TEXTUAL STEERING VECTORS CAN IMPROVE VISUAL UNDERSTANDING IN MULTIMODAL LARGE LANGUAGE MODELS

Woody Haosheng Gan* **Deqing Fu*** **Julian Asilis*** **Ollie Liu***
Dani Yogatama **Vatsal Sharan** **Robin Jia** **Willie Neiswanger**

University of Southern California
{woodygan, deqingfu, asilis}@usc.edu, me@ollieliu.com
{yogatama, vsharan, robinjia, neiswang}@usc.edu
*Equal Contribution

Abstract: Steering methods have emerged as effective and targeted tools for guiding large language models' (LLMs) behavior without modifying their parameters. Multimodal large language models (MLLMs), however, do not currently enjoy the same suite of techniques, due in part to their recency and architectural diversity. Inspired by this gap, we investigate whether MLLMs can be steered using vectors derived from their text-only LLM backbone, via sparse autoencoders (SAEs), mean shift, and linear probing. We find that text-derived steering consistently enhances multimodal accuracy across diverse MLLM architectures and visual tasks. In particular, mean shift boosts spatial relationship accuracy on CV-Bench by up to +7.3% and counting accuracy by up to +3.3%, outperforming prompting and exhibiting strong generalization to out-of-distribution datasets. These results highlight textual steering vectors as a powerful, efficient mechanism for enhancing grounding in MLLMs with minimal additional data collection and computational overhead.

1 Introduction

Steering large language models (LLMs) via their internal representations has emerged as a lightweight, interpretable paradigm for eliciting safe and controllable behavior [Li et al., 2023a, Turner et al., 2023, Sharkey et al., 2025, *inter alia.*]. This approach provides a targeted, interpretable way to guide model outputs without parameter updates. However, similar steering approaches have not yet gained prominence for *multimodal large language models* (MLLMs). This is in part due to their relative recency, as well as the heterogeneity of their architectures compared to text-only LLMs. For example, many steering methods assume access to a dataset of contrast pairs [Marks and Tegmark, 2023] to construct steering vectors, which may not be readily available for multimodal inputs.

To address these limitations, we develop a multimodal steering paradigm that is agnostic to model architecture and does not require specialized multimodal data. Noting that most MLLMs are adapted from a pretrained LLM backbone, we design multimodal steering techniques by repurposing existing language steering methods originally developed for the base (text-modality-only) LLM. This approach leverages existing LLM steering methods that are developed and tested broadly in the text domain. It has the potential to obviate the need for modality-specific modifications. In doing so, we extend the benefits of lightweight and interpretable steering to the multimodal setting with minimal overhead, paving the way for more adaptable and broadly applicable control of MLLMs.

Sitting at the core of our method is the observation that internal representations from a text-only LLM backbone can be repurposed to steer its multimodal counterpart. Specifically, we extract interpretable steering vectors from the text-only LLM backbone using techniques based on Sparse Autoencoders (SAEs), Mean Shift, and Linear Probing, and apply these vectors to the hidden states of the MLLM to enhance multimodal reasoning capabilities.

We evaluate our approach on several open-weight MLLMs across a diverse suite of visual reasoning tasks, including spatial relations and object counting. Our method consistently outperforms prompting, demonstrating the practical utility of leveraging text-only steering techniques in the multimodal regime. Consistent with prior work [Marks and Tegmark, 2023, Wu et al., 2025], we also observe that mean shift outperforms sparse autoencoders (SAEs) in key aspects of our implementation. Interestingly, although direct prompting is most effective for controlling *text-only* LLMs' behavior [Wu et al., 2025], direct prompting barely helps in improving *multimodal* LLMs' visual reasoning capabilities. Our contributions are summarized as follows:

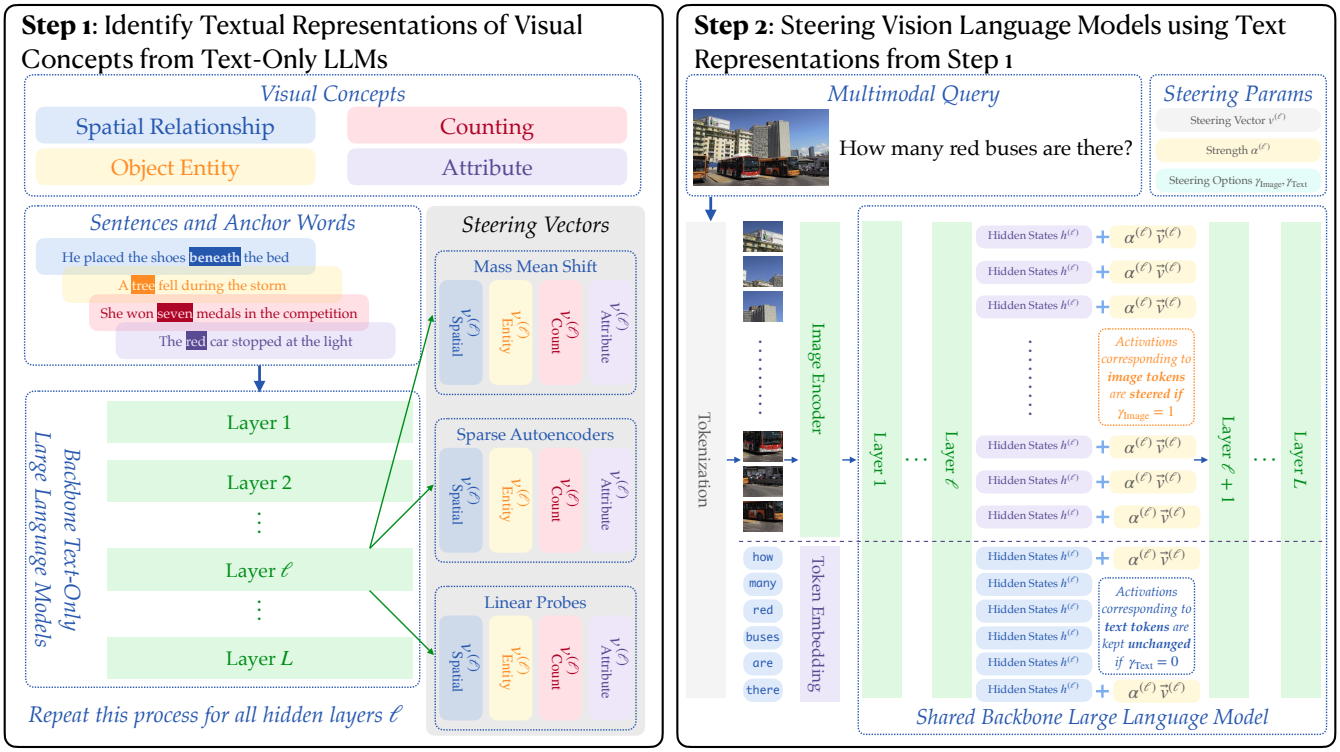


Figure 1: Overview of our steering methodology. For an MLLM with a text-only LLM backbone and a given image-bound prompt, we first identify the visual concept required to address the prompt (*e.g*, spatial relationships, counting, object entity, or attributes). For each hidden layer ℓ, we then determine corresponding steering vectors for the identified concept in the underlying LLM (*e.g.* `Gemma2-2B` is the backbone LLM for the its multimodal counterpart `PaliGemma2-3B`). We study the selection of such vectors using mean shift, linear probing, and sparse autoencoders. Finally, we intervene these steering vectors on activations corresponding to the *image tokens*, text tokens or both, depending on our choices of γ_{Image} and γ_{Text} from the original image-bound prompt.

- We introduce a plug-and-play multimodal steering paradigm that is compatible with existing internal-representation-based LLM steering techniques.
- We identify a transfer effect: representations from the text-only LLM backbone remain effective for steering its multimodal counterpart, even after vision-language post-training.
- We demonstrate consistent performance gains across multiple MLLMs and task categories. Importantly, we also show that textual steering vectors could generalize to out-of-distribution test sets and demonstrate significant performance gains.

2 Related Works

Representation-Based Steering methods are an effective family of methods for steering LLMs, often in two stages. **First**, they identify model components that influence target behaviors, using probing directions [Li et al., 2023a, Zou et al., 2023], activation differences [Li et al., 2023a, Turner et al., 2023, Panickssery et al., 2023, Marks and Tegmark, 2023, Lee et al., 2024], or lifted monosemantic features via SAEs [Lieberum et al., 2024b, Gao et al., 2025, Templeton et al., 2024, Marks et al., 2025] and their variants [Dunefsky et al., 2024], among other techniques. **Second**, they adjust steering hyperparameters to balance desiderata such as truthfulness [Lin et al., 2022, Hernandez et al., 2023, Li et al., 2023a], helpfulness [Zou et al., 2023], and quality.

While widely studied in LLMs, applying activation intervention to MLLMs remains elusive. To our knowledge, the only such effort is the VTI method of Liu et al. [2025], which extends LLM steering pipelines by constructing

intervention vectors from paired multimodal inputs and applying them to both visual and textual representations. In contrast, we show that interventions vectors constructed solely from text inputs in the unimodal LLM can influence the MLLM's multimodal behavior. This result highlights an underexplored form of cross-modal transfer enabled by the preserved semantics [Lieberum et al., 2024b] of the text backbone.

Shared Semantics refer to the representations unifying heterogeneous modalities of the same content, as identified across languages in multilingual LLMs [Artetxe et al., 2019, Wendler et al., 2024, Wu et al., 2024] and text/vision inputs in multimodal models [Huh et al., 2024, Luo et al., 2024, Wu et al., 2024]. Our work studies the transfer of steering effect across different modalities and training stages.

Multimodal Large Language Models are commonly developed by endowing a backbone LLM with visual processing components and fine-tuning on multimodal datasets, with some exceptions still pretrained from scratch [Team, 2024, OLMo et al., 2024, Chen et al., 2025]. Using an LLM backbone typically involves projecting the outputs of an image encoder (*e.g.*, Dosovitskiy et al. [2020], Zhai et al. [2023]) to the same dimension as the underlying LLM by an MLP, and concatenating the resulting image/text tokens as input to the LLM. The model can then be finetuned on multimodal data, possibly with frozen layers (*e.g.*, in the LLM) to preserve pretrained knowledge.

3 Toy Example

To demonstrate that textual representations can effectively intervene in visual understanding, we conduct a simple color perception experiment using GemmaScope [Lieberum et al., 2024a] for `Gemma-2-9B` for feature extraction and `PaliGemma2-10B-mix-448` [Beyer et al., 2024] as our target model. We present the model with a yellow-orange image (whose RGB hex code is #FFB400) and manipulate its perception by intervening in the hidden representations. Specifically, we find the normalized red vector from GemmaScope and we add this vector to the hidden states of image tokens at layer 20 as follows: $h'_{\text{image}} = h_{\text{image}} + \alpha \cdot v_{\text{red}}$, where α is the scale factor controlling intervention strength. Figure 2 shows how increasing the scale factor shifts perception along a color spectrum: initially yellow-orange dominates, then orange peaks at scale factor 50, and finally red becomes dominant beyond scale factor 75. This demonstrates that textual features can integrate with and modify visual understanding, supporting our hypothesis of unified cross-modal representations within these models. We include more color examples in Appendix B.

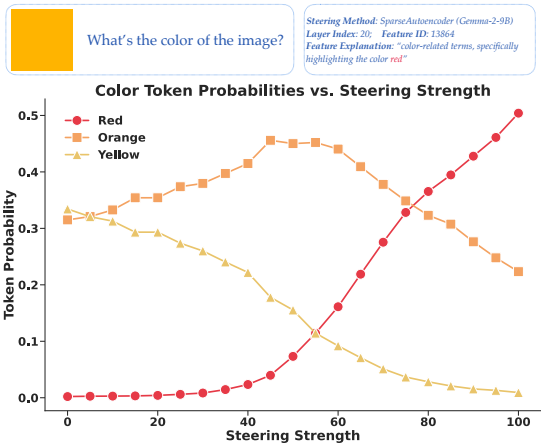


Figure 2: Effect of steering strength on color token probabilities.

4 Methods

Building on our successful intervention in Section 3, where we demonstrate that textual representations can effectively steer visual understanding, we now explore more systematic approaches to improve MLLMs' visual reasoning. Despite their growing success, Multimodal Large Language Models (MLLMs) still stumble on seemingly simple visual queries—miscounting objects, confusing left with right, and mishandling compositional prompts [Fu et al., 2024b]. When the same problems are posed in pure text, foundation models are far better than when they are asked to reason over images [Fu et al., 2024a].

A promising remedy in the text-only world is the use of steering vectors: compact directions in a model's activation space that are learned to emphasize or suppress specific behaviors. At inference time, the steering vectors are simply added to one or more hidden layers, where the best layer and intervention strength could be found via grid search.

This observation motivates the central question of this section:

Can steering mechanisms for textual representations rectify the shortcomings of MLLMs?

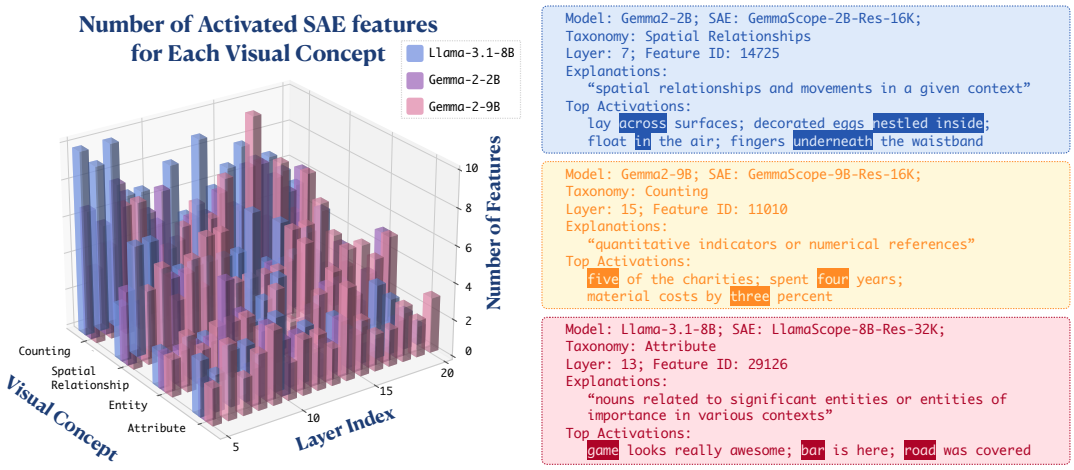


Figure 3: **Left**: Number of SAE features associated with each taxonomy (counting, spatial relationship, entity, and attribute) across the layers of Llama-3.1-8B, Gemma2-2B, and Gemma2-9B. Notably, SAE features for such visual concepts are sparse, numbering fewer than 10 across 16k total SAE features (Gemma2-2B/9B) or 32k features (Llama-3.1-8B). **Right**: Examples of features corresponding to visual concepts, identified by the layer whose activation space they inhabit and their (arbitrary) feature ID. The feature's explanation summarizes its semantic meaning, as evidenced by the tokens and contexts on which it attains the greatest activations.

We now review several techniques for extracting textual steering vectors from LLMs: sparse autencoders (SAE), mean shift, and linear probing (Probe). These techniques constitute Step 1 of our steering methodology, as displayed in Figure 1. In Section 4.5 we briefly discuss prompting, which will serve as a baseline for the three steering methods. The transferability of these methods to MLLMs (*i.e.*, Step 2) will be the subject of Section 5.

4.1 Crafting Datasets to Extract Steering Vectors

In order to find high-level textual representations for visual concepts, we first identify four important taxonomies for static images, as considered by prior work [Huang et al., 2023, Lin et al., 2024, Fu et al., 2025]: spatial relationship, counting, attribute, and entity. For each visual concept, we curate *a small set of sentence-anchor pairs*, where each pair contains a sentence exhibiting the visual concept and the specific anchor word representing that concept. Examples are shown in Table 3 of Appendix A.1. These sentence-anchor pairs serve as the foundation for all three steering vector extraction methods described in the following subsections.

4.2 Sparse Autoencoders (SAE)

Sparse autoencoders reconstruct the activations of an LLM's hidden layer using an MLP with a single hidden layer and a sparsity penalty on the hidden layer. More precisely, let $x = h^{(\ell)}(t) \in \mathbb{R}^D$ be the model activations for a token t at layer ℓ in an LLM. A SAE reconstructs x as $\hat{x} = b_{\text{dec}} + \sum_{i=1}^{F} f_i(x)\, W^{\text{dec}}_{\cdot,i}$, where $b_{\text{dec}} \in \mathbb{R}^D$ and $W^{\text{dec}} \in \mathbb{R}^{D\times F}$ are learned decoder weights, and $f_i(x)$ is the activation corresponding to feature i. Feature activations are computed using learned encoder weights $W^{\text{enc}} \in \mathbb{R}^{F\times D}$ and $b^{\text{enc}} \in \mathbb{R}^F$ as $f_i(x) = \sigma\left(W^{\text{enc}}_{i,\cdot}\, x + b^{\text{enc}}_i\right)$, where σ denotes an activation function of choice, *e.g.*, ReLU or JumpReLU.

The model is trained by minimizing the loss function $L = \mathbb{E}_x\left[\|x - \hat{x}\|_2^2 + \lambda \sum_{i=1}^{F} f_i(x)\, \|W^{\text{dec}}_{\cdot,i}\|_2\right]$, *i.e.*, L_2-reconstruction error and L_1-regularization on feature activations. In this formulation, unit-normalized decoder weight vectors $v_i^{(\ell)} := \frac{W^{\text{dec}}_{\cdot,i}}{\|W^{\text{dec}}_{\cdot,i}\|_2}$ serve as feature directions and $\alpha_i^{(\ell)}(t) := f_i(h^{(\ell)}(t))\, \|W^{\text{dec}}_{\cdot,i}\|_2$ as the activation strength of $v_i^{(\ell)}$ on token t.

For our experiments, we leverage pretrained SAEs—GemmaScope [Lieberum et al., 2024a] for Gemma-2-2B and Gemma-2-9B, and LlamaScope [He et al., 2024] for Llama-3.1-8B. We emphasize that *training SAEs is very costly* and

Algorithm 1 Find Textual Representations for Visual Concepts using SAEs

Require: Desired visual concepts $\mathcal{C}$. Layer index ℓ.
Require: Sentence and anchor word pairs $\{(s_1, w_1), \cdots, (s_K, w_K)\}$.
Require: Pretrained SAEs at layer ℓ.
▷ Find top activations and their corresponding SAE feature vectors.
$\mathcal{V}_0 = \{\}$
for each (s_j, w_j) **do**
  $\{\alpha_i^{(\ell)}(w_j), v_i^{(\ell)}\} \leftarrow$ Pass s_j into the pretrained SAE
  $\{v_{i_1}^{(\ell)}, \cdots v_{i_n}^{(\ell)}\} \leftarrow \text{Top}_n\{\alpha_i^{(\ell)}(w_j), v_i^{(\ell)}\}$ ranked by activation strength $\alpha_i^{(\ell)}(w_j)$
  $\mathcal{V}_0 \leftarrow \mathcal{V}_0 \cup \{v_{i_1}^{(\ell)}, \cdots v_{i_n}^{(\ell)}\}$
end for
▷ Filter out noisy SAE feature vectors.
$\mathcal{V} = \{\}$
for each $v_i^{(\ell)} \in \mathcal{V}_0$ **do**
  Find the explanation e and top activated tokens $\{t_1, \cdots, t_p\}$ for $v_i^{(\ell)}$
  if o3-mini(VerificationPrompt, e, $\{t_1, \cdots, t_p\}$, $\mathcal{C}$) is True **then**
    $\mathcal{V} \leftarrow \mathcal{V} \cup \{v_i^{(\ell)}\}$
  end if
end for
▷ Aggregate SAE vectors to one steering vector.
$v^{(\ell)} = \frac{1}{|\mathcal{V}|} \sum_{u \in \mathcal{V}} u$
return $v^{(\ell)}$

that an advantage of steering using text-based vectors is that one can leverage existing interpretability tools.

We then use the sentence-anchor pairs described in Section 4.1 to identify feature directions corresponding to the ideal visual concepts using Algorithm 1. We employ a two-stage procedure which, at the first stage, finds the top n activated features for anchor words w_j in sentences s_j. Interestingly, as shown in Figure 3, we find that each visual concept activates only a limited number of SAE features, indicating a sparse encoding of these concepts. At the second stage, We then verify their relevance to the target visual concepts. We use o3-mini [OpenAI, 2025] to verify that these features indeed align with the desired visual concept $\mathcal{C}$. When we prompt o3-mini for verification, we craft prompts to include both the explanation for the candidate feature vector $v_i^{(\ell)}$, and sample top activated tokens (see Figure 7 for the prompting template). We find that o3-mini can indeed filter out features unrelated to the desired visual concepts. Finally, we average these relevant feature vectors to create a single steering vector for each visual concept at each layer. Additional details are provided in Appendix A.1.

4.3 Mean Shift

An alternative approach to identify feature directions for visual concepts is through mean shift analysis, which has shown surprising effectiveness for steering in LLMs [Marks and Tegmark, 2023, Wu et al., 2025]. Given a text-only large language model h, for each taxonomy $\mathcal{T} \in \{\text{spatial relationship}, \text{counting}, \text{attribute}, \text{entity}\}$ and layer ℓ, we use the same set of sentence-anchor pairs $\{(s_1, w_1), \ldots, (s_K, w_K)\}$ as described in 4.1. We compute the mean shift vector between hidden states of the residual stream as:

$$m_{\mathcal{T}}^{(\ell)} = \frac{1}{K} \sum_{j=1}^{K} h^{(\ell)}(w_j) - \frac{1}{|\mathcal{S}_{\neg\mathcal{T}}|} \sum_{t \in \mathcal{S}_{\neg\mathcal{T}}} h^{(\ell)}(t)\,,$$

where $h^{(\ell)}(w_j)$ represents the residual stream activation of the anchor word w_j at layer ℓ and $\mathcal{S}_{\neg\mathcal{T}}$ is a control set of non-anchor tokens from the same sentences containing the anchor words. By using tokens from the same contextual environment but excluding the anchor words themselves, we isolate the specific representation of the visual concept from the general contextual information. We deliberately refrain from normalizing the vector $m_{\mathcal{T}}^{(\ell)}$, preserving its magnitude relative to the original hidden states.

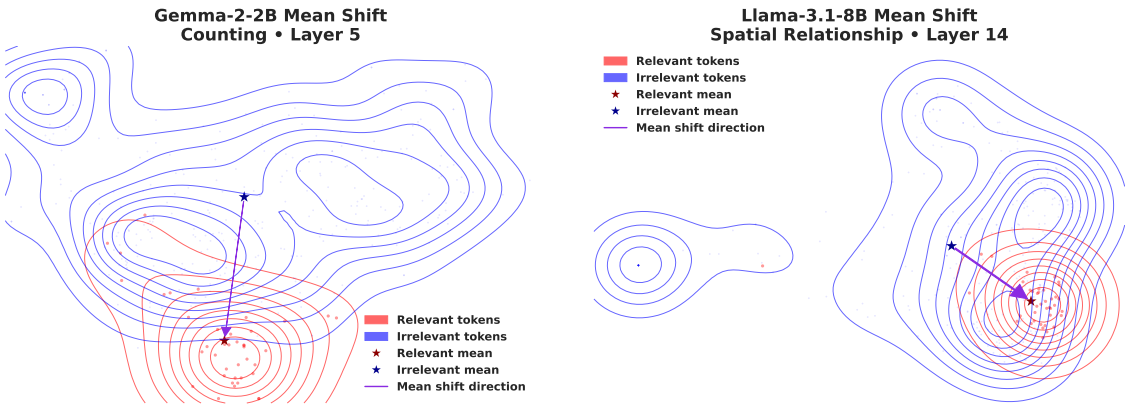


Figure 4: **Left:** Depiction of mean shift method for the counting feature for `Gemma-2-2B`. The mean shift direction points from the mean hidden state of irrelevant tokens to the mean hidden state of relevant tokens (*i.e.*, counting-related tokens). Activations are projected to two dimensions for the sake of visualization. **Right:** Similar depiction for the spatial relationship feature of `Llama-3.1-8B`.

4.4 Linear Probing

The third technique we employ is linear probing (a.k.a probe), training a classifier from the ℓ-th layer of a model [Alain and Bengio, 2016, Park et al., 2024]. Let $h^{(\ell)}(t) \in \mathbb{R}^D$ denote the activations for token t at layer ℓ. Using the same sentence-anchor pairs described in Section 4.3, we train a linear function to distinguish between anchor word activations and a control set of non-anchor tokens from the same sentences, isolating representations specific to the taxonomy.

As the hidden state dimensionality often exceeds our sample size ($K < D$), we first project to dimension $d < K$ using PCA. With $Q \in \mathbb{R}^{d \times D}$ as the PCA matrix, the probe separates:

$$\{h^{(\ell)}(w_j)Q^\top\}_{j \le K}, \qquad \text{and} \qquad \{h^{(\ell)}(t)Q^\top\}_{t \in \mathcal{S}_{\neg\mathcal{T}}},$$

where $\{(s_1, w_1), \ldots, (s_K, w_K)\}$ are the sentence-anchor pairs for concept $\mathcal{T}$ and $\mathcal{S}_{\neg\mathcal{T}}$ is our control set. The learned normal vector $v \in \mathbb{R}^d$ (pointing toward taxonomy-relevant points) yields the final steering vector $v' = Q^\top v$. We use $d = K/2$ in practice.

4.5 Prompting

We now briefly discuss prompting, which, despite not employing a steering *vector*, has displayed impressive abilities in text-only domains [Wu et al., 2025]. In Section 5, prompting will serve as a baseline for comparison with the previous vector-based methods, which will furthermore shed light on whether the efficacy of prompting transfers to multimodal settings.

For a given taxonomy $\mathcal{T}$, we generate a prompt meant to enhance an MLLM's visual reasoning ability with respect to $\mathcal{T}$ as follows: We first curate a collection of 30 prompts $\{p_1, \ldots, p_{30}\}$ of varying lengths by instructing `GPT-4o-2024-08-06` to generate instructions that will elicit improved reasoning with respect to $\mathcal{T}$, similar to the LLM-based prompt generation approach used in AxBench [Wu et al., 2025]. We then do a grid search for all prompts on a training set and retain only the best-performing prompt p_{best} for use on the test set. The performance of p_{best} is thus treated as the performance of prompting. Refer to Appendix A.2 for further detail.

5 Steering Improves Multimodal LLMs

Having established in §3 that textual steering vectors applied to non-output tokens can alter the behavior of MLLMs, we now investigate whether the textual steering vectors we identified in §4 can *improve* visual understanding in MLLMs when applied to intermediate representations.

Model	Intervention Tokens		Relation			Count		
	Text	Image	SAE	Probe	MeanShift	SAE	Probe	MeanShift
PaliGemma2-3B	—		76.0			59.3		
	✓		82.0 (+6.0)	77.3 (+1.3)	83.3 (+7.3)	60.0 (+0.7)	62.0 (+2.7)	60.0 (+0.7)
		✓	78.7 (+2.7)	76.7 (+0.7)	78.7 (+2.7)	62.0 (+2.7)	60.7 (+1.3)	62.0 (+2.7)
	✓	✓	81.3 (+5.3)	78.7 (+2.7)	81.3 (+5.3)	62.7 (+3.3)	62.0 (+2.7)	62.0 (+2.7)
	Prompting		74.0 (−2.0)			60.7 (+1.3)		
PaliGemma2-10B	—		79.3			63.3		
	✓		78.7 (−0.7)	77.3 (−2.0)	83.3 (+4.0)	63.3 (+0.0)	62.7 (−0.7)	64.0 (+0.7)
		✓	79.3 (+0.0)	79.3 (+0.0)	78.7 (−0.7)	63.3 (+0.0)	63.3 (+0.0)	64.7 (+1.3)
	✓	✓	78.7 (−0.7)	78.0 (−1.3)	83.3 (+4.0)	64.0 (+0.7)	63.3 (+0.0)	63.3 (+0.0)
	Prompting		77.3 (−2.0)			62.7 (−0.7)		
Idefics3-8B-Llama3	—		73.3			59.3		
	✓		76.0 (+2.7)	78.0 (+4.7)	80.0 (+6.7)	58.7 (−0.7)	58.0 (−1.3)	60.0 (+0.7)
		✓	78.0 (+4.7)	72.7 (−0.7)	76.7 (+3.3)	60.0 (+0.7)	59.3 (+0.0)	60.7 (+1.3)
	✓	✓	77.3 (+4.0)	78.7 (+5.3)	80.7 (+7.3)	62.0 (+2.7)	60.0 (+0.7)	60.7 (+1.3)
	Prompting		75.3 (+2.0)			59.3 (+0.0)		

Table 1: **Textual Steering Vectors Improve Multimodal LLMs' Visual Understanding**. Task-specific textual steering vectors reliably improve both spatial relation and counting performance across multimodal models. Combined interventions on both image and text tokens under the MeanShift framework yields the strongest gains.

5.1 Setup

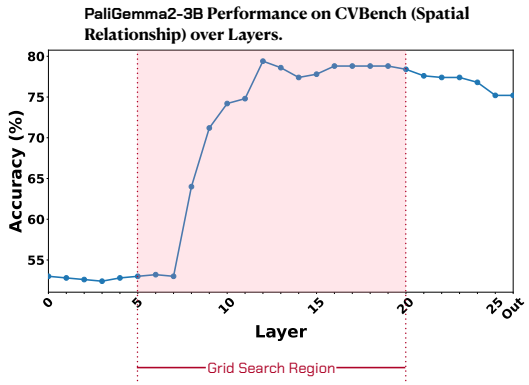
Models. We mainly investigate PaliGemma2 models with 3B and 10B parameters (specifically, `PaliGemma2-3B-mix-448` and `PaliGemma2-10B-mix-448`, which we refer to as `PaliGemma2-3B` and `PaliGemma2-10B` for brevity), and the Idefics3 model with 8B parameters (`Idefic3-8B-Llama3`). These models were chosen because they are slightly different in architecture – PaliGemma2 adopts the same prefix-LM masking strategy as PaliGemma, where the image tokens and textual instructions are cross-attended, whereas Idefics3 is an entirely autoregressive model following LLaVA. The steering vectors are extracted from `Gemma2-2B`, `Gemma2-9B`, and `Llama-3.1-8B` text-only LLMs, respectively, which serve as the pretrained backbones for the multimodal counterparts.

Dataset. In this section, we work with CV-Bench [Tong et al., 2024], an evaluation dataset with 4 major sub-categories: `Count`, `Relation`, `Distance`, and `Depth`. CV-Bench contains a total of 2,638 data points with each sub-category containing around 700 samples. For each sub-category, we split the samples in to 500-600 training samples reserved for grid search, and 150 samples used for testing.

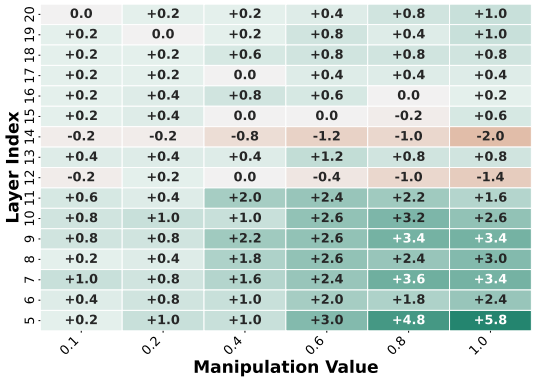
Grid Search. Given a set of concept-specific steering vectors $v^{(\ell)}$ extracted as described in §4, we identify the optimal injection layer ℓ and scale factor α via a simple grid search on a held-out "grid search" training split of CV-Bench. Concretely, for any model with L layers, we grid search on layer indices $\mathcal{I} = \{\ell_1, \cdots, \ell_{|\mathcal{I}|}\}$ and steering strength $\mathcal{A} = \{\alpha_1, \ldots, \alpha_{|\mathcal{A}|}\}$. For each $(\ell, \alpha) \in \mathcal{I} \times \mathcal{A}$, we intervene on the activations of the target tokens (image, text or both) at layer ℓ as

$$h'_{\text{target}}(\ell) = h_{\text{target}}(\ell) \; + \; \alpha \, v^{(\ell)}.$$

We then evaluate the model's task accuracy $\text{Acc}(\ell, \alpha)$ on the grid-search split. The optimal hyperparameters are chosen as $(\ell^*, \alpha^*) = \text{argmax}_{\ell \in \mathcal{L},\, \alpha \in \mathcal{A}} \text{Acc}(\ell, \alpha)$. Once (ℓ^*, α^*) is found, we fix these values for all subsequent evaluations on the held-out test set. In our experiments, we set $\mathcal{A}$ to $\{0.1, 0.2, 0.4, 0.6, 0.8, 1.0\}$ if the steering vectors are unnormalized (e.g., those found via MeanShift). If vectors are normalized (for SAE and Probe), for `PaliGemma2-3B` and `Paligemma2-10B`, we set $\mathcal{A}$ to $\{10, 20, 30, 40, 50, 60\}$, and for `Idefics3-8B-Llama3`, we use $\{0.2, 0.4, 0.6, 0.8, 1.0, 1.2\}$. because the average norm of its hidden states is much smaller than that of the `PaliGemma2` models. We set $\mathcal{I}$ to be the middle layers, where we observe the learning from image tokens is predominantly happening (see Figure 5a): $\{5, 6, \ldots, 20\}$ for `PaliGemma2-3B` and `Idefics3-8B-Llama3`, and $\{15, 16, \ldots, 30\}$ for `PaliGemma2-10B`. Notably, we

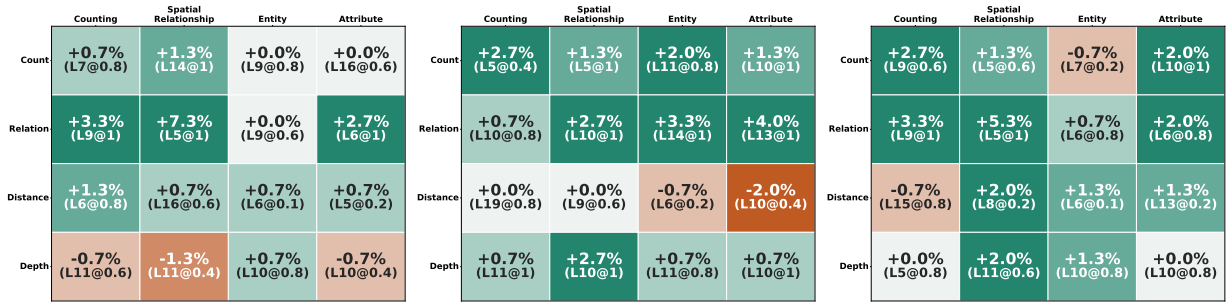


(a) We zero out models' attention to image tokens after layer ℓ and measure model performance. This reveals when visual information is processed and allows efficient grid search.



(b) Grid search on PaliGemma2-3B to locate the best (ℓ^*, α^*) for steering the model's spatial reasoning abilities. In this case, $\ell^* = 5$ and $\alpha^* = 1.0$.

Figure 5: Efficient Grid Search with PaliGemma2-3B on the Spatial Relationship Task.



(a) Intervening Text Tokens (b) Intervening Image Tokens (c) Intervening Both Tokens

Figure 6: Performance improvements on CV-Bench tasks when steering PaliGemma2-3B with MeanShift vectors. Each cell shows the percentage improvement in accuracy relative to the baseline. Rows represent different CV-Bench tasks, while columns represent different feature vectors used for steering. Text below improvements indicates the optimal layer number and intervention strength.

never steer the output token, as our focus is on modifying the model's internal representations.

5.2 Results

Table 1 presents a comparative analysis of three different models, PaliGemma2-3B, PaliGemma2-10B, and Idefics3-8B-Llama3, on tasks related to spatial relationships and counting in CV-Bench. The performance is evaluated with and without intervention tokens (text, image, or both) and across different steering methods (SAE, Probe, MeanShift, and Prompting).

Prompting Barely Steers. Table 1 indicates that prompting is often less effective than targeted interventions using text and/or image tokens, and in some cases even deleterious for model performance (especially for spatial relation tasks). Interestingly, this finding deviates from the text-only observations of AxBench [Wu et al., 2025], most likely due to the multimodality of the models and tasks – *i.e.*, textual prompts are not sufficient for multimodal LLMs to better *understand* images.

Steering Interventions Prove Effective. Table 1 demonstrates that steering interventions, using text, image, or combined tokens, consistently improve model performance on spatial relationship and counting tasks over baseline levels. For instance, PaliGemma2-3B's "Relation" accuracy with MeanShift rose from 76.0 to 83.3 using both tokens,

Dataset	Visual Concept	Model	Intervention Method				
			Baseline	Prompting	SAE	Probe	MeanShift
What'sUp-A	Spatial Relation	PaliGemma2-3B	62.7	59.4 (−3.3)	71.8 (+9.1)	78.5 (+15.8)	75.4 (+12.7)
		PaliGemma2-10B	68.5	71.0 (+2.5)	80.1 (+11.6)	71.6 (+3.0)	74.9 (+6.4)
		Idefics3-8B-Llama3	62.2	63.5 (+1.3)	64.1 (+1.9)	62.2 (+0.0)	61.9 (−0.3)
		Average Improvement	–	+0.2	**+7.6**	+6.3	+6.3
What'sUp-B	Spatial Relation	PaliGemma2-3B	60.6	58.4 (−2.2)	58.9 (−1.7)	57.5 (−3.1)	60.3 (−0.3)
		PaliGemma2-10B	81.8	81.5 (−0.3)	82.4 (+0.6)	82.1 (+0.3)	82.1 (+0.3)
		Idefics3-8B-Llama3	52.0	56.9 (+4.9)	56.2 (+4.2)	57.0 (+5.0)	63.4 (+11.5)
		Average Improvement	–	+0.8	+1.0	+0.8	**+3.8**
BLINK Object Localization	Spatial Relation	PaliGemma2-3B	41.2	42.3 (+1.0)	43.3 (+2.1)	42.3 (+1.0)	44.3 (+3.1)
		PaliGemma2-10B	51.6	51.6 (+0.0)	54.6 (+3.1)	53.6 (+2.1)	57.7 (+6.2)
		Idefics3-8B-Llama3	53.6	54.6 (+1.0)	56.7 (+3.1)	53.6 (+0.0)	55.7 (+2.1)
		Average Improvement	–	+0.7	+2.8	+1.0	**+3.8**
CLEVR	Count	PaliGemma2-3B	52.4	55.1 (+2.7)	70.7 (+18.2)	56.4 (+4.0)	67.1 (+14.7)
		PaliGemma2-10B	70.7	70.7 (+0.0)	74.9 (+4.2)	71.6 (+0.9)	80.4 (+9.8)
		Idefics3-8B-Llama3	59.8	65.1 (+5.3)	88.0 (+28.2)	84.4 (+24.7)	94.0 (+34.2)
		Average Improvement	–	+2.7	+16.9	+9.9	**+19.6**
Super-CLEVR	Count	PaliGemma2-3B	26.9	28.0 (+1.1)	32.0 (+5.1)	30.3 (+3.4)	33.1 (+6.3)
		PaliGemma2-10B	40.0	44.0 (+4.0)	40.6 (+0.6)	40.0 (+0.0)	44.6 (+4.6)
		Idefics3-8B-Llama3	66.5	64.0 (−2.5)	66.5 (+0.0)	67.5 (+1.0)	68.5 (+2.0)
		Average Improvement	–	+0.9	+1.9	+1.5	**+4.3**
Average Improvement			–	+1.0	+6.0	+3.9	**+7.6**

Table 2: Performance of textual steering on out-of-distribution datasets.

illustrating the general efficacy of these mechanisms.

MeanShift Demonstrates Strong Overall Performance. As shown in Table 1, Among the evaluated methods (SAE, Probe, MeanShift), MeanShift frequently performs as the most effective. This highlights MeanShift's robustness in leveraging textual representations and transferring to steer multimodal models, particularly for relational tasks.

Steering More Impactful for Spatial Relationships. Interventions yield more substantial accuracy improvements in the "Spatial Relationship" task than in "Counting". For instance, as shown in Table 1, with both tokens and MeanShift, PaliGemma2-3B gained +7.3 for relationships but only +2.7 for counting. This disparity may stem from spatial relationships being more directly influenced by highlighting salient object features and positions, while counting might demand a more holistic scene interpretation, less directly aided by these specific steering methods.

Intervention Transfers Across Tasks. As shown in Figure 6, intervention using a feature $\mathcal{T}$ can sometimes be effective even when transferred to a different task $\mathcal{T}'$. For instance, intervention using the "Attribute" vector yields improvement in the "Relation" task, as does the "Entity" vector to a lesser extent. It may be that improved attribute and entity abilities aid the model in identifying the objects whose spatial relationship is being queried.

6 Steering Improvements Generalize Out-of-Distribution

We now examine the ability of textual steering methods for MLLMs to generalize out-of-distribution, *i.e.*, to datasets on which the steering method's hyperparameters (ℓ, α) have not been tuned.

6.1 Setup

Datasets. We consider the transferability of textual steering on five datasets: What'sUp-A, What'sUp-B, BLINK Object Localization, CLEVR, and Super-CLEVR. What'sUp-A contains 408 images of pairs of household objects arranged in clear spatial relations of {"on", "under", "left", and "right"}, while What'sUp-B similarly contains 412 images with objects in the image closer in size [Kamath et al., 2023]. The BLINK Object Localization category contains 122 questions related to bounding boxes for large objects [Fu et al., 2024b]. Finally, we sampled 500 datapoints from

CLEVR [Johnson et al., 2017] and 200 datapoints from Super-CLEVR [Li et al., 2023b] to evaluate the OOD accuracy of textual steering in counting.

Steering Vector Hyperparameter Selection. We examine the previous three steering methodologies—SAE, MeanShift, and Probe—with a single choice of layer ℓ and scale factor α chosen independently of the test dataset. Specifically, for each test dataset, we select the (ℓ, α) pair that performed best on the corresponding CV-Bench task category (*e.g.*, "Relation" for the What'sUp datasets and Blink Object Localization focusing on spatial relationships, and "Count" for CLEVR and Super-CLEVR).

We emphasize that the steering methods' hyperparameters are *not* tuned to the datasets considered in this section, making this a true test of out-of-distribution generalization. Similarly, our prompting baseline uses the exact prompt prefix that performed best on the associated CV-Bench tasks. The only adaptation made was the use of a small validation subset (50 datapoints for What'sUp and CLEVR, 25 datapoints for BLINK Object Localization and Super-CLEVR) to determine the most effective token type for intervention (image, text, or both) before evaluating on the remaining data.

6.2 Results

Steering Remains Broadly Effective. Table 2 demonstrates that textual interventions are broadly effective across all 5 datasets considered, attaining an average improvement over all models and datasets of at least +3.9% for all three vector-based steering methods. Prompting, meanwhile, averaged a +1.0% improvement and worsened model performance in 4 cases, suggesting that it may be less effective for MLLMs than for text-only LLMs [Wu et al., 2025]. Notably, the out-of-distribution performance of textual steering appears to surpass its in-distribution performance on CV-Bench. We hypothesize that this may be due to the fact that the What'sUp and CLEVR datasets rely "purely" upon spatial reasoning and counting, respectively, whereas CV-Bench tests broader compositional understanding and thus benefits less from any individual intervention.

MeanShift Often Most Effective. We also find that MeanShift attains the greatest average performance of +7.6% across all models and datasets, and attains the best performance on a majority of all model-dataset pairs. This extends our finding from Section 5, suggesting that MeanShift may be the most broadly effective textual steering method for MLLMs.

Pronounced Gains For CLEVR. The dataset on which the largest improvements in performance are observed is CLEVR, by a 3x factor. The improvements of `Idefics3-8B-Llama3` largely drive these gains, with all vector-based steering methods (*i.e.*, non-prompting) achieving improvements of at least +24%. In contrast, Super-CLEVR achieves an accuracy improvement of only +4.3%, perhaps simply due to the fact that a more difficult dataset is more difficult to improve upon. Curiously, however, the baseline `Idefics3-8B-Llama3` model achieves a *higher* accuracy on this dataset than on CLEVR.

7 Discussion

We examine the ability of multimodal large language models (MLLMs) derived from backbone (text-only) large language models to be modified using textual steering vectors from their text-only backbone. We find that MeanShift, sparse autoencoders (SAEs), and linear probes can all enhance MLLMs' visual reasoning across diverse tasks on CV-Bench, with MeanShift demonstrating the strongest overall performance. Notably, the steering vectors derived from CV-Bench also generalize out-of-distribution to other datasets such as What'sUp, BLINK, and CLEVR, underscoring text-driven steering as a powerful and efficient medium for enhancing visual reasoning in MLLMs. A primary limitation of our work is that it requires the MLLM to have a text-only backbone, and that its efficacy is contingent upon the quality of the textual steering vectors derived from the backbone (which may be imperfect, see e.g. [Wu et al., 2025, Heap et al., 2025]). An interesting direction for future study is to further investigate and contrast the possibility of steering vectors extracted "directly" from MLLMs, rather than from their text-only backbones.

Acknowledgments

This research is supported in part by AWS credits through an Amazon Faculty research award, a NAIRR Pilot award, and Microsft accelerating foundations research grant. R. Jia was also supported by the National Science Foundation under Grant No. IIS-2403436. V. Sharan was supported in part by an NSF CAREER Award CCF-2239265 and an Amazon Research Award. W. Neiswanger was supported in part by the National Science Foundation under Grant No. CMMI-2427856. J. Asilis was supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-1842487. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. The work was done in part while some of the authors were visiting the Simons Institute for the Theory of Computing. D. Fu would like to thank Muru Zhang and Yuqing Yang for meaningful discussions on probing multimodal LLMs.

References

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. On the cross-lingual transferability of monolingual representations. *arXiv preprint arXiv:1910.11856*, 2019.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, G Heigold, S Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

Jacob Dunefsky, Philippe Chlenski, and Neel Nanda. Transcoders find interpretable llm feature circuits. *arXiv preprint arXiv:2406.11944*, 2024.

Deqing Fu, Ruohao Guo, Ghazal Khalighinejad, Ollie Liu, Bhuwan Dhingra, Dani Yogatama, Robin Jia, and Willie Neiswanger. Isobench: Benchmarking multimodal foundation models on isomorphic representations. In *First Conference on Language Modeling*, 2024a. URL https://openreview.net/forum?id=KZd1EErRJ1.

Deqing Fu, Tong Xiao, Rui Wang, Wang Zhu, Pengchuan Zhang, Guan Pang, Robin Jia, and Lawrence Chen. TLDR: Token-level detective reward model for large vision language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Zy2XgaGpDw.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive. In *European Conference on Computer Vision*, pages 148–166. Springer, 2024b.

Leo Gao, Tom Dupre la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=tcsZt9ZNKD.

Zhengfu He, Wentao Shu, Xuyang Ge, Lingjie Chen, Junxuan Wang, Yunhua Zhou, Frances Liu, Qipeng Guo, Xuanjing Huang, Zuxuan Wu, et al. Llama scope: Extracting millions of features from llama-3.1-8b with sparse autoencoders. *arXiv preprint arXiv:2410.20526*, 2024.

Thomas Heap, Tim Lawson, Lucy Farnik, and Laurence Aitchison. Sparse autoencoders can interpret randomly initialized transformers. *arXiv preprint arXiv:2501.17727*, 2025.

Evan Hernandez, Belinda Z Li, and Jacob Andreas. Inspecting and editing knowledge representations in language models. *arXiv preprint arXiv:2304.00740*, 2023.

Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. *Advances in Neural Information Processing Systems*, 36:78723–78747, 2023.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation hypothesis. *arXiv preprint arXiv:2405.07987*, 2024.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

Amita Kamath, Jack Hessel, and Kai-Wei Chang. 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, 2023.

Bruce W Lee, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Erik Miehling, Pierre Dognin, Manish Nagireddy, and Amit Dhurandhar. Programming refusal with conditional activation steering. *arXiv preprint arXiv:2409.05907*, 2024.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a. URL https://openreview.net/forum?id=aLLuYpn83y.

Zhuowan Li, Xingrui Wang, Elias Stengel-Eskin, Adam Kortylewski, Wufei Ma, Benjamin Van Durme, and Alan L. Yuille. Super-clevr: A virtual benchmark to diagnose domain robustness in visual reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14963–14973, June 2023b.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. *arXiv preprint arXiv:2408.05147*, 2024a.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2, 2024b. URL https://arxiv.org/abs/2408.05147.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.229. URL https://aclanthology.org/2022.acl-long.229/.

Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. In *European Conference on Computer Vision*, pages 366–384. Springer, 2024.

Sheng Liu, Haotian Ye, and James Zou. Reducing hallucinations in large vision-language models via latent space steering. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=LBl7Hez0fF.

Grace Luo, Trevor Darrell, and Amir Bar. Task vectors are cross-modal. *arXiv preprint arXiv:2410.22330*, 2024.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *arXiv preprint arXiv:2310.06824*, 2023.

Samuel Marks, Can Rager, Eric J Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=I4e82CIDxv.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2 olmo 2 furious. *arXiv preprint arXiv:2501.00656*, 2024.

OpenAI. o3-mini. https://openai.com/index/openai-o3-mini/, 2025. Accessed: 2025-05-13.

Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering llama 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06681*, 2023.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *International Conference on Machine Learning*, pages 39643–39666. PMLR, 2024.

Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeff Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Bloom, et al. Open problems in mechanistic interpretability. *arXiv preprint arXiv:2501.16496*, 2025.

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

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Ziteng Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal llms, 2024. URL https://arxiv.org/abs/2406.16860.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. Do llamas work in english? on the latent language of multilingual transformers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15366–15394, 2024.

Zhaofeng Wu, Xinyan Velocity Yu, Dani Yogatama, Jiasen Lu, and Yoon Kim. The semantic hub hypothesis: Language models share semantic representations across languages and modalities. *arXiv preprint arXiv:2411.04986*, 2024.

Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Dan Jurafsky, Christopher D Manning, and Christopher Potts. Axbench: Steering llms? even simple baselines outperform sparse autoencoders. *arXiv preprint arXiv:2501.17148*, 2025.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023.

Appendix

A Steering Vector Methodology

A.1 Sparse Autoencoders

We now provide further detail regarding the extraction of textual steering vectors for visual concepts using SAEs.

Recall that we consider four important taxonomies for image-related concepts: spatial relationship, counting, attribute, and entity. For each taxonomy, we sample K sentences $\{s_1, \ldots, s_K\}$ containing such visual concepts. In practice, we set K to 20. For each sentence s_j, we identify the anchor word for such visual concept as w_j, thus forming sentence-anchor pairs (s_j, w_j). See table 3 for several examples.

Table 3: Sample sentence and anchor word pairs for various taxonomies.

TAXONOMY	SENTENCE s_j	ANCHOR WORD w_j
Spatial Relationship	The cat is on the table She put the book under the chair	on under
Counting	There are three apples in the basket The teacher counted five children	three five
Attribute	The red car stopped at the light She wore a beautiful dress	red beautiful
Entity	The dog barked at the mailman A tree fell during the storm	dog tree

A.2 Prompting

We now elaborate upon our generation of prompts for eliciting taxonomy-specific visual reasoning in MLLMs. As described in Section 4.5, we generate a total of 30 candidate prompts for each taxonomy $\mathcal{T}$. To do so, we use template shown in figure 8. Here, we set the `num instructions` to 3 and `word count` $\in \{5, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$, resulting in total $3 \times 10 = 30$ steering prompts.

FEATURE ALIGNMENT VERIFICATION

Task: Determine if a neural network's sparse autoencoder (SAE) feature aligns with the taxonomy "{taxonomy}".

Taxonomy Definition: {taxonomy_definition}

Feature Information:
1. Feature's explanation: {feature_explanation}
2. Top activation examples (tokens wrapped in <top>...</top> have the highest activation values and are the most important to focus on):
   1. {activation_example_1}
   2. {activation_example_2}
   3. {activation_example_3}
   4. {activation_example_4}
   5. {activation_example_5}

Examples of features that DO align with the {taxonomy} taxonomy (notice how the key words are highlighted with <top>...</top> tags):
Example 1:
- Explanation: {explanation_1}
- Activations: {activations_1}
Example 2:
- Explanation: {explanation_2}
- Activations: {activations_2}

When making your decision, you should follow these rules:
1. First pay attention to the feature's explanation.
2. If you cannot decide, you should then pay special attention to the tokens highlighted with <top>...</top> tags, as these are the most highly activated tokens and strongest indicators of what the feature detects.
3. Also consider the diversity of the activation examples provided. If one feature only activates one particular word, it may not be as aligned as a feature that activates on a variety of words.

Based on the feature's explanation and the highlighted tokens in the activation examples, does this feature specifically detect or respond to {taxonomy_definition}? Your answer should start with YES or NO, then provide a brief reason. Do not start with any other words or phrases such as 'answer'.

Figure 7: Prompt template for querying GPT-o3-mini to verify whether a given feature is related to a visual taxonomy. For each taxonomy, the template employs a brief definition of the taxonomy, two example features that align with each taxonomy (for few-shot learning), and the top five activations of the feature in question.

STEERING PROMPT GENERATION

System prompt: You are an expert at creating concise, clear instructions for Multimodal Large Language Models (MLLM).

Your task:
- Generate {num_instructions} different instruction(5) that will make the Model focus on {concept} when answering questions about images
- Each instruction must be within {word_count} words
- Instructions should be direct and actionable, focusing specifically on how to emphasize {concept}

IMPORTANT FORMAT REQUIREMENTS:
- Begin each instruction with "INSTRUCTION:" followed by the instruction text
- Put each instruction on its own line
- Do not include any numbering, bullets, or other text beyond the requested instructions
- Do not include any explanations, introductions, or conclusions

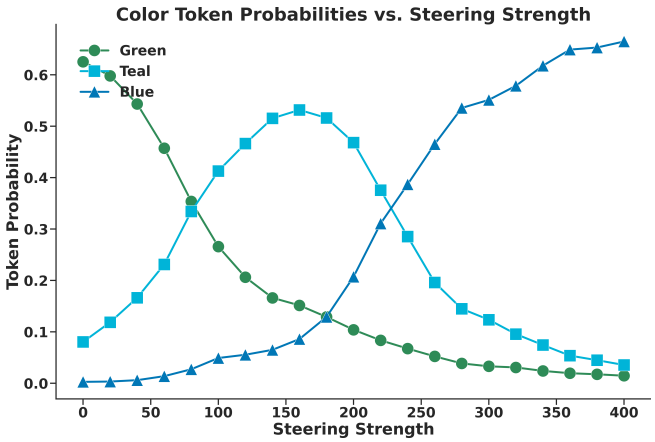
Example format for 2 instructions:
INSTRUCTION: First instruction text here within word limit.
INSTRUCTION: Second instruction text here within word limit.

User prompt: Create {num_instructions} instruction(s) about {concept} using {word_count} words or fewer each.

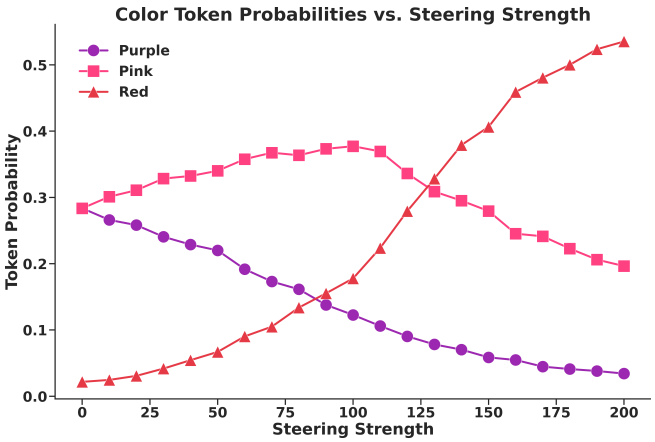
Figure 8: System and user prompt template for generating MLLM prompts.

B Additional Color Perception Intervention Examples

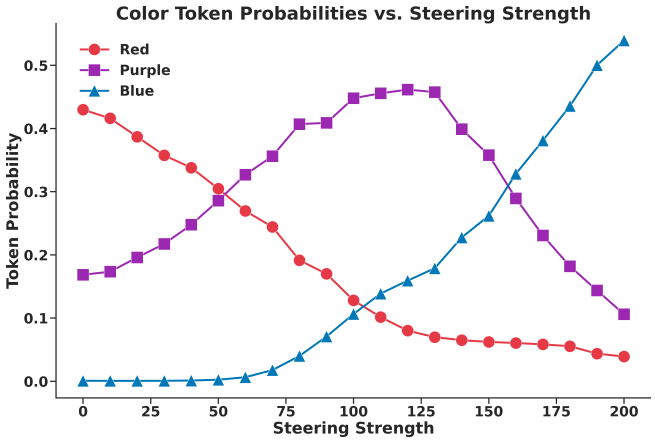
To further demonstrate the effectiveness of textual steering vectors in modifying visual understanding within MLLMs, we present additional color perception intervention examples using the same methodology described in §3.



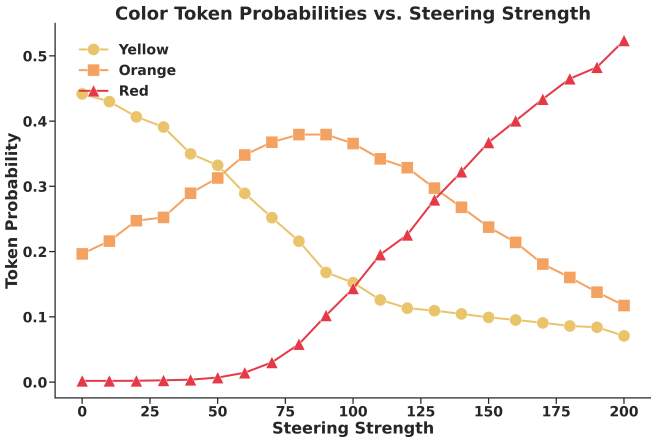
(a) Steering a green image toward blue perception. As the scale factor increases, the model's interpretation shifts from green to teal, and ultimately to blue.



(b) Steering a purple image toward red perception. The intervention gradually shifts the model's color association from purple to pink, and finally to red.



(c) Steering an red image toward blue perception. The intervention causes a gradual shift from red to purple, and ultimately to blue.



(d) Another example of steering a yellow image toward red perception, using a different steering vector from layer 18 of PaliGemma2-10B. As the scale factor increases, the model's interpretation transitions from yellow to orange, and finally to red.

Figure 9: Additional color perception intervention examples. In each case, we apply the normalized textual steering vector for the target color to the image tokens with increasing scale factors. The steering vectors are extracted from and applies to one selected layer from layer 17 to 20 in PaliGemma2-10B. The plots show token probability shifts, demonstrating how textual steering vectors can systematically modify the model's visual perception.

These additional examples further support our findings in §3. In each case, we see a clear progression of perception as the steering strength increases, with intermediate colors appearing during the transition. This confirms that textual steering vectors can produce predictable and continuous modifications to visual understanding.

Notably, all these interventions were performed using steering vectors derived solely from text data, yet they effectively modulate multimodal understanding. This provides additional evidence for our hypothesis that MLLMs develop unified cross-modal representations that can be manipulated through textual steering.

C Dataset Evaluation Details

In this section, we explain in detail how we prompt and evaluate the model's performance across datasets and provide representative examples for each dataset. Each prompt consists of four components: `model prefix`, `task prefix`, `taxonomy prefix`, and `question`. The `model prefix` is the specific instruction token sequence required by different model families to perform certain tasks. For `PaliGemma2` models, we use "`answer en`" as the model prefix, indicating that the model should answer in English for visual question answering tasks. For `Idefics3-8B-Llama3`, no model prefix is required, so this component remains empty. The `task prefix` provides task-specific instructions that constrain the format of the model's response. In multiple-choice questions, we use a task prefix such as "`Answer the multiple choice question by only responding with the letter of the correct answer.`" In `CLEVR` abd `Super-CLEVR` counting questions, we use "`Answer the question by only responding the number.`" The `taxonomy prefix` of each taxonomy is the prompt we sampled in Section 4.5, and it is only non-empty for the `Prompt` method. The `question` component contains the original question format from the dataset. Below are examples illustrating our prompting approach for each dataset.

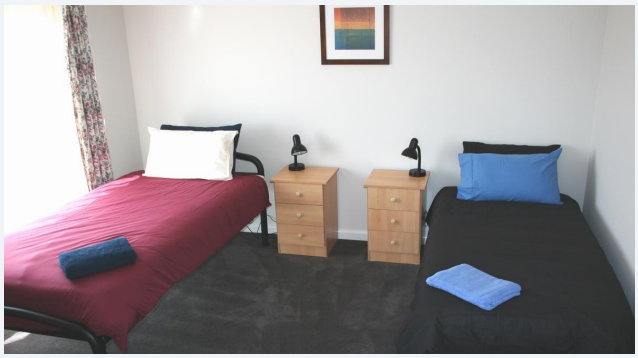
CV-BENCH RELATION

Image: 

[Model Prefix] answer en [Task Prefix] Answer the multiple choice question by only responding the letter of the correct answer. [Taxonomy Prefix] Emphasize objects' positions relative to each other. [Question] Considering the relative positions of the fork and the cup in the image provided, where is the fork located with respect to the cup? Select from the following choices.
(A) left
(B) right

Figure 10: Example prompt for the CV-Bench Relation dataset.

CV-BENCH COUNT

Image: 

[Model Prefix] answer en [Task Prefix] Answer the multiple choice question by only responding the letter of the correct answer. [Taxonomy Prefix] Prioritize counting objects and quantifying elements over other analysis. [Question] Answer the multiple choice question by only responding the letter of the correct answer. How many beds are in the image? Select from the following choices.
(A) 0
(B) 2
(C) 1
(D) 3
(E) 4

Figure 11: Example prompt for the CV-Bench Count dataset.

WHAT'SUP-A

Image: 

[Model Prefix] answer en [Task Prefix] Answer the multiple choice question by only responding the letter of the correct answer. [Taxonomy Prefix] Emphasize objects' positions relative to each other.[Question] Please select the correct caption for the image:
(A) A toilet roll under a chair
(B) A toilet roll to the left of a chair
(C) A toilet roll to the right of a chair
(D) A toilet roll on a chair

Figure 12: Example prompt for the What'sUp-A dataset.

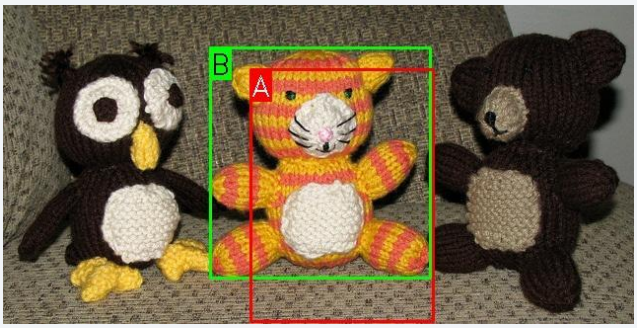
WHAT'SUP-B

Image: 

[Model Prefix] answer en [Task Prefix] Answer the multiple choice question by only responding the letter of the correct answer. [Taxonomy Prefix] Emphasize objects' positions relative to each other. [Question] Answer the multiple choice question by only responding the letter of the correct answer. Please select the correct caption for the image:
(A) A bowl behind a cup
(B) A bowl to the left of a cup
(C) A bowl to the right of a cup
(D) A bowl in front of a cup

Figure 13: Example prompt for the What'sUp-B dataset.

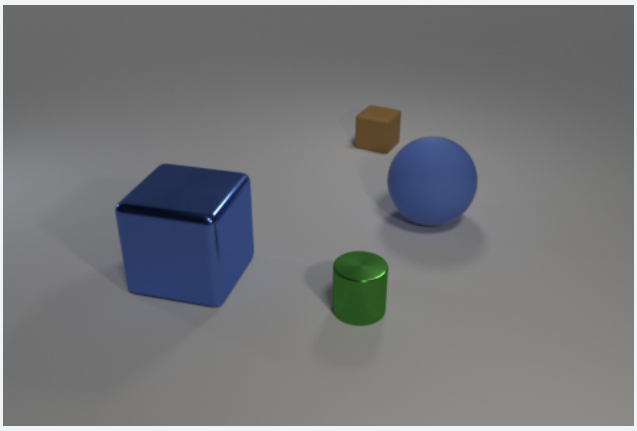
BLINK OBJECT LOCALIZATION

Image: 

[Model Prefix] answer en [Task Prefix] Answer the multiple choice question by only responding the letter of the correct answer. [Taxonomy Prefix] Emphasize objects' positions relative to each other. [Question] A bounding box is an annotated rectangle surrounding an object. The edges of bounding boxes should touch the outermost pixels of the object that is being labeled. Given the two bounding boxes on the image, labeled by A and B, which bounding box more accurately localizes and encloses the teddy bear? Select from the following options.
(A) Box A
(B) Box B

Figure 14: Example prompt for the BLINK Object Localization dataset.

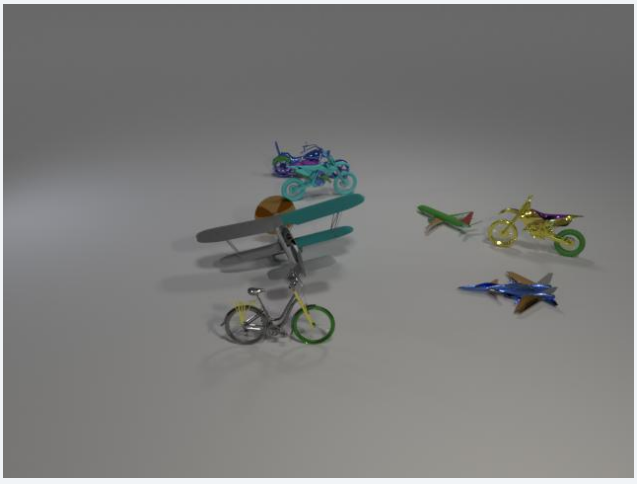
CLEVR

Image: 

[Model Prefix] answer en [Task Prefix] Answer the question by only responding the number. [Taxonomy Prefix] Prioritize counting objects and quantifying elements over other analysis. [Question] How many different items are there in the image?

Figure 15: Example prompt for the CLEVR dataset.

SUPER-CLEVR

Image: 

[Model Prefix] answer en [Task Prefix] Answer the question by only responding the number. [Taxonomy Prefix] Prioritize counting objects and quantifying elements over other analysis. [Question] How many different items are there in the image?

Figure 16: Example prompt for the Super-CLEVR dataset.

D Statistical Significance

In this section, we assessed the statistical significance of steering improvements on the What'sUp-A and CLEVR datasets, where we observe the most improvements of steering methods on spatial relation and counting tasks. We used a non-parametric bootstrap approach (10,000 samples) to derive 95% confidence intervals and computed p-values using McNemar's test. Results are presented in Table 4 and Table 5.

Intervention Method	Model	Statistical Significance			
		Improvement	95% CI	p-value	Significant?
Prompt	PaliGemma2-3B	-3.4%	[-5.6%, -1.4%]	0.00418	No
	PaliGemma2-10B	+2.5%	[-0.3%, 5.3%]	0.08326	No
	Idefics3-8B-Llama3	+1.4%	[-0.3%, 3.4%]	0.22656	No
SAE	PaliGemma2-3B	+9.2%	[5.9%, 12.8%]	<0.0001	Yes
	PaliGemma2-10B	+11.7%	[8.4%, 15.4%]	<0.0001	Yes
	Idefics3-8B-Llama3	+2.0%	[-0.3%, 4.2%]	0.14346	No
Probe	PaliGemma2-3B	+15.9%	[11.7%, 20.1%]	<0.0001	Yes
	PaliGemma2-10B	+3.1%	[1.1%, 5.0%]	0.00342	Yes
	Idefics3-8B-Llama3	+0.0%	[-2.5%, 2.5%]	1.00000	No
MeanShift	PaliGemma2-3B	+12.8%	[8.9%, 17.0%]	<0.0001	Yes
	PaliGemma2-10B	+6.4%	[3.6%, 9.5%]	<0.0001	Yes
	Idefics3-8B-Llama3	-0.3%	[-4.7%, 4.2%]	0.90418	No

Table 4: Statistical significance of improvements on the What'sUp-A dataset using bootstrap analysis (10,000 samples) and McNemar's test.

MeanShift demonstrates strong statistical reliability with significant improvements in 5/6 test cases across both datasets (the lower bound of the confidence interval is greater than 0), achieving substantial gains of up to +34.2% on CLEVR. SAE shows similar effectiveness with 5/6 significant improvements and particularly strong performance on CLEVR (+28.2% for Idefics3). Both methods consistently outperform prompting, which shows significance in only 2/6 cases with smaller effect sizes. These results confirm MeanShift and SAE as robust, effective techniques for enhancing visual understanding in MLLMs with high statistical confidence.

Intervention Method	Model	Statistical Significance			
		Improvement	95% CI	p-value	Significant?
Prompt	PaliGemma2-3B	+2.7%	[0.7%, 4.7%]	0.01431	Yes
	PaliGemma2-10B	+2.0%	[-0.2%, 4.2%]	0.08326	No
	Idefics3-8B-Llama3	+5.3%	[2.2%, 8.4%]	0.00109	Yes
SAE	PaliGemma2-3B	+18.2%	[14.4%, 22.0%]	<0.0001	Yes
	PaliGemma2-10B	+4.2%	[1.6%, 6.9%]	0.00235	Yes
	Idefics3-8B-Llama3	+28.2%	[24.0%, 32.4%]	<0.0001	Yes
Probe	PaliGemma2-3B	+4.0%	[1.6%, 6.7%]	0.00202	Yes
	PaliGemma2-10B	+0.9%	[-0.7%, 2.4%]	0.38770	No
	Idefics3-8B-Llama3	+24.7%	[20.7%, 28.7%]	<0.0001	Yes
MeanShift	PaliGemma2-3B	+14.7%	[11.1%, 18.4%]	<0.0001	Yes
	PaliGemma2-10B	+12.0%	[8.7%, 15.6%]	<0.0001	Yes
	Idefics3-8B-Llama3	+34.2%	[29.1%, 39.1%]	<0.0001	Yes

Table 5: Statistical significance of improvements on the CLEVR dataset using bootstrap analysis (10,000 samples) and McNemar's test.

E Computer Resources

All the experiments discussed in this paper can be done with **only one** NVIDIA A6000. For faster experiments, we use up to 8 NVIDIA A6000 to run experiments in parallel for various tasks and models.

F Broader Impact

The techniques described in this paper are meant to improve the performance of multimodal large language models (MLLMs). Such techniques are capable of being misused to the extent that MLLMs themselves can be misused. We stress that MLLMs and similarly our techniques do not have provable guarantees on the quality or safety of their outputs.

G Licenses

We list the licenses involved in this work as follows,

- `PaliGemma2` models and their backbone LLMs `Gemma2` are under license of *Gemma Terms of Use* https://ai.google.dev/gemma/terms.
- `Idefics3-Llama-8B` model is under the license of Apache license 2.0. Its language backbone, `Llama-3.1-8B` model, is under the license of *Llama 3.1 Community License Agreement*.
- `GemmaScope` pre-trained SAEs are under the license of *Creative Commons Attribution 4.0.*
- `LlamaScope` pre-trained SAEs are under the license of *Apache License 2.0.*
- `CV-Bench` is under the license of *Apache License 2.0*.
- `What'sUp` datasets are under the license of *MIT License*.
- `BLINK` dataset is under the license of the *Apache License 2.0*.
- `CLEVR` and `Super-CLEVER` datasets are under the *Creative Commons CC BY 4.0* license and the *BSD License*, respectively.
- Our usage of OpenAI's models for prompting is under the license of OpenAI's Terms of Service.