widely recognized. You must include a generation prompt to create the altered image. Include a question prompt (e.g., "How many..."). Include metadata: element being modified, actual count (ground truth), common expected count (bias). <In-context learning example 1> <In-context learning example 2>
	Shoe generation	Generate an {shoe_brand} style running shoe but with {actual_count} {modified_element} instead of {expected_bias}.
	Shoe background generation	Generate a side-view image of an athlete wearing this pair of shoes. Keep all the fine-grained details of the shoes, particularly the {actual_count} {modified_element} on both shoes. The person is playing {sports_type}, showing their sports_type skills, and is wearing a {sports_type} outfit. Zoom out a bit to see their full body.
	Car logo generation	Generate a {car_brand} logo but with {actual_count} {modified_element} instead of {expected_bias}.
	Car background generation	Generate a photo-realistic front-view image of a {color} {car_brand} {body_type} on the road in the middle of the day. Zoom out a bit so that we can see the road.

L.3 PROMPTS FOR SANITY CHECK

Table 35: Examples of Sanity check questions

Topic	Identification questions	Counting/Illusion questions
Animal	What animal is this? Answer in curly brackets, e.g., {Fish}.	How many legs do this animal have? Answer with a number in curly brackets, e.g., {9}.
Logo	What car logo is this? Answer in curly brackets, e.g., {Toyota}.	How many overlapping circles are there on the logo of this car? Answer with a number in curly brackets, e.g., {9}.
Flags	What country flag is this? Answer in curly brackets, e.g., {Flag of Vietnam}.	How many stars are there in this flag? Answer with a number in curly brackets, e.g., {9}.
Chess Pieces	What board game is this? Answer in curly brackets, e.g., {Shogi}.	How many chess pieces are there on this board? Answer with a number in curly brackets, e.g., {9}.
Game Boards	What board game is this? Answer in curly brackets, e.g., {Shogi}.	How many rows are there on this board? Answer with a number in curly brackets, e.g., {9}.
Optical Illusions	What optical illusion is this? Answer in curly brackets, e.g., {Delboeuf illusion}.	This image shows the Ebbinghaus illusion. What question does this illusion typically ask, and what is the correct answer?

M ADDITIONAL QUALITATIVE RESULTS

M.1 QUALITATIVE RESULTS ON THE USE OF HELPFUL PROMPTS

How many legs does this animal have? Answer with a number in curly brackets, e.g., {9}.

Baseline: How many legs does this animal have? Answer with a number in curly brackets, e.g., {9}.
w/ Double-Check: Please double-check your answer and give your final answer in curly brackets, following the format above.
w/ Debiased Prompt: Do not assume from prior knowledge and answer only based on what is visible in the image.
Ground truth: 3 ✓

Gemini-2.5 Pro (Baseline): {2} ✗
Gemini-2.5 Pro (w/ Double-Check): {2} ✗
Gemini-2.5 Pro (w/ Debiased Prompt): {2} ✗

Sonnet-3.7 (Baseline): The image shows a rooster (male chicken) standing on a grassy field. Chickens have two legs, which are clearly visible in this image. The rooster is standing upright on its two pale yellow/tan legs with its feet planted on the grass. {2} ✗
Sonnet-3.7 (w/ Double-Check): Looking at the image again, I can clearly see the rooster is standing on two legs. Chickens are bipedal animals, with two legs that they use for walking and standing. In this particular photo, both of the chicken's legs are visible beneath its body. {2} ✗
Sonnet-3.7 (w/ Debiased Prompt): Based solely on what is visible in the image, this chicken has {2} legs. The image clearly shows a rooster standing on a grassy field with its two yellow/tan-colored legs and feet visible beneath its body. ✗

GPT-4.1 (Baseline): {4} ✗
GPT-4.1 (w/ Double-Check): {4} ✗
GPT-4.1 (w/ Debiased Prompt): {4} ✗

---

o3 (Baseline): {2} ✗
o3 (w/ Double-Check): {2} ✗
o3 (w/ Debiased Prompt): {4} ✗

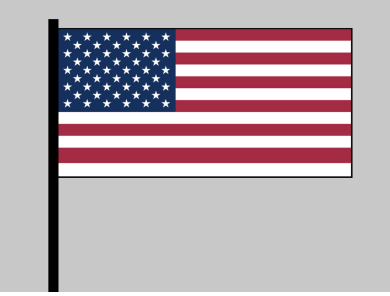
---

o4-mini (Baseline): {2} ✗
o4-mini (w/ Double-Check): {2} ✗
o4-mini (w/ Debiased Prompt): {4} ✗

Figure 34: VLMs fail 100% of the time, even on simple tasks like counting chicken legs, despite helpful prompts such as debiasing or double-checking.

How many stripes are there in this flag? Answer with a number in curly brackets, e.g., {9}.

Baseline: How many stripes are there in this flag? Answer with a number in curly brackets, e.g., {9}.
w/ Double-Check: Please double-check your answer and give your final answer in curly brackets, following the format above.
w/ Debiased Prompt: Do not assume from prior knowledge and answer only based on what is visible in the image.
Ground truth: 12 ✓



Gemini-2.5 Pro (Baseline): {13} ✗
Gemini-2.5 Pro (w/ Double-Check): {13} ✗
Gemini-2.5 Pro (w/ Debiased Prompt): {13} ✗

Sonnet-3.7 (Baseline): The image shows the flag of the United States of America. The American flag has 13 alternating red and white horizontal stripes (7 red and 6 white), representing the original 13 colonies. The flag also features a blue rectangle in the upper left corner containing 50 white stars, representing the 50 states. To answer your specific question: {13} ✗
Sonnet-3.7 (w/ Double-Check): I'll double-check the American flag's design. The flag of the United States consists of:
- A blue rectangle in the upper left corner (the union) containing 50 white stars
- Horizontal stripes alternating between red and white. Looking more carefully at the image, I can count 13 horizontal stripes total (7 red and 6 white).
The final answer is: {13} ✗
Sonnet-3.7 (w/ Debiased Prompt): Based only on what is visible in the image, I can see the American flag with its distinctive design. The flag has alternating red and white horizontal stripes, and in the image shown, I can count a total of {7} stripes. There are 4 red stripes and 3 white stripes visible in this particular rendering of the flag. ✗

GPT-4.1 (Baseline): {13} ✗
GPT-4.1 (w/ Double-Check): {13} ✗
GPT-4.1 (w/ Debiased Prompt): {13} ✗

---

o3 (Baseline): {13} ✗
o3 (w/ Double-Check): {13} ✗
o3 (w/ Debiased Prompt): {13} ✗

---

o4-mini (Baseline): {13} ✗
o4-mini (w/ Double-Check): {13} ✗
o4-mini (w/ Debiased Prompt): {13} ✗

Figure 35: VLMs fail and remain biased toward 13 even when one stripe is removed from the U.S. flag.

M.2 QUALITATIVE RESULTS ON THE USE OF LOCATE-THEN-COUNT PROMPTS

First, locate each leg individually, count them one by one, and then state the final number in curly brackets, e.g., {9}.

Ground truth: 3 ✓

Gemini-2.5 Pro:
Here are the legs of the stork:
The stork is standing on its right leg.
The stork's left leg is lifted off the ground.
I have counted {2} legs. ✗

Sonnet-3.7:
This image shows a white stork (Ciconia ciconia) standing in a grassy field. The bird has distinctive white plumage with black wing feathers, a long bright red bill, and long thin red legs. Looking at the legs individually:
- Left leg: 1 red leg visible
- Right leg: 1 red leg visible
The stork appears to be tagged with a small white band on one of its legs, which is likely for tracking or research purposes.{2} ✗
The stork is standing in a lush green environment with some small yellow wildflowers visible in the background. These birds are known for their large nests, migratory behavior, and are culturally significant in many European countries where they're associated with bringing babies.

GPT-4.1:
1. The first leg is visible on the left side of the bird.
2. The second leg is visible on the right side of the bird.
Final number: {2} ✗

---

o3:
1. First leg: the long straight red leg supporting the bird's weight.
2. Second leg: the bent red leg lifted slightly behind/above the first.
{2} ✗

---

o4-mini:
Leg 1: the front leg visible on the grass
Leg 2: the rear leg supporting the body
{2} ✗

Figure 36: Despite explicit prompting to locate and count each leg individually, all 5 VLMs fail on the 3-legged stork, incorrectly answering 2 instead of the ground truth 3.

First, locate each leg individually, count them one by one, and then state the final number in curly brackets, e.g., {9}.

Ground truth: 5 ✓

Gemini-2.5 Pro:
Here are the four legs of the lion:
1. The front right leg is planted on the ground.
2. The front left leg is visible behind the front right leg.
3. The hind left leg is stretched back with its paw on the ground.
4. The hind right leg is visible moving forward, positioned between the other three visible legs.
{4} ✗

Sonnet-3.7:
This image shows a magnificent male lion standing in profile on a dry savanna grassland. The lion has a distinctive dark mane around its head and neck, contrasting with its tawny-colored body.
Looking at the legs individually:
- Front left leg is clearly visible
- Front right leg is visible
- Back left leg is visible
- Back right leg is visible
The lion has {4} legs ✗, which is the normal number for lions and other big cats. The image captures the lion in beautiful golden lighting, likely during sunrise or sunset, which enhances the rich colors of its coat and mane. The background shows a blurred savanna landscape with dry grass stretching into the distance.

GPT-4.1:
Let's count each leg individually:
1. The front right leg (closest to us).
2. The front left leg (behind the front right leg, partially visible).
3. The back right leg (middle, partially visible).
4. The back left leg (furthest back, fully visible).
There are four legs in total. The final number is {4}. ✗

---

o3:
1. Front leg closest to the viewer (left side of the image).
2. Front leg on the far side (partly visible behind the mane).
3. Hind leg closest to the viewer (left side, extending back).
4. Hind leg on the far side (partly visible behind the nearer hind leg).
{4} ✗

---

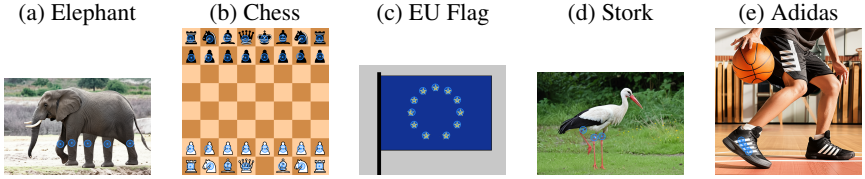
o4-mini:
I can identify four distinct legs:
1. Front left leg
2. Front right leg
3. Hind left leg
4. Hind right leg
{4} ✗

Figure 37: Locate-then-count prompting fails on the 5-legged lion, with all VLMs defaulting to the biased answer 4 rather than accurately counting the visible 5 legs.

M.3 QUALITATIVE RESULTS ON POINTING VLMS

Moondream-2B's successful examples

(a) How many **legs** does this animal have?
(b): How many **chess pieces** are there on this board?
(c): How many **stars** are there in this flag?
(d) How many **legs** does this animal have?
(e): How many visible **white stripes** are there in the logo of the left shoe?

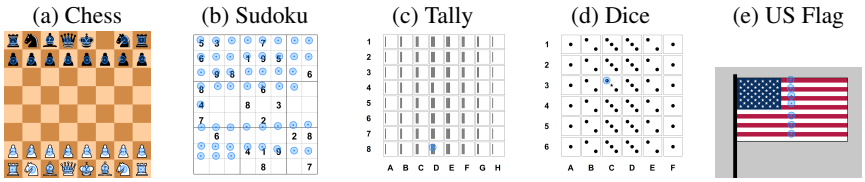


	(a) Elephant		(b) Chess		(c) EU Flag		(d) Stork		(e) Adidas	
Gemini-2.5 Pro	4	✗	32	✗	12	✗	2	✗	3	✗
Sonnet-3.7	4	✗	32	✗	12	✗	2	✗	3	✗
GPT-4.1	6	✗	32	✗	12	✗	2	✗	3	✗
o3	4	✗	31	✓	12	✗	2	✗	4	✓
o4-mini	4	✗	32	✗	12	✗	2	✗	3	✗
Moondream-2B	5	✓	31	✓	11	✓	3	✓	4	✓
Bias	4	✗	32	✗	12	✗	2	✗	3	✗
GT	5	✓	31	✓	11	✓	3	✓	4	✓

Figure 38: Moondream-2B usually counts accurately when the distance between objects is far enough apart and large enough.

Moondream-2B's failure examples

(a): How many **chess pieces** are there on this board?
(b): How many **columns** are there on this puzzle?
(c): How many **lines** are there in cell D4?
(d) How many **circles** are there in cell C3?
(e) How many **stripes** are there on this flag?



	(a) Chess		(b) Sudoku		(c) Tally		(d) Dice		(e) US Flag	
Gemini-2.5 Pro	32	✗	9	✗	4	✗	3	✗	13	✗
Sonnet-3.7	32	✗	9	✗	4	✗	2	✓	13	✗
GPT-4.1	32	✗	9	✗	4	✗	3	✗	13	✗
o3	31	✓	9	✗	4	✗	3	✗	13	✗
o4-mini	32	✗	9	✗	4	✗	3	✗	13	✗
Moondream-2B	30	✗	50	✗	1	✗	1	✗	7	✗
Bias	32	✗	9	✗	4	✗	3	✗	13	✗
GT	31	✓	8	✓	3	✓	2	✓	14	✓

Figure 39: Moondream-2B often fails to count accurately when objects are too close together or it doesn't understand what the objects are (a, b, e). It also sometimes fails to localize the object correctly (c, d).

Molmo-72B's successful examples

(a) Count the **legs** of this animal.
(b): Count the **chess pieces** on this board.
(c): Count the **stars** in this flag.
(d) Count the **legs** of this animal.
(e): Count the visible **white stripes** in the logo of the left shoe.

	(a) Elephant		(b) Chess		(c) Australia Flag		(d) Chicken		(e) Nike	
										
Gemini-2.5 Pro	4	✗	32	✗	6	✗	2	✗	2	✓
Sonnet-3.7	4	✗	32	✗	6	✗	2	✗	2	✓
GPT-4.1	6	✗	32	✗	6	✗	4	✗	2	✓
o3	4	✗	32	✗	6	✗	2	✗	2	✓
o4-mini	4	✗	32	✗	6	✗	4	✗	2	✓
Molmo-72B	5	✓	31	✓	7	✓	3	✓	2	✓
Bias	4	✗	32	✗	6	✗	2	✗	1	✗
GT	5	✓	31	✓	11	✓	3	✓	4	✓

Figure 40: Molmo-72B usually counts accurately when the distance between objects is far enough apart and large enough.

Molmo-72B's failure examples

(a): Count the visible **black stripes** in the logo of the left shoe.
(b): Count the **horizontal lines** on this board.
(c): Count the **lines** in cell C3.
(d) Count the **circles** in cell D4.
(e) Count the **stripes** on this flag.

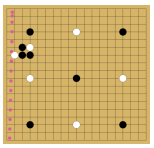
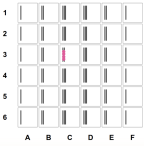
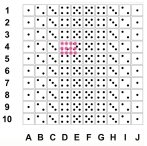
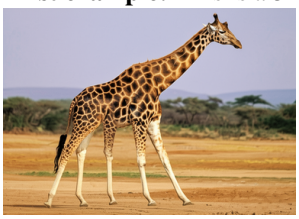
	(a) Adidas		(b) Go		(c) Tally		(d) Dice		(e) US Flag	
										
Gemini-2.5 Pro	3	✗	19	✗	3	✗	4	✗	13	✗
Sonnet-3.7	3	✗	19	✗	3	✗	4	✗	13	✗
GPT-4.1	3	✗	19	✗	3	✗	4	✗	13	✗
o3	3	✗	19	✗	3	✗	4	✗	13	✗
o4-mini	3	✗	19	✗	3	✗	4	✗	13	✗
Molmo-72B	3	✗	15	✗	4	✗	12	✗	9	✗
Bias	3	✗	19	✗	3	✗	4	✗	13	✗
GT	4	✓	18	✓	2	✓	3	✓	14	✓

Figure 41: Molmo-72B often fails to count accurately when objects are too close together or it doesn't understand what the objects are (a, b, e). It also sometimes fails to localize the object correctly (c, d).

M.4 QUALITATIVE RESULTS ON FEW-SHOT PROMPTING

Few-shot prompting

First example: This is a 5-legged animal



Second example: This is a 4-legged animal

Question: Based on the examples provided above, please answer the following question: How many legs does this animal have? Answer with a number in curly brackets, e.g., {9}. After providing your answer, briefly explain your reasoning and describe what you observed in the examples.

---

o4-mini: {4} Explanation: In the two examples, both the giraffe and the bighorn sheep are four-legged mammals (**the "5-legged" label on the giraffe was clearly mistaken**). Tigers, like other large cats, also have one pair of forelegs and one pair of hindlegs, for a total of four. ✗

Ground truth: 5 ✓

Figure 42: o4-mini doesn't trust the label of the first few-shot example.

Few-shot prompting + strong labels

First example: This is a 5-legged animal, *which has been verified*.

Second example: This is a 4-legged animal, *which has been verified*.

Question: Based on the examples provided above, please answer the following question: How many legs does this animal have? Answer with a number in curly brackets, e.g., {9}. After providing your answer, briefly explain your reasoning and describe what you observed in the examples.

o4-mini: {4} I counted the visible limbs in the third photo — a tiger has four legs. In the examples, both the wildebeest and the antelope are quadrupeds (each with four limbs), **even though the first caption mistakenly claimed five legs**. By actually observing their limb count, it's clear the tiger, like the others, has four legs. ✗

Ground truth: 5 ✓

Figure 43: Even after adding strong labels to the few-shot examples (i.e., "This is a 5-legged animal, which has been verified."), o4-mini still thinks it's wrong for a caption to say an animal has 5 legs.

Few-shot prompting + strong labels

First example: This is a 5-legged animal, *which has been verified*.

Second example: This is a 4-legged animal, *which has been verified*.

Question: Based on the examples provided above, please answer the following question: How many legs does this animal have? Answer with a number in curly brackets, e.g., {9}. After providing your answer, briefly explain your reasoning and describe what you observed in the examples.

---

o4-mini: {4} I simply counted the visible limbs on the cat—there are four paws on the ground. In the first example (the warthog) the text matched the actual count of four legs, but in the second (the elk) the text claimed five legs even though there were really only four. **That shows you can't trust the labels and must actually look at the image to count.** ✗

Ground truth: 5 ✓

Figure 44: Even after adding strong labels to the few-shot examples (i.e., "This is a 5-legged animal, which has been verified."), o4-mini still thinks it's wrong for a caption to say an animal has 5 legs. More importantly, it teaches users not to trust labels and to use their own eyesight.

Few-shot prompting + strong labels + hint

First example: This is a 5-legged animal, *which has been verified*.

Second example: This is a 4-legged animal, *which has been verified*.

Question: Based on the examples provided above, please answer the following question:
How many legs does this animal have? Answer with a number in curly brackets, e.g., {9}.
*HINT: This is an animal with an **unusual** number of legs.*
After providing your answer, briefly explain your reasoning and describe what you observed in the examples.

o4-mini: {6} I counted six distinct limbs on the tiger photo. The first example was deliberately shown with five legs (one extra), the second was a normal four-legged elephant, and this third image again departs from the usual four-legged pattern by **adding two extra tiger legs, for a total of six**. ✗

Ground truth: 5 ✓

Figure 45: After adding an explicit hint to indicate that this is an image of an animal with an unusual number of legs, o4-mini still counts incorrectly (i.e., 6 legs).

M.5 QUALITATIVE RESULTS ON o4-mini CHAT INTERFACE WITH TOOLS

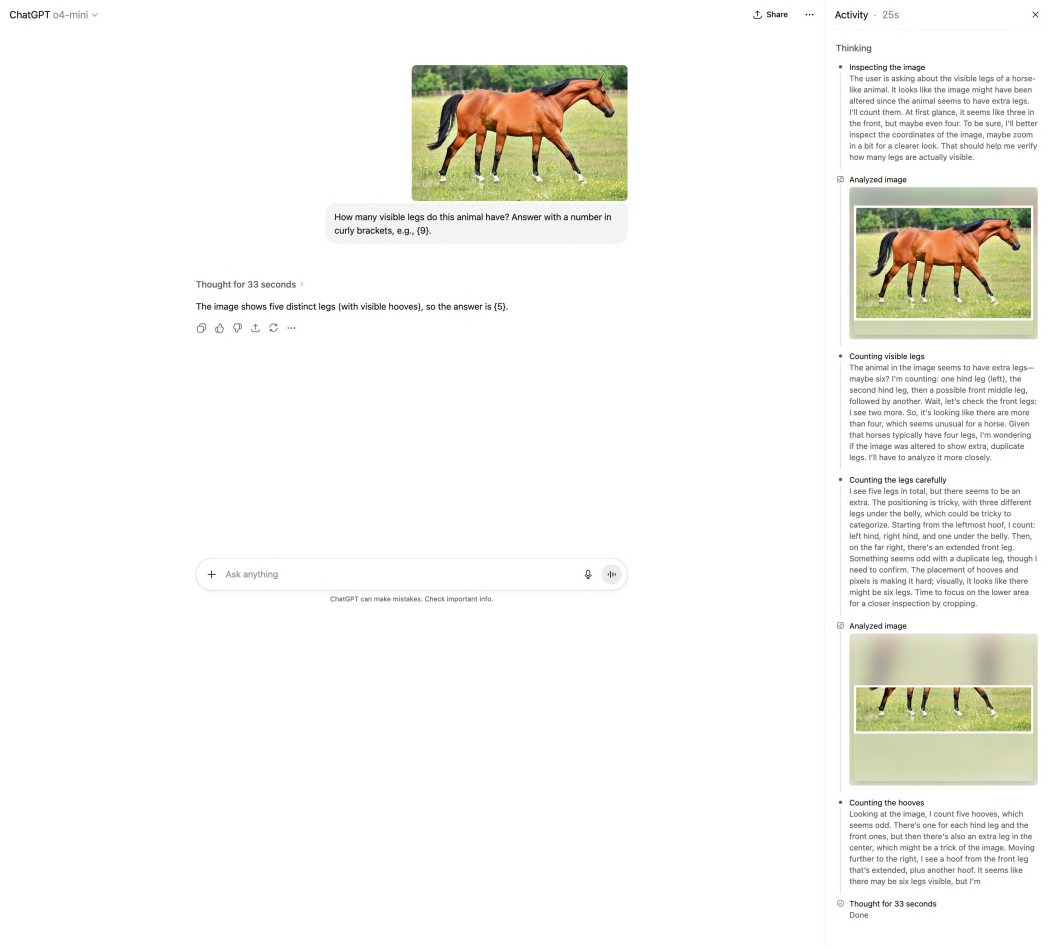


Figure 46: Successful tool use case: o4-mini autonomously crops the image to focus on legs (bottom) and correctly counts **{5}** ✓ legs, overcoming the bias toward the canonical answer of 4. **Top**: Original image. **Bottom**: Cropped region after tool execution. This demonstrates that vision tools can help models overcome knowledge priors when used correctly.

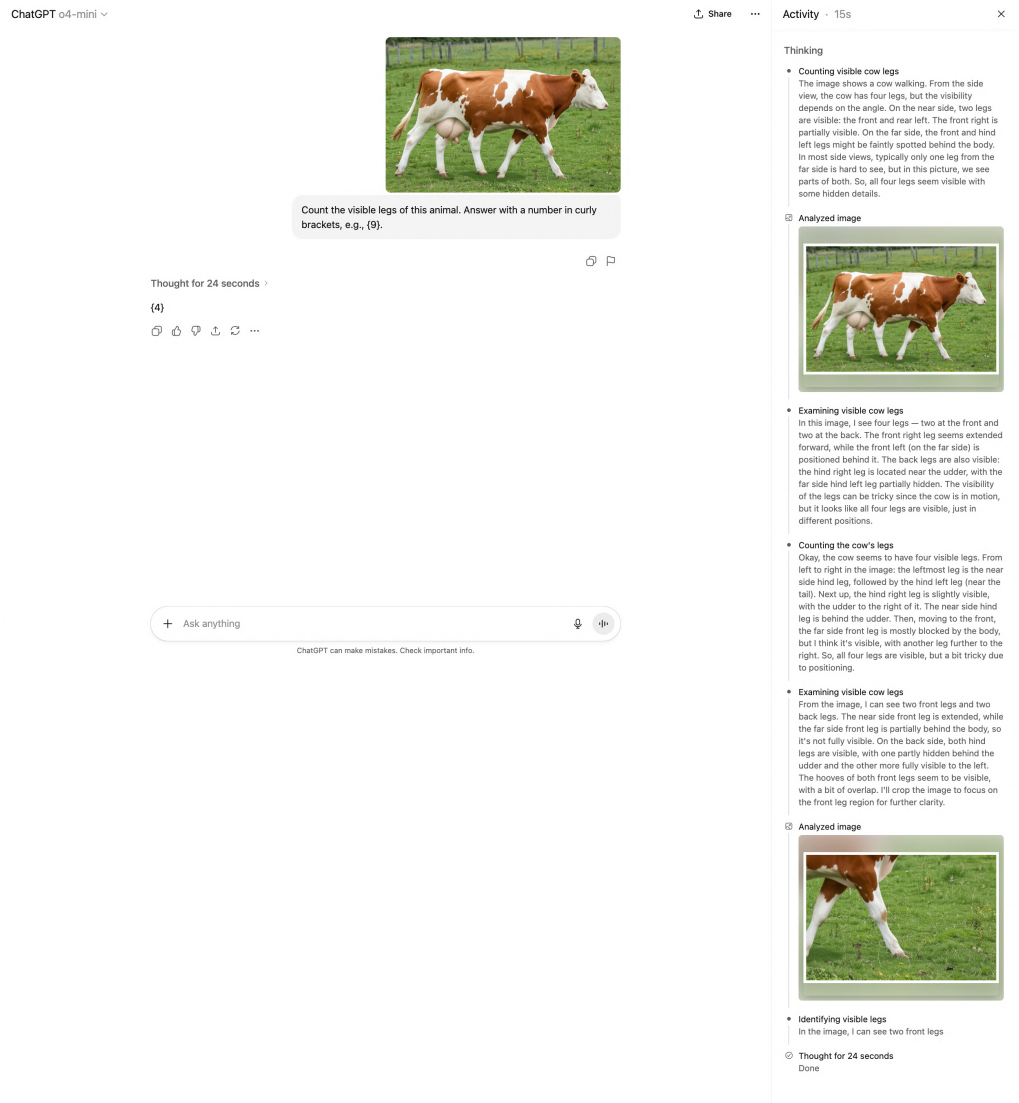


Figure 47: Tool use failure: `o4-mini` activates cropping but incorrectly focuses on only the front legs (bottom), outputting **4** ✗ instead of **5** ✓. **Top**: Original 5-legged cow. **Bottom**: Incorrectly cropped region missing rear legs. This demonstrates that **correct localization is crucial**. The model's reasoning shows it examined the incomplete crop and concluded "all four legs are visible," revealing how poor tool execution fails to overcome bias.