Response Wide Shut? Surprising Observations in Basic Vision Language Model Capabilities

Shivam Chandhok[1,2] **Wan-Cyuan Fan**[1,2] **Vered Shwartz**[1,2,4]
Vineeth N Balasubramanian[3,5] **Leonid Sigal**[1,2,4]

[1]University of British Columbia [2]Vector Institute for AI [3]IIT Hyderabad
[4]CIFAR AI Chair [5]Microsoft Research India

{chshivam, wancyuan, vshwartz, lsigal}@cs.ubc.ca vineeth.nb@microsoft.com

Abstract

Vision-language Models (VLMs) have emerged as general-purpose tools for addressing a variety of complex computer vision problems. Such models have been shown to be highly capable, but, at the same time, lacking some basic visual understanding skills. In this paper, we set out to understand the limitations of SoTA VLMs on fundamental visual tasks by constructing a series of tests that probe which components of design, specifically, may be lacking. Importantly, we go significantly beyond the current benchmarks, which simply measure the final performance of VLM response, by also comparing and contrasting it to the performance of probes trained directly on features obtained from the visual encoder, intermediate vision-language projection and LLM-decoder output. In doing so, we uncover shortcomings in VLMs and make a number of important observations about their capabilities, robustness and how they process visual information. We hope our insights will guide progress in further improving VLMs.

1 Introduction

Recently, vision-language models (VLMs) have emerged as general-purpose tools that can address many complex language and vision tasks such as comprehending charts and interpreting humor in images and videos (Li et al., 2023b; Liu et al., 2023; Li et al., 2024). However, there is also a growing body of evidence that VLMs lack basic capabilities that are considered necessary to solve simple high-level tasks, such as the ability to understand simple negations (Alhamoud et al., 2025) or recognize and count objects (Kim and Ji, 2024; Peng et al., 2024; Zhang et al., 2024; Paiss et al., 2023). These observations suggest that the mechanisms for solving complex tasks in these models may be different from those in humans, relying more on large-scale matching and memory recall, as opposed to functional and step-by-step reasoning.

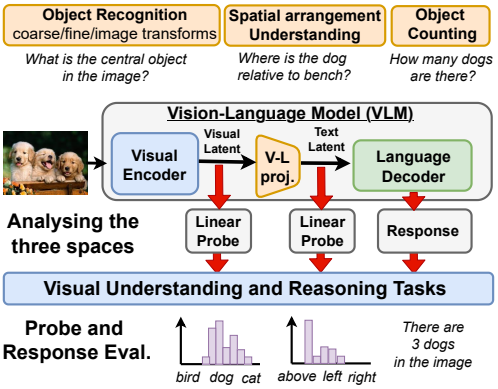

Figure 1: **Overview of our VLM analysis.** Going beyond existing efforts that analyze VLMs as a whole, we study performance of VLMs in terms of intermediate spaces that represent knowledge as it is processed through the VLM network. Specifically, we consider three spaces in VLMs: *visual*, *VL projection* and *response* space; to understand what aspects of visual information are captured (not captured) and where.

Inspired by these contradictory observations, we propose a systematic and nuanced analysis of VLMs' performance, focusing on core vision capabilities that we posit are required for high-level visual reasoning tasks: (i) the ability to recognize objects (coarse and fine-grained classification), (ii) delineate instances of a given object type (counting) and (iii) understand their spatial arrangement.

Previous work that pointed out deficiencies of VLMs on the aforementioned tasks focused only on the final VLM response (the default way to use VLMs) (Peng et al., 2024; Paiss et al., 2023; Kim and Ji, 2024; Kamath et al., 2023; Zhang et al., 2024). While this allows a holistic measure of performance, it does not provide any insights on which components in VLM design maybe lacking or could be improved. In this paper, we propose to analyze VLMs' performance *in terms of intermediate spaces that represent information as it is processed through the VLM network* (Figure 1). Specifically, VLMs typically consist of a visual encoder, text decoder, and (optional) visual-language (VL) projection or alignment mechanism. Therefore, by analyzing the output of these different mod-

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 25530–25545

Figure 2: **Qualitative results supporting the findings of our analysis.** We show prediction (correct vs incorrect) for three spaces i.e *visual, VL projection* and *response*. We notice correct predictions in intermediate spaces and incorrect predictions in response space for object recognition and counting task. Furthermore, we notice a reversal in trend for spatial understanding task, where the response space has more correct predictions compared to intermediate spaces.

ules in their ability to perform visual tasks, we can better understand which capabilities may be missing and where. Knowing which of the components is at fault would give important insights into how the performance can be improved and provide a more targetted strategy for enhancing VLMs.

Our observations in this work point to the fact that contrary to current understanding (Tong et al., 2024), vision encoders are quite proficient at visual tasks and that the VL projection preserves most (if not all) information. We instead find a significant drop in performance in the VLM response layer—the output of the language decoder, especially in tasks like fine-grained recognition and object counting. Figure 2 shows samples of our qualitative results that examine the predictions after each module: visual, VL proj. and response. We see – as stated above – that the final response layer seems to fall short in translating the strong performance of visual and VL projection modules across the recognition and object counting tasks. In spatial understanding, however, our results reaffirm the results in (Tong et al., 2024) that the visual and VL proj. space shows weak performance, supporting the need for better visual encoders.

Beyond these, our studies reveal other interesting observations, w.r.t how background, shape and visual prompting information is processed through spaces within a VLM. Overall, through our insights, our work seeks to identify the key gaps in module performance in VLMs vis-a-vis well-known vision tasks and encourage future work to target efforts on improving these gaps in the next generation of vision-language models.

2 Related Work

Vision Language Models (VLMs). Existing efforts on VLMs can be broadly categorized into three groups based on their training objectives and architectural design: (1) *Contrastive multi-encoder models* such as CLIP (Radford et al., 2021) and ALBEF (Li et al., 2021) consist of a separate encoder for each modality (*i.e.*, image and text) and use contrastive alignment between image and text inputs for training. (2) *Encoder-decoder generative models* such as BLIP-2 (Li et al., 2023a) employ an encoder-decoder architecture to project images to a low-dimensional representation and then use the representation as context to the language decoder to generate language descriptions or captions with generative language modeling loss. (3) *Instruction fine-tuned models* such as LLaVA-1.5 (Liu et al., 2023), InstructBLIP (Dai et al., 2023) and LLaVA-NEXT (Liu et al., 2024) are additionally fine-tuned to follow human instructions and intent while answering visual questions. Usually, these models are fine-tuned with some form or variant of RLHF

(Lambert et al., 2022), or direct preference optimization (Rafailov et al., 2023), which allows them to understand human question intent and answer accordingly. In this paper, we analyze VLMs from all three model groups.

Analyzing VLM Capabilities. Recent interests in VLMs have motivated research that aims to understand their visual capabilities. Han et al. (2024) analyzed how VLMs generalize to distribution shifts, and Udandarao et al. (2023) further showed a strong correlation between the ability to recognize the type of perturbation and the robustness to it. Recently, Peng et al. (2024) evaluated visual-linguistic concepts such as object size, position, existence and count. Similarly, other efforts (Kamath et al., 2023; Paiss et al., 2023; Yuksekgonul et al., 2022; Thrush et al., 2022) pointed out deficiencies of VLMs on recognition, spatial, counting etc. However, they only evaluate the final response from the VLMs whereas we test the performance of each component in the VLM individually through our novel three sub-space analysis design to pinpoint where the problem is coming from. Related to our work, Zhang et al. (2024) explores image classification in VLMs comparing the performance of LLaVA with contrastive VLMs like CLIP. They find that LLM decoder output (of inference and probe) from LLaVA (Liu et al., 2023) performs inferior to the performance of CLIP.

In contrast, we go a step further and present an in-depth analysis of VLM's in three intermediate feature spaces that represent the flow of information through the networks allowing us to pinpoint where the deficiencies are coming from. Furthermore, our comprehensive analysis covers a range of diverse tasks in addition to classification, such as counting, spatial orientation, model robustness, prompting and background transformations. Additionally, a recent effort, EWS (Tong et al., 2024), posit that deficiencies of the visual encoder affect the VLM's overall performance. Although this is true for spatial tasks, we find that for most visual tasks visual encoders are proficient/capture necessary information, however, this information is lost in the language decoder and does not translate to an accurate final VLM response.

3 Experimental Setup

Motivated by the modular design of VLMs, where individual components are often pre-trained separately and later put together for end-to-end fine-tuning (Liu et al., 2023), we propose to investigate VLMs by looking at their performance in terms of intermediate feature spaces that represent information as it is processed by the network (Fig. 1).

Specifically, VLMs typically comprise of a *visual encoder*, *vision-language projection* module and *language decoder* (see Figure 1). This gives rise to 3 different sub-spaces within a VLM: (1) *visual latent* (output of visual encoder) space; (2) *vision-language shared latent* (output of vision-language projection) space, and (3) *language response* space (output of language decoder). We conjuncture that these spaces might capture different aspects of visual information which cumulatively help a VLM understand visual content. To this end, we probe these different spaces and analyze their visual capabilities to get a nuanced understanding of what aspects of visual information are captured within VLMs and where. Our design can help pinpoint where the visual knowledge may be lacking or lost, informing future work on designing better VLM architectures and training strategies.

3.1 Choice of VLMs

To comprehensively analyze the visual capabilities of a diverse set of models, we choose representative models from each category (described in Section 2): CLIP (Radford et al., 2021), ALBEF (Li et al., 2021) as contrastive multi-encoder models; CoCa (Yu et al., 2022) and BLIP-2 (Li et al., 2023a) as encoder-decoder generative models; and InstructBLIP (Dai et al., 2023), LLaVA v.1.5 (Liu et al., 2023), and LLaVa-NEXT (Liu et al., 2024) as instruction fine-tuned models. Our choice of these models is also motivated by recent work on VLM analysis (Tong et al., 2024; Han et al., 2024; Udandarao et al., 2023). Given the popularity of these models, they are often used as a starting point for other more sophisticated VLMs. Hence, analyzing their shortcomings can have a broader impact.[1] Finally, our analysis requires access to the intermediate feature representations, limiting us to open-source models; preventing us from evaluating industrial models such as GPT-4V and Gemini.

3.2 Choice of Visual Tasks and Benchmarks

Following prior work, we conjecture that in order to understand the visual content of an image and ef-

[1]At the time of writing, LLaVA-NEXT is widely considered among SoTA (Liu et al., 2024) open-source models, and the LLaVA-v1.5 model is widely used for downstream applications and fine-tuning VLMs for new tasks.

fectively reason about it, a model must have fundamental capabilities such as the ability to recognize objects (*coarse and fine-grained categorization*), group similar object instances together (*counting*) and understand the *spatial relations* between objects (Cho et al., 2023). Even though these capabilities are not exhaustive, understanding these dimensions is pivotal for visual reasoning (*e.g.*, VQA) as well as building higher-level representations and abstractions like scene graphs (Xu et al., 2017).

We leverage existing benchmarks for these tasks: PaintSkills (Cho et al., 2023) and Pascal VOC (Everingham et al., 2010) for coarse object recognition, Stanford Dogs (Khosla et al., 2011) and CUB (Wah et al., 2011) for fine-grained recognition, and PaintSkills (Cho et al., 2023) for counting (1–4) and spatial reasoning (above, below, left, right).[2]. We further elaborate on exact details about dataset splits and classes in Appendix A.7

3.3 Task Design

We formulate the aforementioned tasks as classification over a set of discrete choices (*e.g.*, for *classification* – a set of coarse/fine object classes; for *counting* – a set of counts; and *spatial orientations* – a set of spatial arrangements, such as above, below, left-of, right-of). We convert these classification tasks to a question form to test visual understanding capabilities of VLM models (as used by previous work: Kim and Ji, 2024; Han et al., 2024) where the VLM is prompted with a VQA query such as "*What is the central object in the image? Choices - dog, cat, ...?*". We further elaborate on task prompts and details of probing in Appendix A.2.

3.4 Evaluation Design

Visual and VL Projection Space. To evaluate the visual and VL projection spaces, we probe the frozen feature representations by training task-specific linear probes. Following previous work (Radford et al., 2021; Caron et al., 2021; El Banani et al., 2024), we use the standard linear probing strategy of a single layer (MLP) logistic regression model. Specifically, we train a probe on an average pooled features from output of vision encoder and VL proj. for each image. The probes are trained on training split and evaluated on val/test splits.

Response Space. We evaluate the response space in two ways: (1) for fair comparison with the visual and VL projection, we similarly train a linear probe on the output token representation from the language decoder (`Probe`). Specifically, the output of the language decoder for a given image is of the form $T \times F$, where T is the number of tokens and F is the feature dimension of token embeddings, which we average pool to produce F-dimensional feature for each image. (2) we directly evaluate the textual response of the VLM to a VQA query for a given task such as "*What is the central object in the image? Choices - dog, cat, ...?*" (`Text`). Note that this is the default way to evaluate VLMs.[3] (See Appendix A.2 for of task prompts and evaluation)

Control Task for Probes. To verify that representation encodes properties useful for the task rather than the linear probe learning the task from the training data, we include a control task in which we randomly shuffle the labels (Zhang and Bowman, 2018; Hewitt and Liang, 2019). Good performance on the target task combined with bad performance on the control tasks validates our linear probing results can serve as proxy for information present (See Appendix A.3 for the control task results).

4 Analyzing the Skills of VLMs

4.1 Object Recognition

(Coarse-grained) object recognition. Table 1 presents the results of VLMs on coarse-grained object recognition. The visual and VL projection spaces for all VLMs perform well, resulting in accuracy above 95%. However, there is some dip in performance in the response space across models (atleast 5% on average). Even though all models show similar high performance in visual and VL proj. space, BLIP family of models perform worse in the response space compared to other models. We attribute this gap to the fact that BLIP-2 wasn't instruction tuned which may prevent it from understanding the instructions in the query even when it otherwise has access to the required information in the visual and projection space. Evidence for this can be seen by comparing BLIP-2 to its instruction-tuned counterpart – InstructBLIP which results in a substantial performance increase in response space. However, InstructBLIP still does not reach the performance of LLaVA-1.5 and LLaVA-NEXT on Pascal VOC, which we attribute to stronger and better instruction-tuning for the latter models. The better performance of the linear prob (`probe`) compared

[2]PaintSkills is a diagnostic dataset designed to measure fundamental skills in foundational models.

[3]CLIP, CoCa, and ALBEF don't have a language decoder, hence we only report the `Probe` for these models.

VLM Method		PaintSkills				Pascal VOC				**Average**			
		Visual probe	VL Proj probe	Response probe	Response text	Visual probe	VL Proj probe	Response probe	Response text	Visual probe	VL Proj probe	Response probe	Response text
CLIP (Radford et al., 2021)		99.2	-	-	96.5	98.5	-	-	95.6	98.9	-	-	96.0
CoCa (Yu et al., 2022)		99.9	-	-	94.0	97.6	-	-	96.6	98.8	-	-	95.3
ALBEF (Li et al., 2021)		99.7	-	-	75.3	97.3	-	-	86.0	98.5	-	-	80.7
BLIP-2 (Li et al., 2023a)	7B	99.2	99.2	87.0	59.0	98.0	99.0	57.3	38.0	98.6	99.1	72.2	48.5
InstructBLIP (Dai et al., 2023)	7B	99.9	99.2	98.5	98.0	98.0	98.6	95.0	70.0	99.0	98.9	96.8	84.0
LLaVA-1.5 (Liu et al., 2023)	7B	99.2	99.2	94.4	97.0	96.8	95.8	91.1	90.2	98.0	97.5	92.8	93.6
LLaVA-NEXT (Liu et al., 2024)	7B	99.8	99.8	94.2	97.7	97.3	96.6	94.1	94.1	98.6	98.2	94.2	95.9

Table 1: **Course-grained Recognition.** Visual and VL proj. spaces for all VLMs perform well (accuracy above 95%). However, there is a small dip in performance in the response space across models (atleast 5% on average).

VLM Method		Stanford Dogs				CUB				**Average**			
		Visual probe	VL Proj probe	Response probe	Response text	Visual probe	VL Proj probe	Response probe	Response text	Visual probe	VL Proj probe	Response probe	Response text
CLIP (Radford et al., 2021)		94.0	-	-	84.0	95.0	-	-	71.0	94.5	-	-	77.5
CoCa (Yu et al., 2022)		94.0	-	-	87.0	94.0	-	-	74.5	94.0	-	-	80.8
ALBEF (Li et al., 2021)		83.0	-	-	39.0	82.5	-	-	16.7	82.8	-	-	27.9
BLIP-2 (Li et al., 2023a)	7B	93.6	92.0	58.1	23.5	92.0	93.2	20.5	10.0	92.8	92.6	39.3	16.8
InstructBLIP (Dai et al., 2023)	7B	93.6	92.6	28.5	12.0	92.2	94.0	60.7	13.0	92.9	93.3	44.6	12.5
LLaVA-1.5 (Liu et al., 2023)	7B	91.9	88.7	25.9	19.9	92.2	87.3	34.1	34.6	92.1	88.0	30.0	27.3
LLaVA-NEXT (Liu et al., 2024)	7B	90.4	87.2	37.5	31.0	90.0	85.0	22.5	18.0	90.2	86.1	30.0	24.5

Table 2: **Fine-grained Recognition.** There is significant drop in performance in the response space for fine- vs. course-grained recognition. In the visual and VL projection feature spaces the drop also exists, but is significantly less, comparatively speaking.

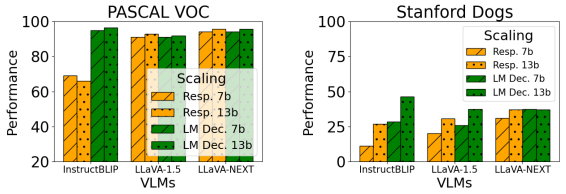

Figure 3: **Effect of Scaling LLM Decoder on Response space.** Refer to Appendix A.1 for elaborate results

to (text) space on InstructBLIP also supports this hypothesis; *i.e.*, it is able to capture the answer, but not verbalize it in (text).

(Fine-grained) object recognition. Table 2 presents the results of VLMs on fine-grained object recognition. The decrease in performance in the response space compared to the visual and VL projection spaces is drastic (atleast 45 % drop), across both datasets. Our experiments point to problems in the response space by showing that the visual and VL projection probes perform very well, with most models achieving an accuracy above 90% across datasets. However, the knowledge does not translate to the response space. These results shed light on the findings in Kim and Ji (2024) that VLMs are overall less capable in fine-grained recognition, and contradict the conclusion of Tong et al. (2024) that the deficiency is due to the visual encoder. Further, we note that Kim and Ji (2024) show that text-only language decoders are themselves good at fine-grained object classification. Hence, we conjecture that the drastic dip in performance is not due to lack of knowledge in the language decoder; rather it is due to ineffective joint fine-tuning

VLM Method		PASCAL VOC			
		Visual probe	VL Proj probe	Response probe	Response text
LLaVA-NEXT (Liu et al., 2024)	7B	97.3	96.6	94.1	94.1
LLaVA-NEXT (Liu et al., 2024)	13B	97.3	96.2	95.7	95.8
LLaVA-NEXT (Liu et al., 2024)	34B	97.4	96.5	97.0	97.0
VLM Method		Stanford Dogs			
		Visual probe	VL Proj probe	Response probe	Response text
LLaVA-NEXT (Liu et al., 2024)	7B	90.4	87.2	37.5	31.0
LLaVA-NEXT (Liu et al., 2024)	13B	90.5	86.5	37.0	36.9
LLaVA-NEXT (Liu et al., 2024)	34B	91.0	86.9	51.3	49.4

Table 3: **Effect of Scaling LLM Decoder.** Results are shown on three spaces for LLaVA-NEXT on coarse-grained (top) and fine-grained (bottom) recognition task.

of the proj. layer and language decoder which is responsible for aligning the vision and language modalities.

Object recognition discussion and implications. While the encoding of the fine-grained classes could be seen as being preserved in the visual and intermediate VL projection space, it is clear that representations in that space do not align well with those in the language decoder. We investigate this further and find that the LLaVA 665k fine-tuning data has only 0.17% of samples about fine-grained dog breeds. Given our findings, we posit that the data used for joint fine-tuning of VLMs' projection with language decoder did not contain enough samples (was not representative) of fine-grained classes. Hence, incorporating fine-grained samples and improving joint fine-tuning of the VL proj. and language decoder would yield improvements in fine-grained recognition.

VLM Method		PaintSkills			
		Visual probe	VL Proj probe	Response probe	Response text
CLIP (Radford et al., 2021)		93.5	-	-	49.0
CoCa (Yu et al., 2022)		91.7	-	-	77.0
ALBEF (Li et al., 2021)		78.1	-	-	40.0
BLIP-2 (Li et al., 2023a)	7B	96.6	95.3	26.0	25.0
InstructBLIP (Dai et al., 2023)	7B	96.6	95.6	82.0	82.0
LLaVA-1.5 (Liu et al., 2023)	7B	93.0	94.0	81.0	81.0
	13B	93.1	94.0	73.0	73.8
LLaVA-NEXT (Liu et al., 2024)	7B	94.4	95.7	81.2	81.2

Table 4: **Object Counting on PaintSkills.** See text for detailed analysis and discussion.

VLM Method		PaintSkills			
		Visual probe	VL Proj probe	Response probe	Response text
CLIP (Radford et al., 2021)		47.5	-	-	31.0
CoCa (Yu et al., 2022)		48.0	-	-	27.3
ALBEF (Li et al., 2021)		47.0	-	-	28.0
BLIP-2 (Li et al., 2023a)	7B	50.4	50.0	27.7	25.0
InstructBLIP (Dai et al., 2023)	7B	50.4	57.0	27.7	26.8
LLaVA-1.5 (Liu et al., 2023)	7B	50.2	51.0	61.0	63.0
	13B	50.5	49.0	72.6	74.0
LLaVA-NEXT (Liu et al., 2024)	7B	49.8	49.8	37.6	37.6

Table 5: **Spatial Understanding on PaintSkills.** See text for detailed analysis and discussion.

Impact of model size. An important question to ask is if the drop in performance in the response space is due to the size of the language decoder. Figure 3, Table 3 and AppendixA.1 show the effect of scaling the language decoder on accuracy. We observe that a bigger LM decoder does generally improve the performance of response space to some extent, including improvements in the VL proj. space. However, the best response accuracy still remains atleast 35% lower than the corresponding visual and VL proj. space performance for fine-grained object recognition.

4.2 Counting and Spatial Understanding

Next, we consider object counting and spatial arrangement understanding more difficult than object recognition, which is a prerequisite for both. Following Cho et al. (2023), we use the PaintSkills diagnostic evaluation dataset for both tasks, as it allows us to control object placement and arrangement. Specifically, we use images with 1–4 instances for the counting and 4 orientations (*left*, *right*, *above*, *below*) for spatial arrangement task. The chance performance on both tasks is 25%.

Object counting. The results reported in Table 4 illustrate trends similar to those observed for fine-grained recognition, albeit less pronounced. Mainly, the performance of the visual and VL projection probes is high (>90%), with the *VL projection* representations marginally trailing *visual*, while performance in the response space is considerably poorer (a drop of at least 14%).

Dataset/Task	Performance		
	LLaVA-1.5 response(text)	(blind) Vicuna LLM response(text)	Chance
PASCAL (Coarse-Cls.)	90.2	8.7	6.6
CUB (Fine-Cls.)	34.6	7.7	6.6
Dogs (Fine-Cls.)	19.9	11.6	6.6
PaintSkills (Spatial)	63.0	25.6	25.0

Table 6: **Impact of language priors.** We compare performance of LLaVA-1.5 (VLM) with (blind) LLM to isolate role of language priors

Spatial understanding. Table 5 shows the trend observed thus far reverses on spatial understanding: the performance of *visual* latent representation is the worst; it increases a little for the *VL projection* space (for InstructBLIP and LLaVA-1.5), and increases further in the response space (for LLaVA-1.5). We hypothesize that the superior performance of the response space compared to visual space is due to the small number of samples pertaining to spatial arrangement in the training dataset for the visual encoder. LAION-2B used by OpenCLIP contains only 0.2% of spatial samples (Kamath et al., 2023). In contrast, to ~35% in LLaVA-665k used by LLaVA-1.5 to fine-tune response space. Additionally, lower performance of LLaVA-NEXT response compared to LLaVA-1.5, could be due to relative drop of relevant data from 35% in LLaVA-665k to ~26% in data used by LLaVA-NEXT.

Based on these results, we make a few important deductions. First, the spatial understanding capabilities of VLMs are clearly inferior, with the visual encoder (*i.e.*, CLIP) being responsible for the loss of information. Second, it appears that for some VLMs (*e.g.*, LLaVA-1.5) the response space makes up for some loss of performance thanks to the relevance of training data. These findings are in line with previous work showing that the performance of VLM on spatial understanding and counting (Kamath et al., 2023; Paiss et al., 2023) highly depend on the number of samples which are representative of these tasks in fine-tuning data.

Impact of language priors. Prior work has shown that language priors can influence the responses generated by VLMs (Goyal et al., 2017; Lin et al., 2024; Wu et al., 2025). For example, VLMs may learn from their language priors that the most likely spatial arrangement of a man and a chair is that the man is on the chair. These models are often able to solve VQA tasks "blindly" – *i.e.*, without an image

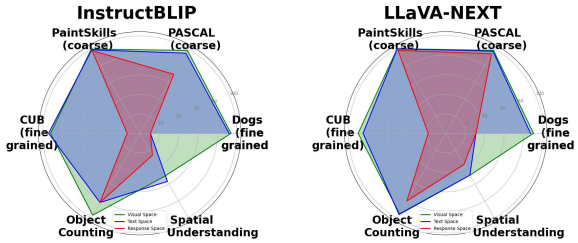

Figure 4: **Consolidation of performance** for instruction-tuned models from the BLIP family and the LLaVA family. Legend: Visual space, Text space, Response space.

encoder (Lin et al., 2024), or when the objects of interest are masked (Wu et al., 2025) – based on language priors. To reduce such confounding effects, our setup and datasets are carefully selected to minimize the influence of language priors. In particular, datasets like PaintSkills are synthetically constructed with uniformly distributed object placements, avoiding spurious correlations between language and visual content. Finally, we also present results from a "blind" language-only LLM baseline, further isolating the role of language priors in VLM performance. Specifically, we compare the text-only responses from instruction-tuned blind LLM (Vicuna) with those of LLaVA-1.5, enabling a clearer assessment of how much the VLM's performance depends on true visual reasoning versus language biases (Table 6). The low accuracy of the (blind) Vicuna LLM indicates the contribution of the language prior is relatively insignificant. These findings are consistent with prior work showing that language priors in generative VLMs – similar to those used in our setup – are generally not strong enough to have substantial positive impact on performance of visual tasks (Wu et al., 2025).

4.3 Summary of Insights on VLM Skills

The summary of all experiments in this section is compactly illustrated for two of the models (InstructBLIP and LLaVA-NEXT) in Figure 4.

① VLMs, overall, are much less capable of recognizing fine-grained categories which is largely attributable to response from language decoder. Knowledge from visual and VL proj. spaces does not translate to the VLM final response due to ineffective joint fine-tuning stage of the proj. and language decoder.

② Scaling the language decoder does not solve the aforementioned problem and the drop in performance in the response space remains significant.

③ The substantial drop in response space performance on counting is attributable to the limitations of the language decoder; similar to observation 1 above.

④ We observe a significant lack of ability of vision encoders to capture arrangement information, which results in a reduced performance overall for spatial understanding tasks.

VLM Method	Data	Visual probe	VL Proj probe	Response probe	Response text
BLIP-2	Orig	98.0	99.0	57.6	38.0
	Corrupt	95.2 -2.8	95.4 -3.6	57.5 -0.1	38.2 +0.2
InstructBLIP	Orig	98.0	98.6	95.0	70.0
	Corrupt	95.3 -2.7	95.9 -2.7	90.5 -4.5	69.5 -0.5
LLaVA-1.5	Orig	96.8	95.8	91.0	90.2
	Corrupt	90.8 -6.0	89.3 -6.5	87.7 -3.3	87.4 -2.8
LLaVA-NEXT	Orig	97.3	96.6	94.1	94.1
	Corrupt	89.7 -7.6	88.2 -8.4	87.3 -6.8	87.3 -6.8

Table 7: **Robustness to Visual Corruptions.** Robustness experiment on the Pascal VOC dataset.

5 Analyzing the Robustness of VLMs

Here we explore another dimension of visual understanding in VLMs: robustness to (1) visual corruptions (§5.1), and (2) background changes (§5.2).

5.1 Visual Corruptions

VLMs are fairly robust to many common corruptions, presumably due to the prevalence of these effects in the large-scale training data (Udandarao et al., 2023; Han et al., 2024). We thus focus on a diverse set of corruptions by which they are more affected. Specifically, we include noise (Gaussian, uniform, shot, and impulse), blurs (motion, defocus) and weather changes (snow) following Michaelis et al. (2019). We use methodology in Hendrycks and Dietterich (2019) and Michaelis et al. (2019), and add corruptions to Pascal VOC.

Robustness Across Spaces. Table 7 shows results for robustness across spaces. Here we report the average performance across corruptions (Avg. Corrupt), the original performance without corruptions (Orig) and their difference (in red). Detailed per-corruption breakdown is in Appendix A.4.

We observe that despite its deficiencies (see §4), the response space is the most robust among the three spaces we consider. Interestingly, in line with our findings that visual information does not transfer well to the response space, we see that the effects of the corruptions did not translate to the response space as well, making it *appear* as the most robust space. Furthermore, we observe that the intermediate VL projection latent space is less robust than the visual and response space, especially for LLaVA-1.5 and LLaVA-NEXT—pointing to defi-

Transformation	LLaVA-NEXT (Liu et al., 2024)				LLaVA-1.5 (Liu et al., 2023)				InstructBLIP (Dai et al., 2023)			
	Visual probe	VL Proj probe	Response probe	Response text	Visual probe	VL Proj probe	Response probe	Response text	Visual probe	VL Proj probe	Response probe	Response text
(1) Original	77.5	74.5	67.9	67.5	76.9	74.9	68.7	68.1	77.5	78.5	62.7	60.0
(2) Black BG	88.3$_{+10.8}$	87.0$_{+12.5}$	73.2$_{+5.3}$	72.5$_{+5.0}$	87.6$_{+10.7}$	87.1$_{+12.2}$	73.6$_{+4.9}$	72.5$_{+4.4}$	90.5$_{+13.0}$	90.7$_{+12.2}$	71.4$_{+8.7}$	61.0$_{+1.0}$
(3) White BG	88.2$_{+10.7}$	86.7$_{+12.2}$	73.0$_{+5.1}$	72.3$_{+4.8}$	88.7$_{+11.8}$	87.3$_{+12.4}$	74.2$_{+5.5}$	73.0$_{+4.9}$	90.4$_{+12.9}$	89.2$_{+10.7}$	70.6$_{+7.9}$	56.1$_{-3.9}$
(4) Silhouette + Orig BG	63.7$_{-13.8}$	59.7$_{-14.8}$	44.4$_{-23.5}$	41.1$_{-26.4}$	60.0$_{-16.9}$	57.8$_{-17.1}$	45.7$_{-23.0}$	42.1$_{-26.0}$	69.3$_{-8.2}$	69.4$_{-9.1}$	39.8$_{-22.9}$	41.7$_{-18.3}$
(5) Silhouette + White BG	47.9$_{-29.6}$	44.7$_{-29.8}$	22.1$_{-45.8}$	20.3$_{-47.2}$	44.3$_{-32.6}$	42.8$_{-32.1}$	23.3$_{-45.4}$	21.4$_{-46.7}$	51.2$_{-26.3}$	46.9$_{-31.6}$	24.5$_{-38.2}$	19.4$_{-40.6}$
(6) Blur Reverse	87.8$_{+10.3}$	87.5$_{+13.0}$	72.7$_{+4.8}$	74.2$_{+6.7}$	86.9$_{+10.0}$	85.3$_{+10.4}$	75.9$_{+7.2}$	75.0$_{+6.9}$	90.9$_{+13.4}$	92.0$_{+13.5}$	75.8$_{+13.1}$	64.2$_{+4.2}$
(7) Red Circle + Orig BG	81.8$_{+4.3}$	78.1$_{+3.6}$	71.1$_{+3.2}$	70.7$_{+3.2}$	80.3$_{+3.4}$	77.3$_{+2.4}$	72.9$_{+4.2}$	72.1$_{+4.0}$	82.4$_{+4.9}$	85.0$_{+6.5}$	69.5$_{+6.8}$	64.8$_{+4.8}$
(8) Red Circle + White BG	86.5$_{+9.0}$	85.5$_{+11.0}$	72.2$_{+4.3}$	71.5$_{+4.0}$	86.5$_{+9.6}$	85.5$_{+10.6}$	73.0$_{+4.3}$	72.0$_{+3.9}$	89.0$_{+11.5}$	88.2$_{+9.7}$	71.5$_{+8.8}$	62.1$_{+2.1}$
(10) Edge	62.0$_{-15.5}$	60.1$_{-14.4}$	53.1$_{-14.8}$	52.4$_{-15.1}$	61.3$_{-15.6}$	58.8$_{-16.1}$	54.2$_{-14.5}$	53.2$_{-14.9}$	70.0$_{-7.5}$	68.4$_{-10.1}$	53.5$_{-9.2}$	50.1$_{-9.9}$
(11) Patch Shuffle	73.5$_{-4.0}$	70.1$_{-4.4}$	65.1$_{-2.8}$	65.2$_{-2.3}$	72.5$_{-4.4}$	70.4$_{-4.5}$	67.2$_{-1.5}$	66.5$_{-1.6}$	75.8$_{-1.7}$	75.1$_{-3.4}$	54.0$_{-8.7}$	53.0$_{-7.0}$

Table 8: **Background Transformation Experiments.** Owing to limited space, we present results on a subset of models.

ciency in the VL projection. Compared to LLaVA family, the VL proj. is more robust for the BLIP family, which use a Q-former and cross-attention for mapping visual information to language modality, making VL proj. more stable than LLaVA which simply uses single layer MLP.

5.2 Background Transformations

Here, we explore how VLMs process foreground vs. background information. Additionally, inspired by recent work that applies background transformations through visual prompting to enhance VLM performance (Yang et al., 2023; Shtedritski et al., 2023), we explore how these techniques impact intermediate spaces. While prior work only focus on CLIP-like contrastive models, we explore how VLMs like LLaVA, InstructBLIP process background and visual prompting information through intermediate spaces. The sample constructed stimuli are illustrated in Figure 5. We use COCO (Malik et al., 2024) and provide details of how we applied the transformations in Appendix A.5. The results of this experiment are summarized in Table 8. Overall, we observe that for most visual prompting transformations, the gains are higher in the visual and text space and relatively diminish in the response space. Targeted efforts to reduce information loss in the response space should increase the VLM response further for visual prompting.

Role of background context in VLMs. In order to understand the role of background, we consider two cases where - 1) background is removed 2) object is masked. We notice that clean (black & white) backgrounds (row 2 & 3 in Table 8) improve performance considerably in the visual and VL proj spaces (∼11-12%) and relatively less in the response space (∼4-5%), pointing to loss of information as discussed earlier. This is likely because removing the background removes any distractors

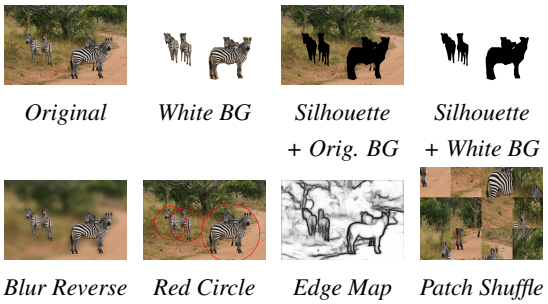

Figure 5: **Background Transformations.** Samples illustrating image transformations that we consider in our analysis of performance in *visual*, *text* and *response* spaces of VLMs.

(*e.g.*, objects in the background), helping the VLMs focus on the object of interest. In the absence of a clear foreground (object is masked with a silhouette; rows 4 & 5), we notice having background context (row 4) plays an important role and improves performance for all three spaces compared to row 5 (no background). Overall, this points to the fact that removing background helps when object details are clear but when objects are masked, VLMs can find hints in the background to help with object recognition . This hypothesis is also confirmed by blurring which we discuss next.

Visual prompting. We now consider a reverse blur transformation where the background is blurred w.r.t the object (Yang et al., 2023). Blurring the background seems to be "the best of both worlds", helping VLMs focus on the object while still preserving some background context—leading to the best performance overall (row 6). Furthermore, even though the performance improvement is similar to the black & white backgrounds for the visual and text spaces (∼11-12%), the increase in performance from blurring is higher for the response space (∼7-8%), pointing to the fact that some visual prompting techniques make up for the loss of information in response space and enhance performance more. Finally, we also experimented with adding a red circle to the object of interest (Sht-

edritski et al., 2023), with (row 7) and without background (row 8). The gains from the red circle in both cases are relatively smaller for all spaces.

Role of Shape. To investigate whether VLMs rely on the shape of object for recognition, we apply the edge maps and patch shuffle image transformations following(Mummadi et al., 2021; Geirhos et al., 2018). Edge maps (row 10) retains the object shape while removing other details such as texture, color and background, while patch shuffle (row 11) distorts the shape of the object. We observe a drastic drop in performance in the case of edge maps compared to a smaller drop for patch shuffle, suggesting that shape plays a relatively less important role in VLMs' object recognition. Finally, we note that for patch shuffle, the performance drop for the visual and text spaces is large while the response space is relatively more stable to such changes in high-performing VLMs (LLaVA, LLAVA-NEXT).

5.3 Summary of Insights on Image Transformations

① VLM's response space is most robust to corruptions, but posit this is largely due to deficiencies of information flow (see §4). VL projection space is least robust compared to other spaces, pointing to potential vulnerabilities.

② VLMs rely on both object information and background context to recognize objects and perform best when both are well-preserved.

④ Visual prompting techniques improve performance dramatically for visual and VL projection spaces but the improvements are comparatively diminished for response space; showing another limitation of miss-alignment of information.

⑤ We observe that VLMs, contrary to humans, rely on texture over shape information for their decision making. This is an important deficiency that may impact their generalization abilities. Inducing reliance on shape is of importance.

6 Conclusion

In this paper, we set out to understand the limitations of prominent VLMs on fundamental visual tasks. We go significantly beyond the current benchmarks, which simply measure final performance, and construct a series of tests that probe which specific components of VLM design may be lacking. Our analysis design reveals that for most visual tasks (except spatial understanding), the knowledge can be seen as being preserved in the visual and intermediate VL projection space, however, it does not translate to the final VLM response space (especially for tasks like fine-grained recognition, object counting, visual prompting). Overall, through our insights, we hope to encourage targeted efforts on reducing info. loss and translating knowledge to an accurate final VLM response.

Acknowledgments. This work was funded, in part, by the Vector Institute for AI, Canada CIFAR AI Chairs, NSERC Canada Research Chair (CRC), and NSERC Discovery Grants. Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, the Digital Research Alliance of Canada[4], companies sponsoring the Vector Institute, and Advanced Research Computing at the University of British Columbia. Additional hardware support was provided by John R. Evans Leaders Fund CFI grant and Compute Canada under the Resource Allocation Competition award.

Limitations and Ethical Considerations

We study capabilities of well-known VLMs to better understand their design on basic visual tasks such as visual recognition, object counting and spatial skills. Extensions to more advanced capabilities such as segmentation and visual reasoning would be interesting directions of future work. In any such analysis of existing models, one limitation is that it is challenging to calibrate for training data and augmentation techniques used in the various VLM models. We have put in our best efforts to objectively analyze all aspects of the experiments that could be under our control, however.

Societal Impact. VLMs themselves can, and already do, have a significant societal impact. Biases and inaccuracies in these models can have profound impacts on a broad spectrum of application domains. Our approach, which attempts to understand capabilities of such models and their shortcomings, can serve to mitigate some of these concerns by attributing the different modules in existing VLMs to specific capabilities. This may enable the possibility of carefully editing models to remove such biases and inaccuracies. We believe that our efforts of understanding the capabilities of these models at a module granularity is a first step to resolve the inaccurate or inappropriate behavior of such models in different applications.

[4] https://vectorinstitute.ai/#partners

References

Kumail Alhamoud, Shaden Alshammari, Yonglong Tian, Guohao Li, Philip Torr, Yoon Kim, and Marzyeh Ghassemi. 2025. Vision-language models do not understand negation. In *arXiv*.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. 2021. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

Jaemin Cho, Abhay Zala, and Mohit Bansal. 2023. Dall-eval: Probing the reasoning skills and social biases of text-to-image generation models. In *International Conference on Computer Vision (ICCV)*.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Preprint*, arXiv:2305.06500.

Mohamed El Banani, Amit Raj, Kevis-Kokitsi Maninis, Abhishek Kar, Yuanzhen Li, Michael Rubinstein, Deqing Sun, Leonidas Guibas, Justin Johnson, and Varun Jampani. 2024. Probing the 3D Awareness of Visual Foundation Models. In *CVPR*.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. 2010. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision (IJCV)*, 88(2):303–338.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix Wichmann, and Wieland Brendel. 2018. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *ArXiv*, abs/1811.12231.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. *Preprint*, CVPR:1612.00837.

Zhongyi Han, Guanglin Zhou, Rundong He, Jindong Wang, Tailin Wu, Yilong Yin, Salman Khan, Lina Yao, Tongliang Liu, and Kun Zhang. 2024. How well does gpt-4v(ision) adapt to distribution shifts? a preliminary investigation. *Preprint*, arXiv:2312.07424.

Dan Hendrycks and Thomas G. Dietterich. 2019. Benchmarking neural network robustness to common corruptions and surface variations. *Preprint*, arXiv:1807.01697.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.

Amita Kamath, Jack Hessel, and Kai-Wei Chang. 2023. What's "up" with vision-language models? investigating their struggle with spatial reasoning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9161–9175, Singapore. Association for Computational Linguistics.

Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. 2011. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization (FGVC)*.

Jeonghwan Kim and Heng Ji. 2024. Finer: Investigating and enhancing fine-grained visual concept recognition in large vision language models. *Preprint*, arXiv:2402.16315.

Nathan Lambert, Louis Castricato, Leandro von Werra, and Alex Havrilla. 2022. Illustrating reinforcement learning from human feedback (rlhf). *Hugging Face Blog*. Https://huggingface.co/blog/rlhf.

Bohao Li, Yuying Ge, Yixiao Ge, Guangzhi Wang, Rui Wang, Ruimao Zhang, and Ying Shan. 2024. Seed-bench-2: Benchmarking multimodal large language models. *Computer Vision and Pattern Recognition (CVPR)*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning (ICML)*.

Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Deepak Gotmare, Shafiq Joty, Caiming Xiong, and Steven Hoi. 2021. Align before fuse: Vision and language representation learning with momentum distillation. In *NeurIPS*.

Kunchang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. 2023b. VideoChat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*.

Zhiqiu Lin, Xinyue Chen, Deepak Pathak, Pengchuan Zhang, and Deva Ramanan. 2024. Revisiting the role of language priors in vision-language models. In *International Conference on Machine Learning*. PMLR.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024. Llava-next: Improved reasoning, ocr, and world knowledge.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Yun Liu, Ming-Ming Cheng, Xiaowei Hu, Jia-Wang Bian, Le Zhang, Xiang Bai, and Jinhui Tang. 2019. Richer convolutional features for edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1939–1946.

Hashmat Shadab Malik, Muhammad Huzaifa, Muzammal Naseer, Salman Khan, and Fahad Shahbaz Khan. 2024. Objectcompose: Evaluating resilience of vision-based models on object-to-background compositional changes. *arXiv preprint arXiv:2403.04701*.

Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S. Ecker, Matthias Bethge, and Wieland Brendel. 2019. Benchmarking robustness in object detection: Autonomous driving when winter is coming. *arXiv preprint arXiv:1907.07484*.

Chaithanya Kumar Mummadi, Ranjitha Subramaniam, Robin Hutmacher, Julien Vitay, Volker Fischer, and Jan Hendrik Metzen. 2021. Does enhanced shape bias improve neural network robustness to common corruptions? *ArXiv*, abs/2104.09789.

Roni Paiss, Ariel Ephrat, Omer Tov, Shiran Zada, Inbar Mosseri, Michal Irani, and Tali Dekel. 2023. Teaching clip to count to ten. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3147–3157.

Wujian Peng, Sicheng Xie, Zuyao You, Shiyi Lan, and Zuxuan Wu. 2024. Synthesize diagnose and optimize: Towards fine-grained vision-language understanding. In *CVPR*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *ArXiv*, abs/2305.18290.

Aleksandar Shtedritski, Christian Rupprecht, and Andrea Vedaldi. 2023. What does clip know about a red circle? visual prompt engineering for vlms. *ICCV*.

Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. 2022. Winoground: Probing vision and language models for visio-linguistic compositionality. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5228–5238.

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

Vishaal Udandarao, Max F. Burg, Samuel Albanie, and Matthias Bethge. 2023. Visual data-type understanding does not emerge from scaling vision-language models. *Preprint*, arXiv:2310.08577.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. 2011. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology.

Zongyu Wu, Yuwei Niu, Hongcheng Gao, Minhua Lin, Zhiwei Zhang, Zhifang Zhang, Qi Shi, Yilong Wang, Sike Fu, Junjie Xu, Junjie Ao, Enyan Dai, Lei Feng, Xiang Zhang, and Suhang Wang. 2025. Lanp: Rethinking the impact of language priors in large vision-language models. *Preprint*, arXiv:2502.12359.

Danfei Xu, Yuke Zhu, Christopher Choy, and Li Fei-Fei. 2017. Scene graph generation by iterative message passing. In *Computer Vision and Pattern Recognition (CVPR)*.

Lingfeng Yang, Yueze Wang, Xiang Li, Xinlong Wang, and Jian Yang. 2023. Fine-grained visual prompting. *Preprint*, arXiv:2306.04356.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. 2022. Coca: Contrastive captioners are image-text foundation models. *Preprint*, arXiv:2205.01917.

Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Y. Zou. 2022. When and why vision-language models behave like bags-of-words, and what to do about it? *ArXiv*, abs/2210.01936.

Kelly Zhang and Samuel Bowman. 2018. Language modeling teaches you more than translation does: Lessons learned through auxiliary syntactic task analysis. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 359–361, Brussels, Belgium. Association for Computational Linguistics.

Yuhui Zhang, Alyssa Unell, Xiaohan Wang, Dhruba Ghosh, Yuchang Su, Ludwig Schmidt, and Serena Yeung-Levy. 2024. Why are visually-grounded language models bad at image classification? *Conference on Neural Information Processing Systems (NeurIPS)*.

A Appendix

In this supplementary section, we discuss the following details, which could not be included in the main paper owing to space constraints:

- Performance of VLM response on scaling LLM Decoder for coarse and fine-grained recognition (in continuation to Figure 3 of main manuscript)
- Implementation details regarding task prompts and probe design (in continuation to Section 3 of main manuscript)
- Implementation details of different background transformations we consider (in continuation to Section 5.2 of main manuscript)
- Details of VLM model variants that were used for our analysis (in continuation to Section 3 of main manuscript)
- More details about the data and classes we use for our experiments (in continuation to Section 3 of main manuscript)

A.1 Scaling LLM Decoder

In this section, we elaborate on the results showing effect of scaling language decoder of competitive instruction-tuned VLMs (*i.e.*, InstructBLIP, LLaVA, LLaVA-NEXT) on visual, text and response space (in continuation to the Figure 3 bar plot of main manuscript). Table 9 shows the results for PASCAL VOC (coarse-grained) and Stanford Dogs (fine-grained) datasets as well as the average performance to help observe overall trends. On average, we notice that a bigger LM decoder (i.e 13B vs 7B model variant) does improve performance to some extent, including in the alignment space, for most cases but major trends discussed for coarse/fine object recognition remain same. Furthermore, we notice that the improvements on increasing decoder size are relatively larger for weaker instruction-tuned models like InstructBLIP but gradually saturate for models like LLaVA-1.5 and LLaVA-NEXT which have stronger instruction tuning and better overall in terms of performance.

A.2 Task Prompts and Evaluation of Response

In this section, we discuss the details of our task prompts and design of our trainable probes and inference mechanism we use to analyse the knowledge in three spaces of a VLM. As discussed in main manuscript Section 3, we consider different categories of VLM models (*i.e.*, Contrastive Multi-Encoder models (like, CLIP), Encoder-Decoder Generative models (like, BLIP2) and instruction-tuned models (like, LLaVA, InstructBLIP) which demand the use of different strategies to best evaluate their capabilities on visual understanding tasks, which we discuss below.

- **CLIP, CoCa and ALBEF:** Given these are multimodal contrastive models with a separate encoder for visual and text modalities, we evaluate the response space performance using standard zero-shot inference procedure with the help of cosine similarity in the shared embedding space. Specifically, we use similarity visual features (from visual encoder) and text prompt features (from text encoder) for inference and choose the class prompt with the maximum similarity as the predicted answer. We use the default prompts used for CLIP such as `[A photo of a cat, A photo of a dog. . . . .]` for **object recognition** task, `[There is one cat, There are two cats, There are three cats ..]` for **object counting** task and `[Dog is to left of cat, Dog is to the right of cat, Dog is below cat . . .]` for **spatial orientation** understanding tasks.
- **LLaVA-1.5, LLaVA-NEXT, BLIP2 and InstructBLIP** For VLMs which generate textual response, we used the standard procedure as followed in previous work (Kim and Ji, 2024; Han et al., 2024), where a given visual task, we prompt the model with a question and choices for the task, and ask it to select (generate) the right answer from the choices. For the task prompts, we follow the prompts similar to those used in previous work by (Han et al., 2024). For example, to evaluate object recognition for LLaVA model, we use the prompt `What is the central object in the image out of the following list of choices? Make sure to answer in one word. Choices: [cat, dog, frog ...]`. Note that LLaVA-1.5 and LLaVA-NEXT are specifically instruction-tuned to understand if the user wants it to answer in one word which is why we incorporate the phrase "Make sure to answer in one word". For BLIP2 and InstructBLIP model we use the same prompt *i.e.*, `What is the central object`

VLM Method		PASCAL VOC				Dogs				**Average**			
		Visual	VL proj	Res.(probe)	Res.(text)	Visual	VL proj	Res.(probe)	Res.(text)	Visual	VL proj	Res.(probe)	Res.(text)
InstructBLIP (Dai et al., 2023)	7B	98.0	98.6	95.0	70.0	93.6	92.6	28.5	12.0	95.8	95.6	61.8	41.0
InstructBLIP (Dai et al., 2023)	13B	98.0	98.6	96.5	66.0	93.6	92.8	46.2	26.8	95.8	95.7	71.4	46.4
LLaVA-1.5 (Liu et al., 2023)	7B	96.8	95.8	91.1	90.2	91.9	88.7	25.9	19.9	94.4	92.3	58.5	55.1
LLaVA-1.5 (Liu et al., 2023)	13B	97.0	96.6	95.1	93.2	91.9	90.0	37.3	30.6	94.5	93.3	66.2	61.9
LLaVA-NEXT (Liu et al., 2024)	7B	97.3	96.6	94.1	94.1	90.4	87.2	37.5	31.0	93.9	91.9	65.8	62.6
LLaVA-NEXT (Liu et al., 2024)	13B	97.3	96.2	95.7	95.8	90.0	86.5	37.0	36.9	93.7	91.4	66.4	66.4

Table 9: **Effect of Scaling LLM Decoder on three spaces for PASCAL (coarse-grained) and Stanford Dogs (fine-grained) recognition**

Transformation	LLaVA-NEXT (Liu et al., 2024)				LLaVA-1.5 (Liu et al., 2023)				InstructBLIP (Dai et al., 2023)			
	Visual	VL proj	Resp. (probe)	Resp.(text)	Visual	VL proj	Resp. (probe)	Resp.(text)	Visual	VL proj	Resp. (probe)	Resp.(text)
Snow	94.0	93.3	92.8	92.9	95.6	94.5	92.0	91.6	96.7	97.1	92.7	70.5
Uniform noise	90.0	88.0	87.2	87.2	90.5	88.2	87.5	87.0	94.7	95.5	88.5	69.5
Gaussian Noise	85.5	84.3	82.7	83.0	88.0	86.0	85.0	85.0	94.3	95.2	89.3	68.0
Shot Noise	89.0	86.4	85.7	85.8	89.5	88.5	86.5	86.3	95.7	96.1	92.5	69.0
Impulse Noise	87.5	85.5	84.8	85.0	87.8	87.0	85.5	85.5	95.1	95.8	90.5	68.0
Motion blur	92.2	92.0	90.5	90.5	93.2	91.9	90.0	90.0	95.6	95.8	89.5	70.5
Defocus blur	89.5	88.0	86.7	86.8	91.0	89.2	87.0	86.4	95.2	95.8	89.6	71.0
Average Corrupt.	89.7	88.2	87.2	87.3	90.8	89.3	87.7	87.4	95.3	95.9	90.5	69.5
Original	97.3	96.6	94.1	94.1	96.8	95.8	91.0	90.2	98.0	98.6	95.0	70.0

Table 10: **Visual Corruptions**- Results for individual corruptions for VLMs in continuation to Table 7 from main manuscript.

Transformation	BLIP-2 (Li et al., 2023a)			
	Visual	Text	Resp. (probe)	Resp. (text)
Snow	96.2	96.8	65	40.0
Uniform noise	94.7	94.7	55.0	34.5
Gaussian noise	93.9	94.4	54.0	38.6
Shot noise	95.7	96.1	54.0	38.5
Impulse noise	95.0	94.7	54.5	37.5
Motion blur	95.8	96.0	55.0	38.4
Defocus blur	95.2	95.4	65.5	39.9
Average Corrupt.	95.2	95.4	57.5	38.2
Original	98.0	99.0	57.6	38.0

Table 11: **Visual Corruptions**- Results for individual corruptions for VLMs in continuation to Table 7 from main manuscript.

`in the image out of the following list of choices? Choices: cat, dog, frog ...` without the phrase "Make sure to answer in one word" since these models do not understand answering in one word and have continuous outputs. We found that adding the phrase reduced performance by distracting these VLMs model, thus we remove it as we want to fairly and best evaluate the response space capabilities of all VLMs. We follow a similar strategy for counting and spatial orientation tasks where we ask `How many dogs are there in the image? Choices: 1,2,3,4` and `What is the spatial location of dog relative to cat? Choices: left, right, above, below` and expect it to answer with the total number of objects or the spatial orientation between objects.

Evaluation of textual VLM response - As detailed in the main manuscript, to facilitate fair comparison with visual and text feature probes and comprehensive evaluation of response space, we use 2 strategies - 1) training linear probe atop of output response tokens from the language decoder (we refer to this LM Dec.) and 2) evaluating the textual response of the VLM for a given task using matching between the textual VLM response and the ground truth class label (we refer to this Resp.). We discuss the latter Resp. strategy here and discuss the former LM Dec. strategy in next Section A.3 (probe design).Specifically, in order to evaluate whether the VLM answered with the correct class from the choices, we use matching between the textual VLM response and the ground truth class label similar to previous work on evaluating VLMs (Kim and Ji, 2024), where we check which classname in the choices matches the best with the VLM output with the help of regex based matching and fuzzy matching to make sure we match the output to the correct classname irrespective of punctuation, capitalization or style of output (one word vs long/-continuous output).

A.3 Control Task for Probes

When probing is used as a mechanism to test specific properties encoded by representations, it is essential to make sure that success on the probing task implies the representation encodes the properties useful for the task rather than they were learned by the probe itself. We follow previous work and report the performance on a control task in which the gold labels have been randomly shuffled (Zhang and Bowman, 2018; Hewitt and Liang, 2019). We expect a good probe to have low accuracy on the control task. Table 12 shows that

VLM Method	Control Task Performance		
	Visual	Text	Resp. (Probe)
Classification (coarse-grained)	10.4	10.6	15.5
Classification (fine-grained)	4.7	6.1	4.9
Spatial	25.4	23.6	22.5
Counting	23.8	25.7	9.0

Table 12: **Control Task for Probes**: We show performance for control task for Coarse-grained classification (PASCAL), Fine-Grained (Stanford Dogs), Counting and Spatial Tasks (PaintSkills). The performance is low for all three spaces supporting the validity of our probing mechanism

the performance of all three spaces of the state-of-the-art LLaVA-NEXT model on the control task for all tasks (coarse- and fine- grained classification, counting, and spatial arrangement) is poor; supporting the validity of our probing mechanism.

Given that we treat each visual task as classification and the datasets we use for visual tasks have ground-truth labels for each image (*i.e.*, object class labels, spatial orientation labels and object counting labels), we use simple *accuracy* (on validation set) as the metric for evaluation on object recognition, spatial understanding, and object counting tasks; following previous work (Cho et al., 2023). Given our linear probes are lightweight we use 1 A100 GPU for all our experiments.

A.4 Visual Corruption Results

In the main manuscript, we reported only the average performance across all corruptions (i.e Corrupt.) Table 7 and compared it to the original performance (without corruptions) i.e Orig, owing to space constraints. Here, we provide the tables of performances for each corruptions (Table 10 and Table 11)

A.5 Implementation: Background Transforms

In this section, we discuss the implementation of background transforms used in Sec.5.2. Specifically, we refer to recent work on background transformation (Malik et al., 2024) which provides COCO images and corresponding masks for foreground objects extracted using FastSAM. The presence of foreground object masks for each image helps us separate the object foreground and background and apply simple transformation techniques to get the `white background`, black `background changes` and `silhouettes`. For `reverse blur`, we follow (Yang et al., 2023) and blur the background with the help of gaussian filter. To generate `edge images` which retain global object shapes and remove texture related information, we use the technique previously used by (Mummadi et al., 2021) to get edge maps with the help of RCF (Liu et al., 2019) based edge detection. For the `red circle` transformation, we simply us contour detection for the object of interest and draw the minimum enclosing circle using OpenCV library. Finally, for `patch shuffle`, we use pytorch transformations to divide image into 4x4 grid and shuffle the patches.

A.6 Experimental Setup Details

Choice of VLM variants. Here, we elaborate on the details of specific model variants we use for our analysis.

- ***CLIP*** (Radford et al., 2021). Contrastive Language-Image Pre-Training (CLIP) consists of separate image and language encoders trained jointly using a contrastive objective that aligns the two latent representations. From the perspective of analysis in this paper, CLIP has the *visual latent* which can be linear probed and the *response space* performance can be obtained by image-text similarity based matching in the latent space; it does not have an projection/alignment (*text space*).

  For the experiments in this paper we use a variant with `ViT-L-14-336` backbone as visual encoder. Our choice is motivated by the fact that this variant is similar to the encoders of other VLMs we consider for our analysis (in terms of architecture and parameter size) and hence allows for fair comparison.

- ***CoCa*** (Yu et al., 2022).- Contrastive Captioners are Image-Text Foundation Models (CoCa) uses contrastive loss (similar to CLIP) jointly in combination with a captioning (generative) loss making it a hybrid contrastive + encoder-decoder generative model. It has *visual latent* and the *response space* performance can be obtained by image-text similarity based matching in the latent space similar to CLIP.

  For the experiments in this paper we use a variant with `ViT-L-14` backbone as visual encoder sinces it is similar to the encoders of other VLMs we consider for our analysis (in terms of architecture and parameter size) and hence allows for fair comparison.

- ***ALBEF*** (Li et al., 2021).- ALign BEfore Fuse first encodes the image and text independently and adds an additional multimodal encoder

to fuse the image features with the text features through cross- modal attention. It uses a combination of image-text contrastive (ITC), image-text matching (ITM) and masked language modeling (MLM) losses.

For the experiments in this paper we, we use the standard LAVIS library implementation and evaluate the `albef-feature-extractor (base)` model variant. We use the same inference procedure as used for CLIP and CoCa models where we probe the visual latent features and *response space* performance can be obtained by image-text similarity based matching.

- ***BLIP 2*** (Li et al., 2023a) and ***InstructBLIP (Dai et al., 2023)***. Bootstrapping Language-Image Pre-training (BLIP) is a popular VLM model that leverages pre-trained image encoder (CLIP) and a LLM decoder (*e.g.*, OPT, FlanT5) and learn to align the image representation to the expected input of the LLM decoder. The alignment mechanism takes the form of a small transformer-based neural network (Q-former) that is trained with a variety of losses. While standrad BLIP2 model keeps the image encoder and language decoder frozen, the instruction fine-tuned InstructBLIP model variant uses high quality image-language pairs to instruction fine-tune the performance of the full architecture.

  For our analysis we use variants of BLIP2 and InstructBLIP that are closest in size to other VLM models we consider. For BLIP, this means `VIT-G-14` as the visual encoder and `OPT` as the language decoder. For InstructBLIP, we use the model with `VIT-G-14` as the visual encoder and `Vicuna` as the language decoder.

- ***LLaVA and LLaVA-NEXT*** (Liu et al., 2023). Large Language and Vision Assistant (LLaVA) is a popular and one of the most competitive VLM model for general-purpose visual and language understanding. It uses language-only GPT-4 to generate multimodal language-image instruction-following data and implements carefully designed training procedures and architectural choices that results in impressive multimodal chat abilities. In our analysis, we use the `LLaVA-1.5` and `LLaVA-NEXT` model variants which shows excellent instruction following and multimodal chat capabilities across different VLM benchmarks. The variant we consider uses `CLIP ViT-L-14` backbone as visual encoder (same as the CLIP model above) and `Vicuna` as LLM decoder. LLaVA-NeXT has improved reasoning, OCR, and world knowledge compared to LLaVA-1.5, thanks to better instruction tuning on high quality user instruct data. It achieves the best performance among open-source LMMs; exceeds Gemini Pro and outperforms Qwen-VL [1] on several benchmarks (Liu et al., 2024).

A.7 Dataset Statistics and Class Details

In this section, we elaborate on the details of object classes and dataset statistics we use for the different tasks we consider in our paper. Note that as discussed before, we train our probes on the training data and test it on the val/test split following standard protocol.

Specifically, for coarse-grained recognition we consider the the PASCAL VOC with object classes `['aeroplane', 'bicycle', 'bird', 'boat', 'bottle', 'bus', 'car', 'cat', 'chair', 'cow', 'diningtable', 'dog', 'horse', 'motorbike', 'person']`and,
PaintSkills dataset with object classes `['person', 'airplane', 'bicycle', 'dog', 'boat', 'car', 'fire hydrant', 'stop sign', 'umbrella', 'bench', 'suitcase', 'traffic light', 'bird', 'bear', 'potted plant']`.
For fine-grained recognition we consider the CUB data with classes `['black footed albatross', 'laysan albatross', 'sooty albatross', 'groove billed ani', 'crested auklet', 'least auklet', 'parakeet auklet', 'rhinoceros auklet', 'brewer blackbird', 'red winged blackbird', 'rusty blackbird', 'yellow headed blackbird', 'bobolink', 'indigo bunting', 'lazuli bunting']` and
Stanford Dogs dataset with object classes`['chihuahua', 'japanese spaniel', 'maltese dog', 'pekinese', 'shih-tzu', 'blenheim spaniel', 'papillon', 'toy terrier', 'rhodesian ridgeback', 'afghan hound', 'basset', 'beagle', 'bloodhound', 'bluetick', 'black-and-tan coonhound']`

As mentioned previously, for counting and spatial recognition task we use object counts `[1,2,3,4]` and spatial relations `[left,right,above,below]`

We follow standard splits of PaintSkills dataset (Cho et al., 2023) with 23,250/21,600/13,500 and 2,325/2,160/2,700 scenes for train and test splits

of object recognition/object counting/spatial relation understanding skills, respectively. For Pascal VOC/Stanford Dogs/CUB datasets we have 2100/1500/500 and 2100/1300/400 for train and test splits, respectively, with the aforementioned object classes.