Are Vision Language Models Texture or Shape Biased and Can We Steer Them?

Paul Gavrikov[1] Jovita Lukasik[2] Steffen Jung[3,6] Robert Geirhos[4] Bianca Lamm[1] Muhammad Jehanzeb Mirza[5] Margret Keuper[6,3] Janis Keuper[1,6]

[1]IMLA, Offenburg University [2]University of Siegen [3]Max Planck Institute for Informatics, Saarland Informatics Campus [4]Google DeepMind [5]ICG, Graz University of Technology [6]University of Mannheim

GitHub

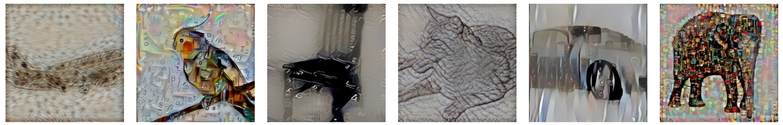
CVPR SEATTLE, WA JUNE 17-21, 2024

Motivation

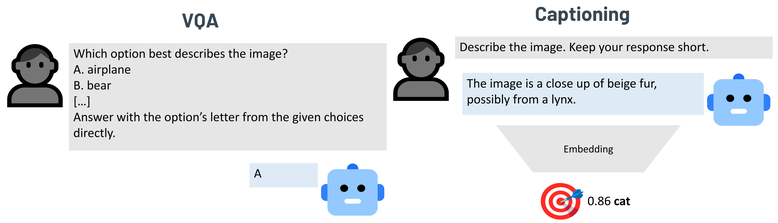
Perception biases have been studied extensively in uni-modal vision models. One well-known bias that received significant attention is the **texture/shape bias** [1,2,3,4]. It shows that while humans strongly prioritize shape information to recognize objects (96 %), most models prefer texture cues. Yet, the current generation of deep learning models is increasingly multi-modal. So what happens to this *purely visual* bias in those models? Does a VLM simply inherit it from its vision encoders, or does the bias interact with language?

[1] R. Geirhos et al., "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness", ICLR, 2019. [2] K. L. Hermann et al., "The Origins and Prevalence of Texture Bias in Convolutional Neural Networks", NeurIPS, 2020. [3] M. M. Naseer et al., "Intriguing Properties of Vision Transformers", NeurIPS, 2021. [4] A. Subramanian et al., "Spatial-frequency channels, shape bias, and adversarial robustness", NeurIPS, 2023.

Measuring the Texture/Shape Bias of VLMs



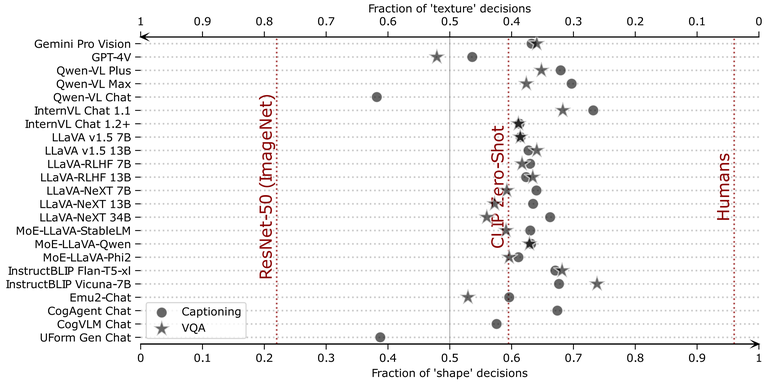
We measure the texture/shape bias on the *texture-shape cue-conflict* classification dataset [1]. It consists of 1,280 ImageNet validation samples with conflicting shape and texture cues synthetically generated via a style transfer from ImageNet samples. **Every sample has two conflicting labels from 16 ImageNet super-classes for texture and shape**, respectively. *Shape bias* is determined by the ratio of predictions with respect to the shape over texture class. The accuracy is defined by the ratio of predictions that match either the shape or texture class.



We test recognition in **two tasks** that we fit into the classification setting.

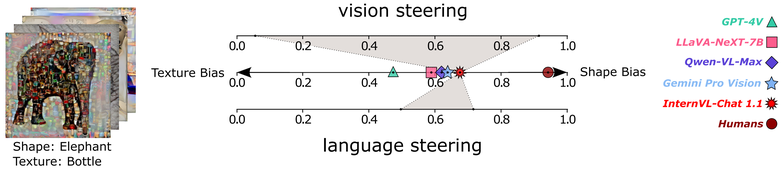
- **VQA:** We prompt the VLM with all class labels in a multiple-choice setting and ask it to select the best-fitting option. This is similar to traditional image classification by only allowing the model to respond with a single class and enforcing an answer.
- **Image Captioning:** We instruct VLMs to generate a brief description of the image. Then we use an embedding model to discriminate the description into a single label. Additionally, we analyze descriptions with an LLM to detect generic or multi-label descriptions.

Key Takeaways

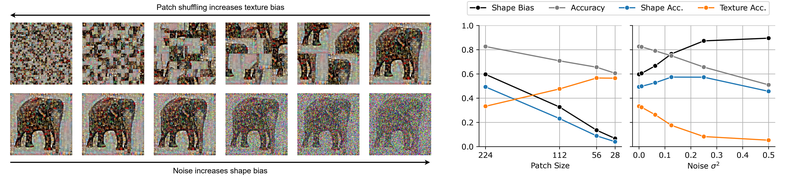


- **VLMs often prioritize shapes** in recognition tasks more often than ImageNet models but also than their vision encoders.
- **GPT-4V is a notable outlier:** the shape bias is almost neutral but the model also refuses to classify many test samples.
- The vision encoder determines the initial bias but the **representation is somewhat flexible**. Ultimately, the language model produces the biased decision (by forgetting the other cue).
- Shape bias in VLMs can be **aggressively steered through vision** in either direction.
- VLMs learn a multi-modal association between the terms "shape"/"texture", their synonyms, and their respective visual concepts → **Shape bias in VLMs can be steered through language during inference** to some extent.

Steering the Texture/Shape Bias

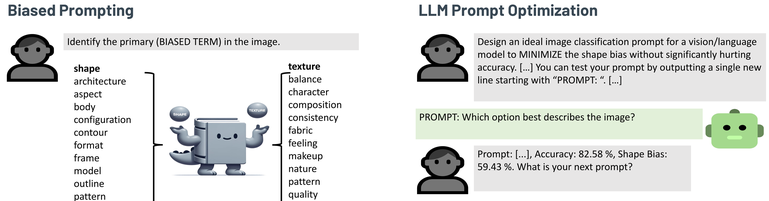


Vision Steering

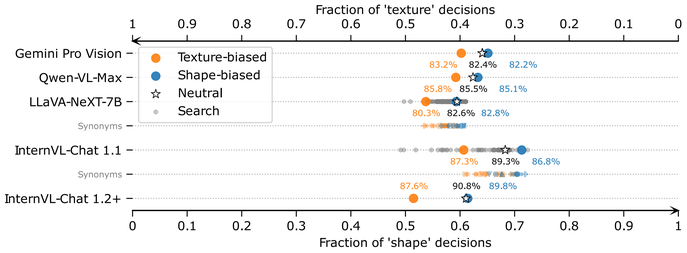


We can split an image into (increasingly smaller) patches and shuffle them to destroy shape cues. Similarly, we can add increasingly stronger Gaussian noise to destroy texture cues. This form of steering the texture/shape bias to either extreme comes at a cost in accuracy.

Language Steering



We specifically instruct the VLM to identify the "texture" or "shape" in the VQA task. Additionally, we test synonyms to understand if VLM has really learned the underlying concepts. Lastly, we optimize our simple hand-crafted prompts with a separate LLM.



Language steering (prompting) can steer to a smaller extent than vision steering but has only a small impact on accuracy (number next to the markers) if at all. This is a unique capability of LLM-based models.

Code: https://github.com/paulgavrikov/vlm_shapebias