# Textual Steering Vectors Can Improve Visual Understanding in Multimodal Large Language Models

**Woody Haosheng Gan*** **Deqing Fu*** **Julian Asilis*** **Ollie Liu***
**Vatsal Sharan** **Robin Jia** **Willie Neiswanger**
University of Southern California
{woodygan, deqingfu, asilis, vsharan, robinjia, neiswang}@usc.edu, me@ollieliu.com

## Abstract

Steering methods have emerged as effective tools for guiding large language models' behavior, yet multimodal large language models (MLLMs) lack comparable techniques due to architectural diversity and limited availability of multimodal steering vectors. Inspired by this gap, we demonstrate that steering vectors derived solely from text-only LLM backbones can effectively guide and enhance their multimodal counterparts, revealing a novel cross-modal transfer that enables reuse of existing interpretability tools. Using community-standard methods—Sparse Autoencoders (SAE), Mean Shift, and Linear Probing—we validate this transfer effect across diverse MLLM architectures and visual reasoning tasks. Text-derived steering consistently enhances multimodal performance, with Mean Shift achieving up to +7.3% improvement in spatial relationship accuracy and +3.3% in counting accuracy on CV-Bench, and exhibits strong generalization to out-of-distribution datasets, for example reaching +34.2% on CLEVR counting tasks. This reveals that textual representations alone can effectively enhance visual grounding in MLLMs, bridging the mature ecosystem of text-based steering to MLLMs with minimal additional data collection or 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.*). However, similar steering approaches have not yet gained prominence for *multimodal large language models* (MLLMs). This is in part due to the heterogeneity of their architectures compared to text-only LLMs. Moreover, many steering methods assume access to a dataset of contrast pairs (Marks and Tegmark, 2023) to construct steering vectors, which are particularly difficult to obtain for multimodal inputs, as visual scenes inherently contain multiple overlapping concepts that are hard to isolate in a controlled manner.

Our key finding is that internal representations from a text-only LLM backbone retain their steering effectiveness even after multimodal adaptation. This transfer effect enables a new multimodal steering paradigm that is agnostic to architecture and does not require specialized multimodal data, and is broadly applicable to the dominant class of MLLMs constructed by augmenting a text-only LLM backbone with visual components (Liu et al., 2023; Dai et al., 2023; Bai et al., 2025; Wang et al., 2025; Yu et al., 2025). Importantly, it also allows us to directly repurpose steering methods originally developed for text-only models—such as Sparse Autoencoders (SAEs), Mean Shift, and Linear Probing—without modality-specific modifications. This bridges the mature ecosystem of text-based steering (McGrath et al., 2024; Durmus et al., 2024; Hanna et al., 2025) with the emerging space of multimodal models, providing a lightweight and interpretable pathway for enhancing multimodal reasoning.

Building on this insight, we propose a plug-and-play framework for multimodal steering. We extract steering vectors from text-only LLM backbones using established techniques and then apply them to the hidden states of their multimodal counterparts. This approach leverages the existing toolbox of steering methods, which have been extensively studied and evaluated in the text domain, to ensure accessibility and broader applicability for multimodal research. We evaluate our approach using different steering methods across multiple open-weight MLLMs and a broad suite of visual reasoning tasks, with Mean Shift performing best, achieving up to +7.3% improvement in spatial relationship accuracy in CV-Bench and

*Equal contribution.



*Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 40056–40087

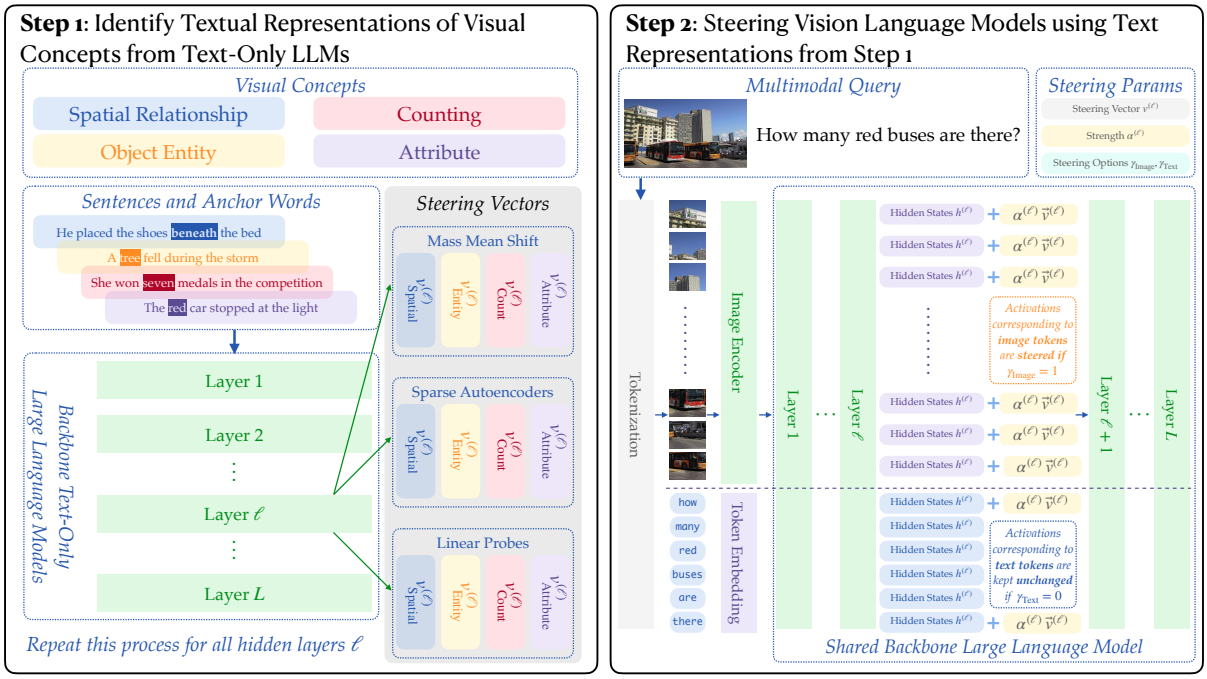



Figure 1: Overview of our steering methodology. Given an MLLM with a text-only LLM backbone and image-bound prompt, we first identify the required visual concept (*e.g*, spatial relationships). For each hidden layer $\ell$, we then extract corresponding steering vectors from the LLM using Mean Shift, Sparse Autoencoders, or Linear Probing. Finally, we apply these vectors to image tokens, text tokens, or both, controlled by parameters $\gamma_{\text{Image}}$ and $\gamma_{\text{Text}}$.

even stronger out-of-distribution gains, consistent with findings in text-only LLMs (Wu et al., 2025). Notably, while direct prompting is effective for steering *text-only* LLM behavior, it provides little benefit for multimodal reasoning. We also compare against LoRA fine-tuning: although LoRA achieves stronger in-distribution accuracy, it exhibits limited out-of-distribution generalization and lacks the lightweight and interpretability advantage of steering. Our contributions are as follows:

- We introduce a lightweight, plug-and-play multimodal steering method built directly on existing LLM representation-based techniques, requiring no gradient computations or parameter updates.
- We identify a novel 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 generalize to out-of-distribution test sets with no additional tuning, demonstrating significant performance gains.

## 2 Related Works

### 2.1 Multimodal Large Language Models

Multimodal large language models are commonly developed by endowing a backbone LLM with visual processing components and fine-tuning on multimodal datasets (Liu et al., 2023; Dai et al., 2023; Bai et al., 2025; Wang et al., 2025; Yu et al., 2025), with some exceptions pretrained from scratch (Chameleon Team, 2024; Team OLMo et al., 2024; Chen et al., 2025). Using an LLM backbone typically involves projecting outputs of an image encoder (Dosovitskiy et al., 2020; Zhai et al., 2023) to the same dimension as the 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. Our work specifically targets the dominant class of backbone-based MLLMs, as the shared text-only backbone enables direct reuse of existing interpretability tools such as GemmaScope and LlamaScope.

### 2.2 Representation-Based Steering

Methods for representation-based steering 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. Applying such methods to MLLMs remains less explored. VTI (Liu et al., 2025) constructs intervention vectors from paired multimodal inputs and applies them to visual and textual representations. Recent work explores gradient-based attribution for selective steering (Nguyen et al., 2025) and parameter-efficient training methods (Bi et al., 2025). These require multimodal contrast pairs or gradient computation. In contrast, we show that intervention vectors from text-only LLM backbones can steer MLLMs, enabling reuse of existing text-based methods (Durmus et al., 2024; Hanna et al., 2025) and revealing cross-modal transfer through preserved semantics (Lieberum et al., 2024b).

### 2.3 Shared Semantics

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. Recently, Papadimitriou et al. (2025) show that SAE features co-activate across multimodal inputs, while our work explores how such shared features can be exploited to steer MLLMs.

## 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

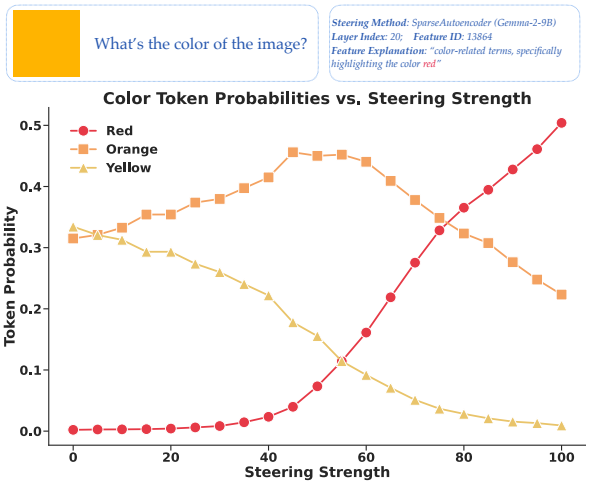



Figure 2: Effect of steering strength on color token probabilities.

#FFB400) and manipulate its perception by intervening in the hidden representations. Specifically, we obtain 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 $\alpha$ 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.

## 4 Methods

Building on our demonstration that textual representations can effectively steer visual understanding, we now explore systematic approaches to improve MLLMs' visual reasoning. Despite their growing success, MLLMs still struggle with seemingly simple visual queries—miscounting objects, confusing spatial relationships, and mishandling compositional prompts (Fu et al., 2024b). When the same problems are posed in pure text, foundation models perform far better (Fu et al., 2024a).

This motivates our central question: *Can existing steering mechanisms for textual representations rectify MLLMs' shortcomings?* A promising remedy is steering vectors: compact directions in activation space that encode specific concepts. By adding these vectors to hidden representations at

inference as $h'^{(\ell)}_{\text{target}} = h^{(\ell)}_{\text{target}} + \alpha \cdot v^{(\ell)}$, we amplify desired concepts without parameter updates, where $\ell^*$ and $\alpha^*$ are found via grid search. We extract $v^{(\ell)}$ from text-only LLM backbones using three established methods—Sparse Autoencoders, Mean Shift, and Linear Probing—demonstrating broad applicability of cross-modal transfer.

### 4.1 Dataset Construction for Steering Vector Extraction

To extract steering vectors for visual concepts from text, we identify four important taxonomies: spatial relationship, counting, attribute, and entity (Huang et al., 2023; Lin et al., 2024; Fu et al., 2025). For each visual concept, we curate small sets of sentence-anchor pairs, where each pair contains a sentence exhibiting the visual concept and the specific anchor phrase (one or more words) representing that concept. These sentence-anchor pairs serve as the foundation for all three steering vector extraction methods. Examples are provided in Table 3 in Appendix A.1.

### 4.2 Interpretable Steering Vector Extraction Methods

**Sparse Autoencoders (SAE).** Sparse Autoencoders reconstruct LLM activations using an MLP with sparsity penalties. Let $x = h^{(\ell)}(t) \in \mathbb{R}^D$ be activations for token $t$ at layer $\ell$. A SAE reconstructs $x$ as

$$\hat{x} = b_{\text{dec}} + \sum_{i=1}^{F} f_i(x)\, W^{\text{dec}}_{\cdot,i}, \tag{1}$$

where $b_{\text{dec}} \in \mathbb{R}^D$ and $W^{\text{dec}} \in \mathbb{R}^{D \times F}$ are learned decoder bias and weights. Feature activations are computed as

$$f_i(x) = \sigma\big(W^{\text{enc}}_{i,\cdot}\, x + b^{\text{enc}}_i\big) \tag{2}$$

using encoder weights $W^{\text{enc}} \in \mathbb{R}^{F \times D}$ and bias $b^{\text{enc}} \in \mathbb{R}^F$, where $\sigma$ is an activation function (*e.g.*, ReLU or JumpReLU).

The model minimizes the loss function:

$$L = \mathbb{E}_x \left[ \|x - \hat{x}\|_2^2 + \lambda \sum_{i=1}^{F} f_i(x)\, \|W^{\text{dec}}_{\cdot,i}\|_2 \right], \tag{3}$$

*i.e.*, $L_2$-reconstruction error and $L_1$-regularization on feature activations. Here, unit-normalized decoder weight vectors:

$$v^{(\ell)}_i := \frac{W^{\text{dec}}_{\cdot,i}}{\|W^{\text{dec}}_{\cdot,i}\|_2} \tag{4}$$

serve as feature directions, and

$$\alpha^{(\ell)}_i(t) := f_i\Big(h^{(\ell)}(t)\Big)\, \|W^{\text{dec}}_{\cdot,i}\|_2 \tag{5}$$

denotes the activation strength of $v^{(\ell)}_i$ on token $t$.

We leverage existing pretrained SAEs: GemmaScope (Lieberum et al., 2024b) for Gemma-2 and LlamaScope (He et al., 2024) for Llama-3.1-8B, leveraging existing interpretability infrastructure without training costs. Using our sentence-anchor pairs, we identify features with high activations on anchor phrases, verify their relevance to target visual concepts, and then average these relevant feature vectors to create a single steering vector for each visual concept per layer (details in Appendix A.1).

**Mean Shift.** This method identifies feature directions by computing activation differences (Figure 3), showing surprising effectiveness for LLM steering (Marks and Tegmark, 2023; Wu et al., 2025). For each taxonomy $\mathcal{T}$ and layer $\ell$, using sentence-anchor pairs $\{(s_1, w_1), \ldots, (s_K, w_K)\}$, we compute the mean shift vector:

$$m^{(\ell)}_{\mathcal{T}} = \frac{1}{K} \sum_{j=1}^{K} h^{(\ell)}(w_j) - \frac{1}{|\mathcal{S}_{\neg\mathcal{T}}|} \sum_{t \in \mathcal{S}_{\neg\mathcal{T}}} h^{(\ell)}(t), \tag{6}$$

where $h^{(\ell)}(w_j)$ is the residual stream activation of anchor phrase $w_j$ at layer $\ell$ and $\mathcal{S}_{\neg\mathcal{T}}$ is a control set of non-anchor tokens from the same sentences. We do not normalize the vector $m^{(\ell)}_{\mathcal{T}}$, preserving its magnitude relative to the original hidden states.

**Linear Probing.** We train a linear classifier distinguishing anchor phrase activations from control tokens at layer $\ell$ (Alain and Bengio, 2016; Park et al., 2024). As the hidden state dimensionality exceeds our sample size ($K < D$), we first project to dimension $d < K$ using PCA (we use $d = K/2$ in practice). With $Q \in \mathbb{R}^{d \times D}$ as the PCA matrix, the probe separates $\{h^{(\ell)}(w_j)Q^\top\}_{j \le K}$ and $\{h^{(\ell)}(t)Q^\top\}_{t \in \mathcal{S}_{\neg\mathcal{T}}}$, 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 anchor points) yields the steering vector:

$$v' = Q^\top v. \tag{7}$$

**Prompting Baseline.** Like our steering methods, prompting represents an interpretable approach that requires no parameter updates and has shown strong results in text-only domains (Wu et al.,

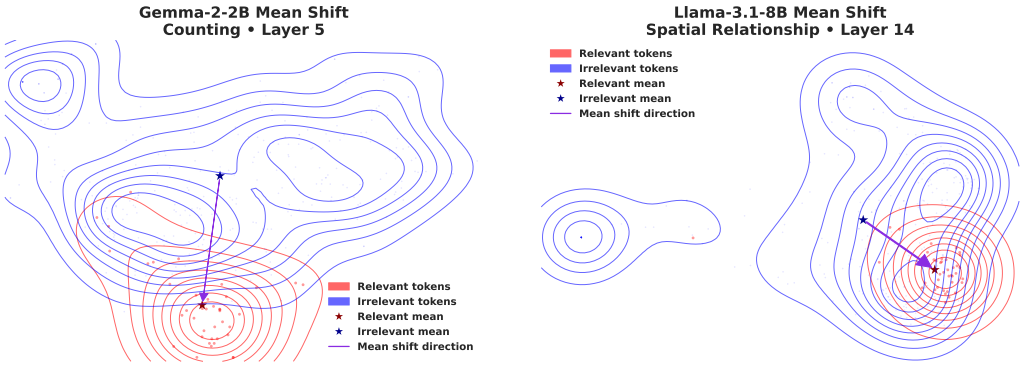
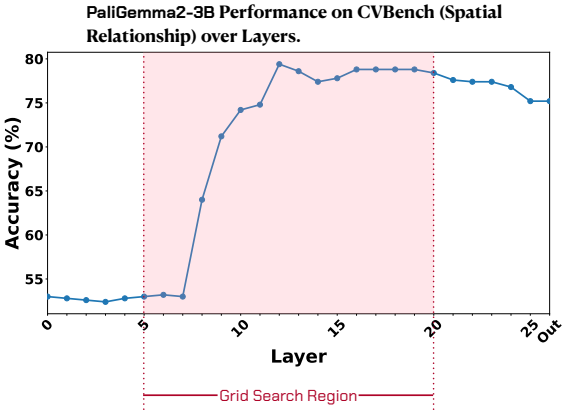
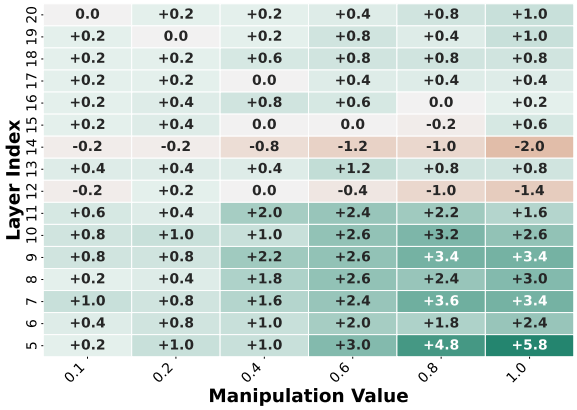



Figure 3: **Left:** Mean Shift method for counting features in `Gemma-2-2B`. The direction points from mean control token states to mean counting-related token states. **Right:** Spatial relationship features for `Llama-3.1-8B`. Activations projected to 2D for visualization.




(a) We zero out models' attention to image tokens after layer $\ell$ 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 $(\ell^*, \alpha^*)$ for steering the model's spatial reasoning abilities. In this case, $\ell^* = 5$ and $\alpha^* = 1.0$. Across all models and tasks, we observe that optimal layers $\ell^*$ tend to cluster in the lower third of the searched range, which practitioners can use as a heuristic to further narrow the search space.

Figure 4: Efficient Grid Search with `PaliGemma2-3B` on the Spatial Relationship Task.

2025). For a given taxonomy $\mathcal{T}$, we first curate a collection of 96 prompts of varying lengths by instructing `GPT-4o` to generate prompts that guide the model to reason with respect to $\mathcal{T}$, similar to the LLM-based prompt generation in AxBench (Wu et al., 2025), and then select the best-performing prompt via grid search on training data. Refer to Appendix A.2 for further detail.

## 5 Steering Improves Multimodal LLMs

Having established in Section 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 Section 4.2 can *improve* visual understanding in MLLMs when applied to intermediate representations.

### 5.1 Setup

**Models.** We primarily investigate PaliGemma2 models (`PaliGemma2-3B-mix-448` and `PaliGemma2-10B-mix-448`, referred to as `PaliGemma2-3B` and `PaliGemma2-10B`) and Idefics3-8B-Llama3. These models are selected because high-quality SAEs are available for their text-only backbones (`Gemma2-2B`, `Gemma2-9B`, and `Llama-3.1-8B`), enabling systematic comparison of all three steering methods. Architecturally, PaliGemma2 adopts prefix-LM masking where image tokens and textual instructions are cross-attended, while Idefics3 is fully autoregressive following LLaVA, allowing us to assess our approach across diverse fusion architectures.

**Dataset.** We use CV-Bench (Tong et al., 2024) with 4 sub-categories: `Count`, `Relation`, `Distance`, and `Depth`, totaling 2,638 data points.

Each sub-category contains around 700 samples, split into 500-600 training samples for grid search and 150 for testing.

**Grid Search.** We identify optimal injection layer $\ell \in \mathcal{L}$ and scale factor $\alpha \in \mathcal{A}$ via grid search on the training split. For each $(\ell, \alpha)$ pair, we intervene as $h'_{\text{target}}(\ell) = h_{\text{target}}(\ell) + \alpha v^{(\ell)}$ and select $(\ell^*, \alpha^*) = \text{argmax}_{\ell,\alpha} \text{Acc}(\ell, \alpha)$. We use $\mathcal{A} = \{0.1, 0.2, 0.4, 0.6, 0.8, 1.0\}$ for unnormalized vectors (Mean Shift). For normalized vectors (SAE, Probe), we use $\{10, 20, 30, 40, 50, 60\}$ on PaliGemma2 models and $\{0.2, 0.4, 0.6, 0.8, 1.0, 1.2\}$ on Idefics3 due to smaller hidden state norms. We set $\mathcal{L}$ to be the middle layers, where we observe the learning from image tokens is predominantly happening (see Figure 4a): $\{5, 6, \ldots, 20\}$ for `PaliGemma2-3B` and `Idefics3-8B-Llama3`, and $\{15, 16, \ldots, 30\}$ for `PaliGemma2-10B`. Notably, we never steer output tokens, focusing on internal representations.

### 5.2 Results

Table 1 presents a comparative analysis of three models 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, Mean Shift, and Prompting).

**Steering Interventions Prove Effective.** Table 1 demonstrates that steering interventions, especially Mean Shift, consistently improve model performance on spatial relationship and counting tasks over baseline levels. For instance, PaliGemma2-3B's "Relation" accuracy with Mean Shift rose from 76.0 to 83.3 using both tokens. We also observe that improvements tend to be larger for spatial relationship tasks compared to counting (e.g., +7.3 vs. +2.7 for PaliGemma2-3B with both tokens), likely because spatial relationships are more directly influenced by highlighting salient object features and positions, while counting demands more holistic scene interpretation, less directly aided by these steering methods. Notably, the 3B model shows larger absolute gains than the 10B model, which may reflect either lower baseline performance or greater sensitivity to activation interventions in smaller models.

**Mean Shift Shows Superior Performance.** Among the evaluated methods, Mean Shift performs most effectively and demonstrates more stable effects across different models, aligning with recent text-only steering findings (Wu et al., 2025). Mean Shift's superiority stems from its robustness: while SAE relies on learned sparse representations that may suffer from overfitting or incomplete concept capture, and probing operates in lower-dimensional space with sensitivity to specific projections, Mean Shift operates on full-dimensional representations using distributional properties, leading to more deterministic and stable effects.

**Prompting Barely Steers.** Table 1 indicates that prompting is often less effective than targeted interventions and sometimes even deleterious. This deviates from text-only observations (Wu et al., 2025), reflecting MLLMs' challenges in following fine-grained visual reasoning instructions. Unlike text-only models that reliably execute linguistic guidance, multimodal models may struggle with translating textual prompts into enhanced visual understanding, making prompting less effective.

**Intervention Transfers Across Tasks.** As shown in Figure 5, intervention using a feature $\mathcal{T}$ can transfer effectively to different tasks $\mathcal{T}'$. For instance, enhancing attribute and entity recognition improves spatial relationship performance, suggesting that accurate object identification helps spatial reasoning. This cross-task transfer reflects the interconnected nature of visual understanding, where strengthening one capability can have cascading benefits for related reasoning processes, and crucially, steering for one concept rarely causes notable degradation on unrelated tasks.

## 6 Steering Improvements Generalize Out-of-Distribution

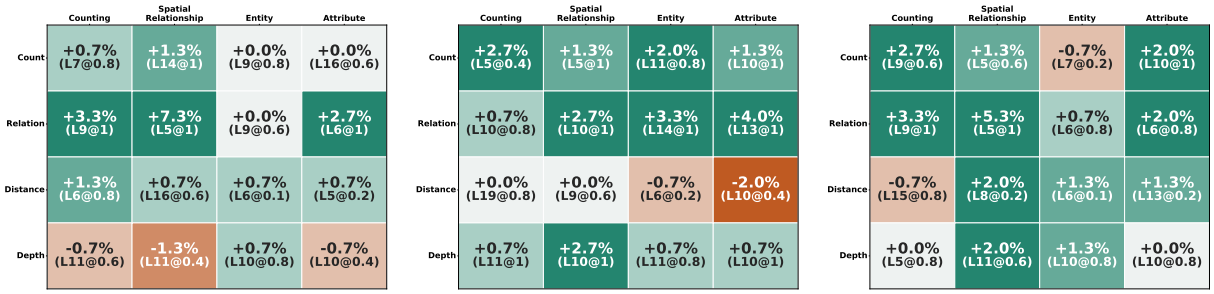
We now examine whether textual steering methods generalize out-of-distribution, *i.e.*, to datasets on which the steering method's hyperparameters $(\ell, \alpha)$ have not been tuned.

### 6.1 Setup

**Datasets.** We first evaluate on five datasets benchmarking isolated visual reasoning capabilities: What'sUp-A, What'sUp-B, BLINK Object Localization, CLEVR, and Super-CLEVR. What'sUp-A contains 412 images of pairs of household objects arranged in clear spatial relations of {"on", "under", "left", and "right"}, while What'sUp-B similarly contains 408 images with objects 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

| Model | Intervention Tokens | | Relation | | | Count | | |
|---|---|---|---|---|---|---|---|---|
| | Text | Image | SAE | Probe | MeanShift | SAE | Probe | MeanShift |
| | — | | 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) |
| PaliGemma2-3B | | ✓ | 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 | | 76.7 (+0.7) | | | 60.0 (+0.7) | | |
| | — | | 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) |
| PaliGemma2-10B | | ✓ | 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 | | 76.7 (−2.7) | | | 63.3 (+0.0) | | |
| | — | | 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) |
| Idefics3-8B-Llama3 | | ✓ | 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) | | | 58.7 (−0.7) | | |

Table 1: **Textual Steering Vectors Improve Multimodal LLMs' Visual Understanding**. Task-specific textual steering vectors reliably improve both spatial relation and counting performance across models. Stars (⋆) denote statistically significant improvements ($p < 0.05$, bootstrap test with 10,000 iterations).




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

Figure 5: Performance improvements on CV-Bench tasks when steering PaliGemma2-3B with Mean Shift 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.

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.

**Transfer Setup.** For each combination of test dataset, model, and steering method, we directly transfer the intervention configuration from CV-Bench without any tuning on the target dataset. Specifically, we use the $(\ell, \alpha)$ pair that performed best on the corresponding CV-Bench task category for that model and method. Steering vectors are aligned with each dataset's focus: "Spatial Relationship" vectors for What'sUp-A, What'sUp-B, and BLINK Object Localization, and "Counting" vectors for CLEVR and Super-CLEVR. Critically, neither the layer-scale hyperparameters nor the steering vectors themselves are adapted to the test datasets, making this a true out-of-distribution evaluation. Our prompting baseline similarly uses the exact prompt that performed best on the associated CV-Bench tasks. The only adaptation is using a small validation subset (50 datapoints for What'sUp and CLEVR, 25 for BLINK and Super-CLEVR) to determine the token type for intervention (image, text, or both).

### 6.2 Results

**Steering Remains Broadly Effective.** Table 2 demonstrates that textual interventions are effective across all 5 tasks, achieving average improvements over all models and datasets of at least +3.9 for all vector-based steerings. In contrast, prompting averaged only +0.8 and worsened performance in 5 cases, suggesting it's less effective for MLLMs

| Dataset | Visual Concept | Model | Intervention Method | | | | |
|---|---|---|---|---|---|---|---|
| | | | Baseline | Prompting | SAE | Probe | Meanshift |
| What'sUp-A | Spatial Relation | PaliGemma2-3B | 62.7 | 65.8 (+3.1)⋆ | 71.8 (+9.1)⋆ | 78.5 (+15.8)⋆ | 75.4 (+12.7)⋆ |
| | | PaliGemma2-10B | 68.5 | 63.3 (−5.2) | 80.1 (+11.6)⋆ | 71.6 (+3.1)⋆ | 74.9 (+6.4)⋆ |
| | | Idefics3-8B-Llama3 | 62.2 | 61.9 (−0.4) | 64.1 (+1.9) | 62.2 (+0.0) | 61.9 (−0.3) |
| | | Average Improvement | – | -0.8 | **+7.6** | +6.3 | +6.3 |
| What'sUp-B | Spatial Relation | PaliGemma2-3B | 60.6 | 56.7 (−3.9) | 58.9 (−1.7) | 57.5 (−3.1) | 60.3 (−0.3) |
| | | PaliGemma2-10B | 81.8 | 77.8 (−3.0) | 82.4 (+0.6) | 82.1 (+0.3) | 82.1 (+0.3) |
| | | Idefics3-8B-Llama3 | 52.0 | 57.3 (+5.3)⋆ | 56.2 (+4.2)⋆ | 57.0 (+5.0)⋆ | 63.4 (+11.5)⋆ |
| | | Average Improvement | – | -0.5 | +1.0 | +0.8 | **+3.8** |
| BLINK Object Localization | Spatial Relation | PaliGemma2-3B | 41.2 | 41.2 (+0.0) | 43.3 (+2.1) | 42.3 (+1.0) | 44.3 (+3.1)⋆ |
| | | PaliGemma2-10B | 51.6 | 52.6 (+1.0) | 54.6 (+3.1) | 53.6 (+2.1) | 57.7 (+6.2)⋆ |
| | | Idefics3-8B-Llama3 | 53.6 | 53.6 (+0.0) | 56.7 (+3.1)⋆ | 53.6 (+0.0) | 55.7 (+2.1) |
| | | Average Improvement | – | +0.3 | +2.8 | +1.0 | **+3.8** |
| CLEVR | Count | PaliGemma2-3B | 52.4 | 53.6 (+1.2) | 70.7 (+18.2)⋆ | 56.4 (+4.0)⋆ | 67.1 (+14.7)⋆ |
| | | PaliGemma2-10B | 70.7 | 72.4 (+1.7) | 74.9 (+4.2)⋆ | 71.6 (+0.9) | 80.4 (+9.8)⋆ |
| | | Idefics3-8B-Llama3 | 59.8 | 60.2 (+0.4) | 88.0 (+28.2)⋆ | 84.4 (+24.7)⋆ | 94.0 (+34.2)⋆ |
| | | Average Improvement | – | +1.1 | +16.9 | +9.9 | **+19.6** |
| Super-CLEVR | Count | PaliGemma2-3B | 26.9 | 30.3 (+3.4) | 32.0 (+5.1)⋆ | 30.3 (+3.4) | 33.1 (+6.3)⋆ |
| | | PaliGemma2-10B | 40.0 | 48.5 (+8.5)⋆ | 40.6 (+0.6) | 40.0 (+0.0) | 44.6 (+4.6)⋆ |
| | | Idefics3-8B-Llama3 | 66.5 | 65.7 (−0.8) | 66.5 (+0.0) | 67.5 (+1.0) | 68.5 (+2.0)⋆ |
| | | Average Improvement | – | +3.7 | +1.9 | +1.5 | **+4.3** |
| Average Improvement | | | – | +0.8 | +6.0 | +3.9 | **+7.6** |

Table 2: Performance of textual steering on out-of-distribution datasets. Stars (⋆) denote statistically significant improvements ($p < 0.05$, bootstrap test with 10,000 iterations).

than for text-only LLMs (Wu et al., 2025).

**Validation Against Linguistic Bias.** The improved performance on the What'sUp datasets provides evidence that our steering enhances genuine visual understanding rather than exploiting linguistic patterns. These datasets contain controlled image groups where identical objects are arranged in systematically varied spatial relationships (e.g., an apple positioned left, right, above, or below the same plate). If our methods were merely exploiting textual patterns, we would expect biased outputs regardless of visual content, rather than the observed accurate tracking of true spatial relationships.

**Superior OOD Performance on Focused Tasks.** Out-of-distribution performance often surpasses in-distribution results, particularly on datasets requiring "pure" reasoning abilities. For example, CLEVR, which isolates counting using simple geometric objects without complex object recognition, shows pronounced gains (+19.6 average), while CV-Bench Count and Super-CLEVR demand broader compositional understanding, resulting in more moderate improvements. This pattern suggests our steering precisely targets the intended cognitive capabilities.

**Mean Shift Demonstrates Consistent Superiority.** Across all experimental conditions, Mean Shift consistently outperforms other methods, achieving +7.6 average improvement compared to +6.0 for SAE and +3.9 for Probe. This mirrors results from CV-Bench and AxBench (Wu et al., 2025). Given this consistent effectiveness, we select Mean Shift for further evaluating on additional models without pretrained SAEs (Section 7 and Appendix C.2), demonstrating the generalization of our steering approach across diverse MLLMs.

The effectiveness of textual steering vectors across diverse MLLMs is further supported by our CKA analysis (Appendix D), which shows that backbone representations are substantially preserved after vision fine-tuning, particularly in the middle layers where steering is most effective.

## 7 Extended Evaluations

To assess broader applicability and practical implications, we further evaluate our steering approach on realistic multimodal tasks, additional models, and fine-tuning comparisons. Complete experimental details and results are in Appendix C.

**Real-World Task Performance.** Beyond isolated capabilities, we evaluated our steering methods on practical multimodal tasks including visual question answering (VQAv2 (Goyal et al., 2017)), open-ended image captioning (COCO Captions (Chen et al., 2015)), document understanding (DocVQA (Mathew et al., 2021)), chart reasoning (ChartQA (Masry et al., 2022)), and table understanding (VTabFact (Kim et al., 2024)).

Mean Shift achieved improvements in 15 out of 18 model-task combinations with 7 statistically significant gains. We note that modest improvements on these tasks are expected: unlike CLEVR or What'sUp in Section 6, which isolate the specific capabilities our vectors directly target, these holistic tasks simultaneously demand object recognition, OCR, commonsense reasoning, and domain knowledge—only a subset of which our spatial and counting vectors address. Despite this, the consistent positive impact across diverse applications confirms that enhancing core visual reasoning through steering yields tangible benefits even in complex, real-world settings. Full results are in Table 4 (Appendix C.1).

**Generalization to Additional Models.** Our primary experiments focused on models with high-quality pretrained SAEs (GemmaScope, LlamaScope) to enable systematic comparison of steering methods. Having identified Mean Shift as most effective, we validate its broader applicability on models without pretrained SAE infrastructure: InternVL3.5-1B and InternVL3.5-4B (Wang et al., 2025), Qwen3-VL-2B (Bai et al., 2025), and MiniCPM-V-4.5 (8B) (Yu et al., 2025). These models span parameter scales from 1B to 8B and employ different vision encoders and fusion mechanisms, providing a comprehensive testbed for evaluating cross-modal steering robustness. Mean Shift consistently improves performance across all four models, achieving +5.8 average improvement on CV-Bench and +6.7 on out-of-distribution datasets (Tables 5 and 6, Appendix C.2), consistent in magnitude with our primary results and demonstrating effectiveness across diverse MLLM designs.

**Comparison with Fine-Tuning.** We compared our steering approach against Low-Rank Adaptation (LoRA) (Hu et al., 2022) fine-tuning. While LoRA achieves stronger in-distribution performance (+10.8 average on CV-Bench), it shows limited OOD generalization (+1.7 average). In contrast, Mean Shift maintains consistent effectiveness across diverse datasets (+7.6 average OOD), reflecting fundamental differences in their mechanisms: LoRA adapts models to specific task distributions, while steering enhances underlying cognitive abilities that remain applicable across contexts, highlighting steering's superior generalization alongside its interpretability advantages. Moreover, steering requires no gradient computation or parameter updates, making it substantially more memory-efficient than LoRA.

## 8 Conclusion

We demonstrated that textual steering vectors extracted from text-only LLM backbones effectively enhance their multimodal counterparts' visual reasoning capabilities, and generalize robustly to out-of-distribution datasets. Our work bridges the mature ecosystem of text-based steering to MLLMs, providing an interpretable and efficient approach for enhancing visual reasoning without requiring multimodal contrast pairs or gradient computation.

## Limitations

While our approach demonstrates broad effectiveness, several limitations warrant consideration. First, our method's performance depends critically on the quality of extracted steering vectors—existing extraction methods, particularly SAE and Linear Probing, can produce vectors that inadequately represent target concepts, leading to variable performance across layers and models. Second, our evaluation reveals that steering provides strongest benefits on isolated reasoning benchmarks designed to test specific capabilities, while improvements on more realistic and holistic multimodal tasks are more modest (Appendix C.1). This limitation stems partly from our framework's requirement for manual selection of appropriate steering vectors for each task—we apply spatial vectors to What'sUp and counting vectors to CLEVR based on known task requirements. Real-world scenarios often demand multiple reasoning capabilities simultaneously, yet our current approach inteFrvenes on single concepts. Developing methods for automatic concept selection or dynamic multi-concept steering that can adaptively combine multiple vectors would significantly enhance practical utility and enable stronger performance on complex tasks. Third, while we demonstrate that textual representations preserve their effectiveness after vision-language fine-tuning and successfully transfer to visual understanding, we lack a deeper mechanistic understanding of why and how this cross-modal transfer occurs. Deeper investigation into the underlying mechanisms could reveal additional opportunities for more effective steering or identify potential failure modes. Fourth, our approach currently applies specifically to MLLMs constructed by augmenting a pretrained text-only LLM backbone with visual components—the dominant paradigm among current open-weight MLLMs (Liu et al., 2023; Dai

et al., 2023; Bai et al., 2025; Wang et al., 2025; Yu et al., 2025)—as this enables direct reuse of existing interpretability tools such as GemmaScope and LlamaScope; extending to models pretrained from scratch on multimodal data (Chameleon Team, 2024; Team OLMo et al., 2024; Chen et al., 2025) remains an open challenge. Future work should focus on developing more robust extraction methods, enabling automatic and compositional steering, uncovering the mechanistic basis of cross-modal representation transfer, and extending cross-modal steering beyond backbone-based MLLMs.

## Ethical Considerations

While our steering methods enhance visual reasoning on standard benchmarks, we acknowledge potential risks. Like any model modification technique, steering vectors could be misused to manipulate MLLM outputs in harmful ways. Specific concerns include: (1) inducing systematic misidentifications in safety-critical applications such as medical imaging, autonomous vehicles, or security systems; (2) exploiting cross-modal transfer to create targeted attacks where textual manipulations affect visual understanding in hard-to-detect ways; and (3) potential fairness implications if steering vectors trained on limited datasets fail to generalize equitably across diverse populations or visual contexts. However, we emphasize that our work advances MLLM interpretability for safety and alignment purposes. The risk profile does not exceed that of the underlying MLLMs themselves. We strongly encourage practitioners to thoroughly validate steering effects in high-stakes domains and consider societal impacts before deployment.

## 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. 2016. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*.

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

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, and 1 others. 2024. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*.

Jinhe Bi, Yujun Wang, Haokun Chen, Xun Xiao, Artur Hecker, Volker Tresp, and Yunpu Ma. 2025. Llava steering: Visual instruction tuning with 500x fewer parameters through modality linear representation-steering. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15230–15250.

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

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

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. 2015. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, and 1 others. 2024. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in neural information processing systems*, 36:49250–49267.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, G Heigold, S Gelly, and 1 others. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

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

Esin Durmus, Alex Tamkin, Jack Clark, Jerry Wei, Jonathan Marcus, Joshua Batson, Kunal Handa, Liane Lovitt, Meg Tong, Miles McCain, Oliver Rausch, Saffron Huang, Sam Bowman, Stuart Ritchie, Tom Henighan, and Deep Ganguli. 2024. Evaluating feature steering: A case study in mitigating social biases.

Deqing Fu, Ruohao Guo, Ghazal Khalighinejad, Ollie Liu, Bhuwan Dhingra, Dani Yogatama, Robin Jia, and Willie Neiswanger. 2024a. Isobench: Benchmarking multimodal foundation models on isomorphic representations. In *First Conference on Language Modeling*.

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

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

Leo Gao, Tom Dupre la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. 2025. Scaling and evaluating sparse autoencoders. In *The Thirteenth International Conference on Learning Representations*.

Gemma Team. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913.

Michael Hanna, Mateusz Piotrowski, Jack Lindsey, and Emmanuel Ameisen. 2025. circuit-tracer. https://github.com/safety-research/circuit-tracer. The first two authors contributed equally and are listed alphabetically.

Zhengfu He, Wentao Shu, Xuyang Ge, Lingjie Chen, Junxuan Wang, Yunhua Zhou, Frances Liu, Qipeng Guo, Xuanjing Huang, Zuxuan Wu, and 1 others. 2024. Llama scope: Extracting millions of features from llama-3.1-8b with sparse autoencoders. *arXiv preprint arXiv:2410.20526*.

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

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

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

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

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

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.

Yoonsik Kim, Moonbin Yim, and Ka Yeon Song. 2024. Tablevqa-bench: A visual question answering benchmark on multiple table domains. *arXiv preprint arXiv:2404.19205*.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. 2019. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMlR.

Hugo Laurençon, Andrés Marafioti, Victor Sanh, and Léo Tronchon. 2024. Building and better understanding vision-language models: insights and future directions. In *Workshop on Responsibly Building the Next Generation of Multimodal Foundational Models*.

Bruce W Lee, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Erik Miehling, Pierre Dognin, Manish Nagireddy, and Amit Dhurandhar. 2024. Programming

refusal with conditional activation steering. *arXiv preprint arXiv:2409.05907*.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023a. Inference-time intervention: Eliciting truthful answers from a language model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Zhuowan Li, Xingrui Wang, Elias Stengel-Eskin, Adam Kortylewski, Wufei Ma, Benjamin Van Durme, and Alan L. Yuille. 2023b. 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.

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

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

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

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

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

Sheng Liu, Haotian Ye, and James Zou. 2025. Reducing hallucinations in large vision-language models via latent space steering. In *The Thirteenth International Conference on Learning Representations*.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. 2024. Dora: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning*.

Llama Team at Meta. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

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

Samuel Marks, Can Rager, Eric J Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. 2025. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. In *The Thirteenth International Conference on Learning Representations*.

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

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. 2021. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209.

Thomas McGrath, Daniel Balsam, Myra Deng, and Eric Ho. 2024. Understanding and steering llama 3 with sparse autoencoders. *Goodfire Research*.

Lingchen Meng, Jianwei Yang, Rui Tian, Xiyang Dai, Zuxuan Wu, Jianfeng Gao, and Yu-Gang Jiang. 2024. Deepstack: Deeply stacking visual tokens is surprisingly simple and effective for lmms. *Advances in Neural Information Processing Systems*, 37:23464–23487.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Duy Nguyen, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. 2025. Grains: Gradient-based attribution for inference-time steering of llms and vlms. *arXiv preprint arXiv:2507.18043*.

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

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

Isabel Papadimitriou, Huangyuan Su, Thomas Fel, Sham Kakade, and Stephanie Gil. 2025. Interpreting the linear structure of vision-language model embedding spaces. *arXiv preprint arXiv:2504.11695*.

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

Zihan Qiu, Zekun Wang, Bo Zheng, Zeyu Huang, Kaiyue Wen, Songlin Yang, Rui Men, Le Yu, Fei Huang, Suozhi Huang, and 1 others. 2025. Gated attention for large language models: Non-linearity,

sparsity, and attention-sink-free. *arXiv preprint arXiv:2505.06708*.

Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeff Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Bloom, and 1 others. 2025. Open problems in mechanistic interpretability. *arXiv preprint arXiv:2501.16496*.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, and 1 others. 2024. 2 olmo 2 furious. *arXiv preprint arXiv:2501.00656*.

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, and 3 others. 2024. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*.

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. 2024. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *Preprint*, arXiv:2406.16860.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, and 1 others. 2025. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*.

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

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, and 1 others. 2025. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*.

Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. 2024. 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.

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

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

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Tianyu Yu, Zefan Wang, Chongyi Wang, Fuwei Huang, Wenshuo Ma, Zhihui He, Tianchi Cai, Weize Chen, Yuxiang Huang, Yuanqian Zhao, and 1 others. 2025. Minicpm-v 4.5: Cooking efficient mllms via architecture, data, and training recipe. *arXiv preprint arXiv:2509.18154*.

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

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, and 1 others. 2023. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.

# 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 these visual concepts. In practice, we set $K$ to 20. For each sentence $s_j$, we identify the anchor phrase for this visual concept as $w_j$, thus forming sentence-anchor pairs $(s_j, w_j)$. See table 3 for several examples.

Using our sentence-anchor pairs, we identify features with high activations on anchor phrases. Interestingly, as shown in Figure 6, we find that each visual concept activates only a limited number of SAE features, indicating a sparse encoding of these concepts. We then verify their relevance to the target visual concepts and average these relevant feature vectors to create a single steering vector for each visual concept at each layer.

We then use these sentence-anchor pairs 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 phrases $w_j$ in sentences $s_j$. At the second stage, we use `o3-mini` (OpenAI, 2025) to verify that these features indeed align with the desired visual concept $\mathcal{C}$. To accomplish the procedure, we use pretrained SAEs with detailed explanations and top activations developed by the interpretability community, such as `GemmaScope` (Lieberum et al., 2024b) for `Gemma-2-2B` and `Gemma-2-9B` (Gemma Team, 2024), and `LlamaScope` (He et al., 2024) for `Llama-3.1-8B` base model (Llama Team at Meta, 2024). 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.

**Algorithm 1** Find Textual Representations for Visual Concepts using SAEs

**Require:** Desired visual concepts $\mathcal{C}$. Layer index $\ell$.
**Require:** Sentence and anchor phrase pairs $\{(s_1, w_1), \cdots, (s_K, w_K)\}$.
**Require:** Pretrained SAEs at layer $\ell$.
▷ 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** $v^{(\ell)}$

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

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

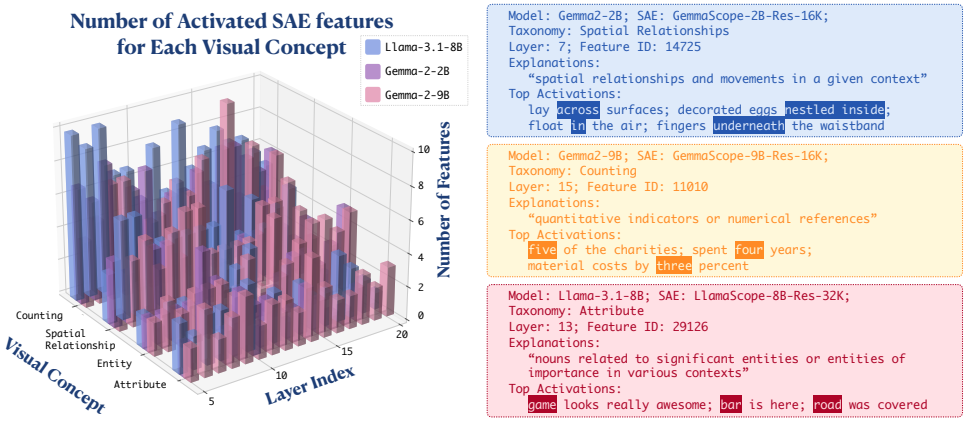



Figure 6: **Left**: Number of SAE features associated with each taxonomy (counting, spatial relationship, entity, and attribute) across the layers of Llama-3.1-8B, Gemma2-2B, and Gemma2-9B. Notably, SAE features for such visual concepts are sparse, numbering fewer than 10 across 16k total SAE features (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.

### A.2 Prompting

We now elaborate upon our generation of prompts for eliciting taxonomy-specific visual reasoning in MLLMs. As described in Section 4.2, we generate a total of 96 candidate prompts for each taxonomy $\mathcal{T}$. To do so, we use template shown in figure 8. Here, we set the `num instructions` to 6 and `word count` $\in \{5, 10, 15, \ldots, 80\}$, resulting in total $6 \times 16 = 96$ steering 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 Section 3.

These additional examples further support our findings in Section 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.

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).

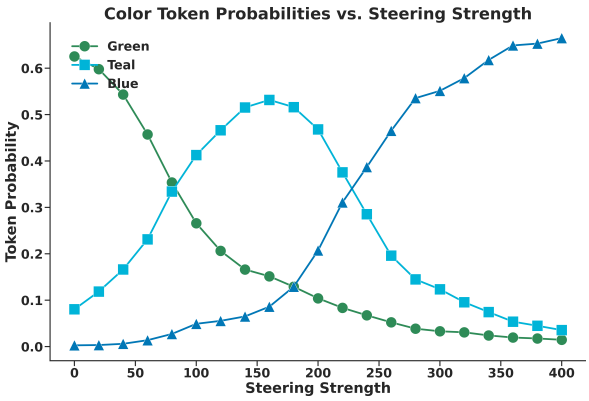
Your task:
- Generate {num_instructions} different instruction(5) that will make the Model focus on {concept} when answering questions about images
- Each instruction must be within {word_count} words
- Instructions should be direct and actionable, focusing specifically on how to emphasize {concept}

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

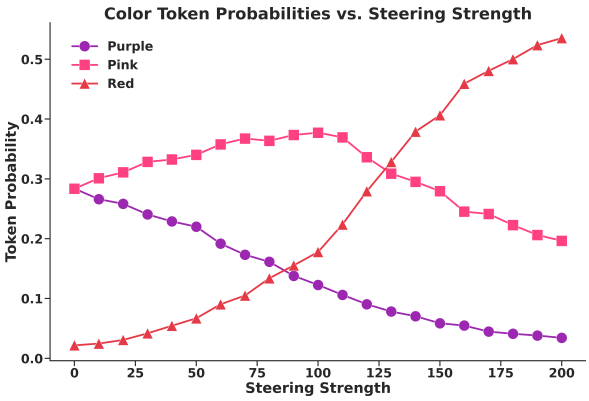
Example format for 2 instructions:
INSTRUCTION: First instruction text here within word limit.
INSTRUCTION: Second instruction text here within word limit.

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

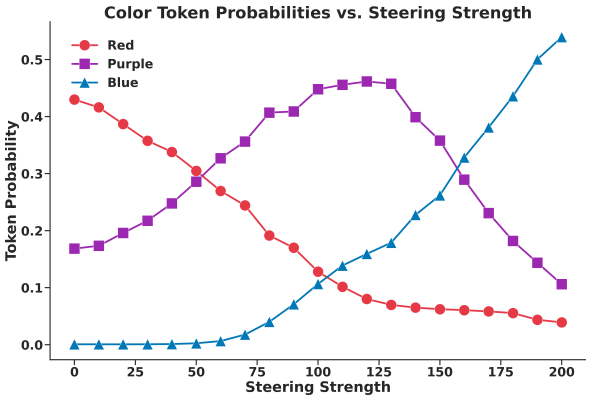
Figure 8: System and user prompt template for generating MLLM prompts.



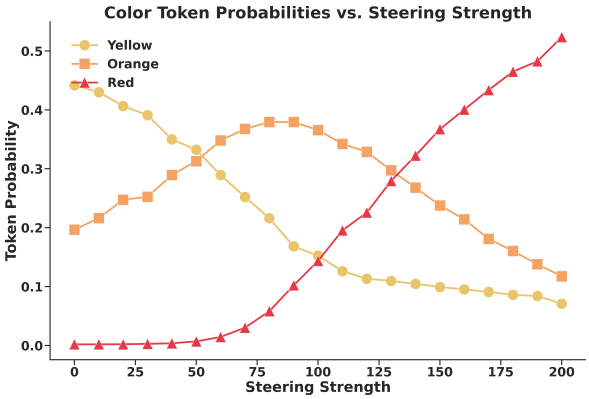

(a) Steering a green image toward blue perception. As the scale factor increases, the model's interpretation shifts from green to teal, and ultimately to blue.




(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.

## C Extended Experimental Results

### C.1 Real-World Task Evaluation

The datasets evaluated in Section 6 were specifically designed to benchmark isolated visual reasoning capabilities—spatial relationships and counting—making them ideal for controlled evaluation of our steering methods. To examine broader practical applicability, we further evaluated our cross-modal steering approach on real-world multimodal tasks that MLLMs encounter in practical applications.

**Experimental Setup:** We follow the identical protocol from Section 6.1, evaluating our steering methods on six real-world multimodal tasks using 500 examples per dataset for testing (200 for VTabFact due to dataset size limitations). Critically, we use the same $(\ell, \alpha)$ hyperparameters identified on CV-Bench (Section 5.1) without any tuning on these real-world datasets, making this a true test of cross-task generalization. We apply counting steering vectors to numerical reasoning tasks (DocVQA Number, ChartQA, VTabFact) and spatial relationship vectors to layout and captioning tasks (VQAv2, COCO Captions, DocVQA Layout). We use 50 validation examples per dataset only to determine the optimal intervention token type (image, text, or both).

**Task Details and Metrics:**

- **VQAv2 (Goyal et al., 2017):** General visual question answering task, evaluated using the official VQA Accuracy metric.
- **COCO Captions (Chen et al., 2015):** Open-ended image captioning task, evaluated using CIDEr-D metric.
- **DocVQA Layout (Mathew et al., 2021):** Document QA task focusing on spatial layout and structure questions, evaluated using ANLS×100.
- **DocVQA Number (Mathew et al., 2021):** Document QA task focusing on numerical information extraction, evaluated using ANLS×100.
- **ChartQA (Masry et al., 2022):** Chart interpretation and reasoning QA task, evaluated using the Relaxed Accuracy metric.
- **VTabFact (Kim et al., 2024):** Table reasoning multiple choice task, evaluated using accuracy.

**Results and Analysis:** Table 4 demonstrates that our steering methods achieve consistent effectiveness across diverse real-world applications. Mean Shift continues to demonstrate superior performance, producing improvements in 15 out of 18 model-task combinations with 7 statistically significant gains (marked with $\star$), remaining consistent with our main findings and confirming its effectiveness across both specialized and real-world tasks. Notably, PaliGemma2-3B shows particularly strong responsiveness to Mean Shift steering, achieving significant improvements on VQAv2 (+3.5), COCO Captions (+4.6 CIDEr-D), DocVQA Layout (+6.0), and VTabFact (+4.0).

The relatively modest improvement magnitudes compared to capability-focused benchmarks (Section 6) are expected and reflect the multi-faceted nature of these tasks. Unlike CLEVR or What'sUp, which isolate specific reasoning capabilities that our steering vectors directly target, real-world tasks require comprehensive abilities including object recognition, scene understanding, compositional reasoning, and domain-specific knowledge—only some of which directly benefit from our targeted spatial and counting interventions. For instance, VQAv2 requires not just spatial understanding but also common sense reasoning and object recognition, while COCO Captions demands diverse linguistic and compositional skills beyond spatial layout. Despite this, the consistent positive impact across diverse applications demonstrates that enhancing core visual reasoning capabilities through steering provides tangible benefits even in complex, holistic scenarios.

### C.2 Evaluation on Additional Models

Our primary experiments (Section 5) requires models with high-quality pretrained SAEs—GemmaScope for Gemma-2 and LlamaScope for Llama-3.1—to enable systematic comparison across all three steering methods (SAE, Probe, and Mean Shift). However, training SAEs requires substantial computational resources and SAE exists only for a limited set of models. Having identified Mean Shift as the most effective method (Section 6), we now evaluate it on additional MLLMs to validate the generalizability of our cross-modal steering approach. This demonstrates that our approach can be applied broadly across the MLLM landscape using only Mean Shift, without dependence on specialized SAEs.

**Models Evaluated:** We test the following models to validate broader applicability:

- **InternVL3.5-1B and InternVL3.5-4B (Wang et al., 2025):** Built on InternViT-

| Task | Visual Concept | Model | Intervention Method | | | | |
|---|---|---|---|---|---|---|---|
| | | | Baseline | Prompting | SAE | Probe | MeanShift |
| VQAv2 | Spatial Relations | PaliGemma2-3B | 86.8 | 87.0 (+0.2) | 88.2 (+2.4) | 87.1 (+0.3) | 89.3 (+3.5)* |
| | | PaliGemma2-10B | 88.2 | 86.8 (−1.4) | 87.4 (−0.8) | 86.9 (−1.3) | 88.9 (+0.7) |
| | | Idefics3-8B | 76.7 | 76.7 (+0.0) | 78.1 (+1.4) | 77.8 (+1.1) | 74.6 (−2.1) |
| | | Average Improvement | – | -0.4 | **+1.0** | +0.0 | +0.7 |
| COCO Captions | Spatial Relations | PaliGemma2-3B | 147.9 | 144.4 (−3.5) | 151.2 (+3.3)* | 151.0 (+3.1)* | 152.5 (+4.6)* |
| | | PaliGemma2-10B | 155.8 | 141.3 (−14.5) | 160.0 (+4.2)* | 161.1 (+5.3)* | 158.4 (+2.6) |
| | | Idefics3-8B | 70.0 | 70.3 (+0.3) | 71.2 (+1.2) | 70.9 (+0.9) | 69.6 (−0.5) |
| | | Average Improvement | – | -5.9 | +2.9 | **+3.1** | +2.2 |
| DocVQA Layout | Spatial Relations | PaliGemma2-3B | 79.4 | 81.4 (+2.0) | 84.8 (+5.4)* | 81.0 (+1.6) | 85.4 (+6.0)* |
| | | PaliGemma2-10B | 81.3 | 82.5 (+1.2) | 83.8 (+2.5) | 83.9 (+2.6)* | 83.8 (+2.5)* |
| | | Idefics3-8B | 88.5 | 86.3 (−2.2) | 89.6 (+1.1) | 88.2 (−0.3) | 89.7 (+1.3) |
| | | Average Improvement | – | +0.3 | +3.0 | +1.3 | **+3.3** |
| DocVQA Number | Counting | PaliGemma2-3B | 76.1 | 76.2 (+0.1) | 75.8 (−0.3) | 76.3 (+0.2) | 76.5 (+0.4) |
| | | PaliGemma2-10B | 77.7 | 75.8 (−1.9) | 77.6 (−0.1) | 79.4 (+1.7) | 76.9 (−0.8) |
| | | Idefics3-8B | 86.8 | 84.5 (−2.3) | 87.3 (+0.5) | 86.9 (+0.1) | 89.8 (+3.0) |
| | | Average Improvement | – | -1.4 | +0.0 | +0.7 | **+0.9** |
| ChartQA | Counting | PaliGemma2-3B | 46.4 | 45.4 (−1.0) | 46.6 (+0.2) | 47.4 (+1.0) | 48.0 (+1.6) |
| | | PaliGemma2-10B | 51.8 | 53.2 (+1.4) | 53.8 (+2.0) | 53.4 (+1.6) | 54.4 (+2.6)* |
| | | Idefics3-8B | 68.2 | 67.4 (−0.8) | 71.0 (+2.8)* | 67.2 (−1.0) | 72.6 (+4.4)* |
| | | Average Improvement | – | -0.1 | +1.7 | +0.5 | **+2.9** |
| VTabFact | Counting | PaliGemma2-3B | 56.5 | 54.5 (−2.0) | 58.0 (+1.5) | 56.0 (−0.5) | 60.5 (+4.0)* |
| | | PaliGemma2-10B | 57.0 | 58.5 (+1.5) | 58.5 (+1.5) | 59.0 (+2.0) | 58.5 (+1.5) |
| | | Idefics3-8B | 70.0 | 71.0 (+1.0) | 75.5 (+5.5)* | 71.0 (+1.0) | 73.5 (+3.5) |
| | | Average Improvement | – | +0.2 | +2.8 | +0.8 | **+3.0** |

Table 4: Performance of textual steering methods on real-world multimodal tasks. Stars ($\star$) denote statistically significant improvements ($p < 0.05$, bootstrap test with 10,000 iterations).

300M (Chen et al., 2024) vision encoder and Qwen3-0.6B/Qwen3-4B (Yang et al., 2025) language backbones (28 and 36 transformer layers respectively), connected via a two-layer MLP projector. The vision encoder processes images at dynamic resolutions using pixel unshuffle operations.

- **Qwen3-VL-2B (Bai et al., 2025):** Built on SigLIP2-So400M (Tschannen et al., 2025) vision encoder with DeepStack integration (Meng et al., 2024) and Qwen3-1.7B language backbone (28 transformer layers), connected via interleaved-MRoPE for spatial-temporal modeling. DeepStack leverages multi-level ViT features for vision-language alignment by fusing features from different Vision Transformer layers to capture both fine-grained details and high-level semantics.
- **MiniCPM-V-4.5 (8B) (Yu et al., 2025):** Built on SigLIP2-So400M (Tschannen et al., 2025) vision encoder and Qwen3-8B language backbone (36 transformer layers), connected via a unified 3D-Resampler using cross-attention with learnable queries. The resampler compresses visual features with 2D spatial and temporal positional embeddings (64 tokens per 448×448 image).

These models span parameter scales from 1B to 8B, employ different vision encoders (InternViT-300M, SigLIP2-So400M) and fusion mechanisms (MLP projector, interleaved-MRoPE, 3D-Resampler), providing a comprehensive testbed for evaluating cross-modal steering robustness.

**Experimental Setup:** We follow the protocol from Section 5.1, using Mean Shift to extract steering vectors from each model's text-only LLM backbone. However, we observe that these backbones exhibit attention sink behavior (Xiao et al.; Qiu et al., 2025), where the initial token disproportionately accumulates attention scores. To prevent this from biasing our steering vectors, we prepend a special token (<|im_start|>) to all text inputs and exclude it when computing Mean Shift vectors, ensuring the extracted directions capture genuine concept representations rather than attention sink artifacts.

We then perform grid search on the CV-Bench training split to identify optimal steering settings $(\ell^*, \alpha^*)$. We search over layers $\mathcal{L} = \{5, 6, \ldots, 20\}$ for InternVL3.5-1B and Qwen3-VL-2B, $\mathcal{L} = \{10, 11, \ldots, 25\}$ for InternVL3.5-4B and MiniCPM-V-4.5, and scale factors $\mathcal{A} = \{0.1, 0.2, 0.4, 0.6, 0.8, 1.0\}$ for all models. Using the optimal $(\ell^*, \alpha^*)$ pairs identified on the training

| Task | Intervention Tokens | | Model | | | | Average |
|---|---|---|---|---|---|---|---|
| | Text | Image | InternVL3.5-1B | Qwen3-VL-2B | InternVL3.5-4B | MiniCPM-V-4.5 | |
| | — | | 66.7 | 84.0 | 84.7 | 86.0 | 80.3 |
| CV-Bench Relation | ✓ | — | 71.3 (+4.7)⋆ | 85.3 (+1.3) | 88.0 (+3.3)⋆ | 87.3 (+1.3) | 83.0 (+2.7) |
| | — | ✓ | 73.3 (+6.7)⋆ | 88.7 (+4.7)⋆ | 88.7 (+4.0)⋆ | 90.0 (+4.0)⋆ | 85.2 (+4.8) |
| | ✓ | ✓ | 74.0 (+7.3)⋆ | 88.0 (+4.0)⋆ | 92.0 (+7.3)⋆ | 90.7 (+4.7)⋆ | 86.2 (+5.8) |
| | — | | 60.0 | 59.3 | 65.3 | 66.0 | 62.7 |
| CV-Bench Count | ✓ | — | 63.3 (+3.3) | 69.3 (+10.0)⋆ | 68.0 (+2.7) | 67.3 (+1.3) | 67.0 (+4.3) |
| | — | ✓ | 64.7 (+4.7)⋆ | 61.3 (+2.0) | 64.0 (−1.3) | 66.0 (+0.0) | 64.0 (+1.3) |
| | ✓ | ✓ | 66.7 (+6.7)⋆ | 66.7 (+7.3)⋆ | 69.3 (+4.0)⋆ | 71.3 (+5.3)⋆ | 68.5 (+5.8) |
| **Average Improvement (Both Tokens)** | | | **+7.0** | **+5.7** | **+5.7** | **+5.0** | **+5.8** |

Table 5: Mean Shift Steering on Additional Models (CV-Bench). Performance on CV-Bench test split using Mean Shift vectors extracted from each model's text-only backbone. The average column shows mean performance across all four models. The bottom row shows average improvement when intervening on both token types. Stars (⋆) denote statistically significant improvements ($p < 0.05$, bootstrap test with 10,000 iterations).

| Dataset | Method | Model | | | | Average |
|---|---|---|---|---|---|---|
| | | InternVL3.5-1B | Qwen3-VL-2B | InternVL3.5-4B | MiniCPM-V-4.5 | |
| What'sUp-A | Baseline | 74.0 | 98.6 | 92.0 | 92.3 | 89.2 |
| | MeanShift | 86.5 (+12.5)⋆ | 98.6 (+0.0) | 95.8 (+3.8)⋆ | 96.1 (+3.9)⋆ | 94.2 (+5.0) |
| What'sUp-B | Baseline | 69.3 | 88.3 | 82.1 | 89.4 | 82.3 |
| | MeanShift | 76.5 (+7.3)⋆ | 87.7 (−0.5) | 94.1 (+12.0)⋆ | 96.4 (+7.0)⋆ | 88.7 (+6.4) |
| BLINK Object Localization | Baseline | 51.5 | 56.7 | 57.7 | 51.5 | 54.4 |
| | MeanShift | 52.6 (+1.1) | 61.9 (+5.2)⋆ | 66.0 (+8.2)⋆ | 51.5 (+0.0) | 58.0 (+3.6) |
| *Spatial Reasoning Tasks Average Improvement: +5.0%* | | | | | | |
| CLEVR | Baseline | 9.3 | 96.7 | 79.6 | 90.2 | 69.0 |
| | MeanShift | 22.4 (+13.1)⋆ | 94.7 (−2.0) | 88.0 (+8.4)⋆ | 89.6 (−0.7) | 73.7 (+4.7) |
| Super-CLEVR | Baseline | 10.3 | 84.6 | 72.6 | 50.9 | 54.6 |
| | MeanShift | 33.7 (+23.4)⋆ | 89.1 (+4.6)⋆ | 81.1 (+8.6)⋆ | 62.3 (+11.4)⋆ | 66.6 (+11.9) |
| *Counting Tasks Average Improvement: +8.3%* | | | | | | |
| **Overall Average Improvement** | | **+11.5** | **+1.5** | **+8.2** | **+5.4** | **+6.7** |

Table 6: Out-of-Distribution Generalization with Additional Models. The rightmost column shows average performance across all four models. Rows with italics show category-specific averages. Stars (⋆) denote statistically significant improvements ($p < 0.05$, bootstrap test with 10,000 iterations).

split, we evaluate steering effectiveness on both the CV-Bench test split and our suite of OOD datasets.

**Results:** Tables 5 and 6 demonstrate that Mean Shift steering transfers effectively across diverse MLLM architectures. On CV-Bench (Table 5), Mean Shift achieves consistent improvements across all four models when intervening on both text and image tokens, with an average improvement of +5.8. Interestingly, we observe that smaller models tend to benefit more from steering interventions, mirroring our earlier findings in Section 5. The out-of-distribution evaluation (Table 6) reveals even stronger effectiveness, achieving an overall average improvement of +6.7 across all models and datasets without any hyperparameter tuning on target datasets. Notably, InternVL3.5-1B shows exceptional gains (+11.5 average), particularly on Super-CLEVR (+23.4) and CLEVR (+13.1), suggesting that smaller models may be more responsive to steering interventions. Interestingly, Qwen3-VL-2B shows more modest improvements (+1.5 average), largely attributable to its already strong baseline performance—it achieves 98.6 on What'sUp-A, 88.3 on What'sUp-B, and 96.7 on CLEVR without any intervention, leaving limited headroom for further improvement.

These results confirm that our cross-modal steering approach generalizes robustly on diverse MLLMs. The consistent effectiveness across models without pretrained SAEs demonstrates that Mean Shift can be applied broadly throughout the MLLM landscape, requiring only the text-only

| Task | Data Type | LoRA Performance | | | Average Improvement |
|---|---|---|---|---|---|
| | | PaliGemma-3B | PaliGemma-10B | Idefics-8B | |
| CVBench Relation | In-dist | 91.3 (+15.3)⋆ | 91.3 (+12.0)⋆ | 88.0 (+12.7)⋆ | +13.3 |
| CVBench Count | In-dist | 67.3 (+8.0)⋆ | 72.0 (+8.7)⋆ | 67.3 (+8.0)⋆ | +8.2 |
| Average In-Distribution | | +11.7 | +10.4 | +10.4 | **+10.8** |
| What'sUp-A | OOD | 67.7 (+5.0)⋆ | 69.3 (+0.8) | 61.6 (−0.6) | +1.7 |
| What'sUp-B | OOD | 58.4 (−2.2) | 86.0 (+4.2)⋆ | 58.1 (+6.1)⋆ | +2.7 |
| BLINK Object | OOD | 42.3 (+1.1) | 49.5 (−2.1) | 52.6 (+1.0) | +0.0 |
| CLEVR | OOD | 54.2 (+1.8) | 68.7 (−2.0) | 66.7 (+6.9)⋆ | +2.2 |
| Super-CLEVR | OOD | 28.6 (+1.7) | 43.4 (+3.4) | 66.9 (+0.4) | +1.8 |
| Average Out-of-Distribution | | +1.3 | +1.2 | +2.8 | **+1.7** |

Table 7: Performance comparison between LoRA and baseline models across in-distribution and out-of-distribution tasks. Stars (⋆) denote statistically significant improvements ($p < 0.05$, bootstrap test with 10,000 iterations).

LLM backbone for steering vector extraction.

### C.3 Comparison with LoRA Fine-Tuning

Beyond our interpretable steering methods, fine-tuning represents another common approach for enhancing model performance on specific tasks. To provide context for our steering approach and understand the trade-offs between different adaptation strategies, we compare against Low-Rank Adaptation (LoRA) (Hu et al., 2022) on both in-distribution and out-of-distribution tasks.

**Experimental Setup:** We trained LoRA adapters using the training split from our grid search procedure (Section 5.1) with an 80:20 train-validation split. We explored the following hyperparameter space:

- Rank: $r \in \{1, 2, 4\}$
- Alpha: $\alpha \in \{4, 8\}$
- Learning rate: $\eta \in \{1 \times 10^{-5}, 5 \times 10^{-5}, 1 \times 10^{-4}\}$
- Training epochs: 3
- Dropout: 0.1

We applied LoRA to the query and value projection parameters at the same layers used in our steering grid search: layers 5-20 for `PaliGemma2-3B` and `Idefics3-8B-Llama3`, and layers 15-30 for `PaliGemma2-10B`. Given the limited training data (500-600 samples), we restricted LoRA to these specific parameters and used small rank values to mitigate overfitting. For each model and task combination, we selected the hyperparameter configuration that achieved optimal validation performance.

**Results and Analysis:** Table 7 reveals complementary strengths between LoRA and our approach. LoRA achieves superior in-distribution performance on CV-Bench (+10.8 average). However, this advantage diminishes on out-of-distribution datasets, where LoRA achieves only +1.7 average improvement compared to Mean Shift's +7.6. This performance differential reflects fundamental differences in how these methods operate. LoRA fine-tuning adapts model parameters to align with specific task distributions, learning patterns or even shortcuts that may be dataset-specific. When the distribution shifts—even for tasks testing the same underlying capability—these learned adaptations may no longer apply. In contrast, steering methods enhance the model's internal representation of general cognitive abilities (spatial reasoning, counting) that remain applicable across diverse contexts and presentations. By operating on fundamental semantic representations rather than task-specific patterns, steering achieves more robust cross-domain transfer.

These results suggest that the choice between steering and fine-tuning depends on the deployment scenario. For applications with well-defined, stable data distributions where maximum performance is critical, LoRA fine-tuning may be preferable. However, for scenarios requiring robust generalization across diverse inputs, or where interpretability and understanding of model behavior is important, steering methods offer significant advantages. Moreover, steering requires no gradient computation or parameter updates, making it substantially more efficient at inference time.

Notably, these approaches are not mutually exclusive. Future work might explore combining steering with fine-tuning to achieve both strong

in-distribution performance and robust out-of-distribution generalization.

## D Representational Similarity Analysis

**Setup.** To provide mechanistic grounding for our cross-modal transfer effect, we measure layer-wise representational similarity between each text-only LLM backbone and its multimodal counterpart using Centered Kernel Alignment (CKA) (Kornblith et al., 2019). Given activation matrices $X \in \mathbb{R}^{n \times p}$ and $Y \in \mathbb{R}^{n \times q}$ collected from corresponding layers of two models on the same $n$ inputs, linear CKA is defined as:

$$\text{CKA}(X, Y) = \frac{\|Y^\top X\|_F^2}{\|X^\top X\|_F \|Y^\top Y\|_F}, \quad (8)$$

where $X$ and $Y$ are first mean-centered. CKA is invariant to orthogonal transformations and isotropic scaling, making it well-suited for comparing representations across independently trained models. A value of 1 indicates identical representational geometry; 0 indicates no alignment.

For each model pair, we extract mean-pooled hidden states from each transformer layer on the full validation set of WikiText-103 (Merity et al., 2016) using text-only inputs, and compute linear CKA between the resulting activation matrices. Results are shown in Figure 10.

**Results.** The degree of representational preservation varies across models, and correlates closely with how much the LLM backbone is updated during multimodal training. We identify two broad categories:

**Backbone frozen or lightly updated.** Idefics3, Qwen3-VL, and MiniCPM-V-4.5 all adopt training recipes that start with vision-language alignment before updating the LLM backbone. Specifically, early training stages focus on the newly initialized components—the projection layer and/or vision encoder—while keeping the LLM backbone frozen, ensuring that the backbone's pretrained representations are not disrupted during initial multimodal adaptation. Idefics3 freezes its backbone while training the projection layer first, before applying DoRA (Liu et al., 2024) to the backbone in later stages (Laurençon et al., 2024). Qwen3-VL similarly trains only the MLP merger during its initial alignment stage, leaving both the vision encoder and LLM backbone frozen (Bai et al., 2025). MiniCPM-V-4.5 first trains its 2D-Resampler module alone, then unfreezes the vision encoder, and only unfreezes the LLM decoder in the final stage (Yu et al., 2025). This approach is reflected in high CKA similarity of Idefics3 (mean $= 0.993$), and Qwen3-VL-2B (mean $= 0.970$). The relatively lower similarity for MiniCPM-V-4.5 (mean $= 0.727$) despite its frozen-backbone early stages likely reflects the end-to-end fine-tuning of all parameters in its final training stage, which introduces more representational shift in the backbone.

**Backbone jointly trained from the start.** PaliGemma2 updates all parameters jointly from the first pre-training stage without any frozen-backbone phase (Beyer et al., 2024), resulting in lower overall CKA similarity (mean $= 0.568$ and $0.542$ for the 3B and 10B variants respectively). InternVL3.5 similarly trains all parameters jointly (Wang et al., 2025), yet achieves substantially higher CKA (mean $= 0.903$ and $0.748$ for the 1B and 4B variants), likely because its pre-training corpus maintains a roughly 1:2.5 text-to-multimodal data ratio, which anchors backbone representations close to the original text-only model throughout training. For PaliGemma2, despite the lower global similarity, the middle layers where our steering intervention is applied remain relatively more similar than the global mean, and steering remains effective in practice as shown in Tables 1 and 2.

**Implications.** Across all models, the final layer exhibits the sharpest drop in similarity, reflecting its specialization toward multimodal next-token prediction. More importantly, our steering method remains effective across both categories. Even for PaliGemma2, where backbone representations are more shifted, steering consistently improves visual reasoning (Tables 1 and 2), and the color perception experiment (Section 3) directly demonstrates that textual feature directions integrate with visual understanding even after joint multimodal training. This suggests that vision fine-tuning preserves the directional semantic structure of backbone representations sufficiently for cross-modal steering transfer, even when overall representational similarity is moderate—providing a principled explanation for the cross-modal transfer effect observed throughout our experiments.

## E Dataset Evaluation Details

In this section, we explain in detail how we prompt and evaluate the model's performance across datasets and provide representative exam-

ples 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 COCO dataset specifically, we use "`caption en`", indicating that it is a captioning task. For other models, 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.`" for example. In `CLEVR` and `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 A.2, 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.

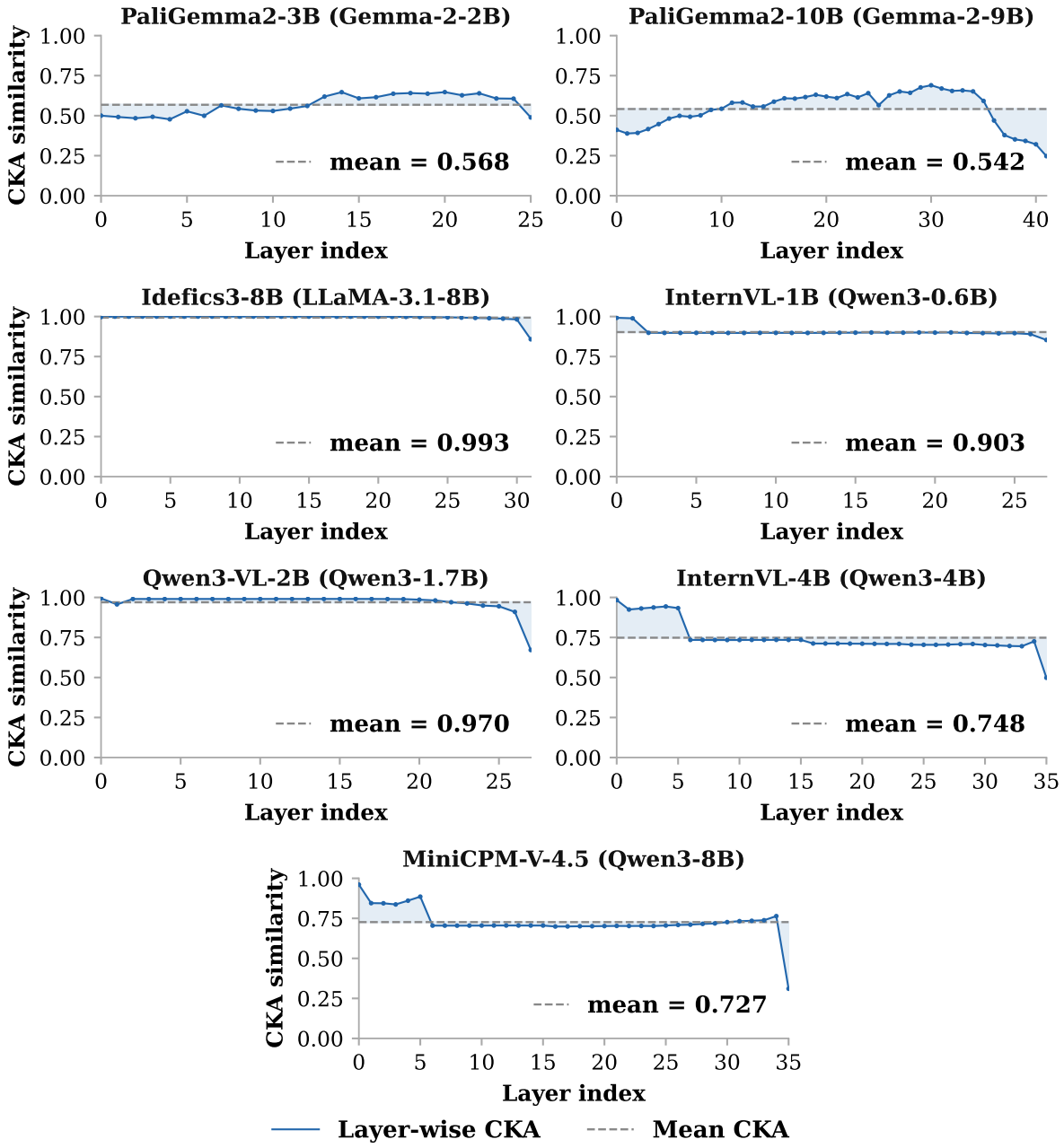



Figure 10: Layer-wise CKA similarity between each text-only backbone and its multimodal counterpart. Backbone name is shown in parentheses.

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 11: Example prompt for the CV-Bench Relation dataset.

CV-BENCH COUNT

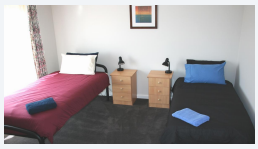


Image:
[Model Prefix] answer en
[Task Prefix] Answer the multiple choice question by only responding the letter of the correct answer. [Taxonomy Prefix] Prioritize counting objects and quantifying elements over other analysis. [Question] Answer the multiple choice question by only responding the letter 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 12: 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 13: 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 14: Example prompt for the What'sUp-B dataset.

BLINK Object Localization

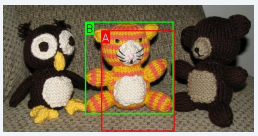


Image:
**[Model Prefix]** answer en
**[Task Prefix]** Answer the multiple choice question by only responding the letter of the correct answer. **[Taxonomy Prefix]** Emphasize objects' positions relative to each other. **[Question]** A bounding box is an annotated rectangle surrounding an object. The 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 15: Example prompt for the BLINK Object Localization dataset.

CLEVR

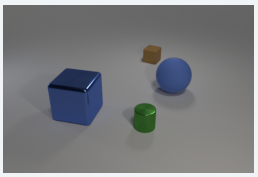


Image:
**[Model Prefix]** answer en **[Task Prefix]** Answer the question by only responding the number. **[Taxonomy Prefix]** Prioritize counting objects and quantifying elements over other analysis. **[Question]** How many different items are there in the image?

Figure 16: Example prompt for the CLEVR dataset.

Super-CLEVR

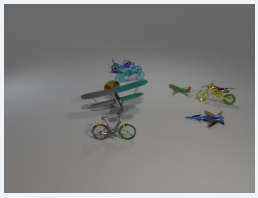


Image:
**[Model Prefix]** answer en **[Task Prefix]** Answer the question by only responding the number. **[Taxonomy Prefix]** Prioritize counting objects and quantifying elements over other analysis. **[Question]** How many different items are there in the image?

Figure 17: Example prompt for the Super-CLEVR dataset.

VQAv2

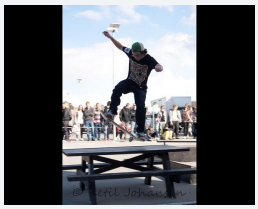


Image:
**[Model Prefix]** answer en **[Task Prefix]** Answer the question about the image. Provide a short, direct answer. **[Taxonomy Prefix]** Emphasize objects' positions relative to each other. **[Question]** Where is he looking?

Figure 18: Example prompt for the VQAv2 dataset.

## COCO

Image: **[Model Prefix]** caption en **[Task Prefix]** [blank for COCO] **[Taxonomy Prefix]** Emphasize objects' positions relative to each other. **[Question]** Generate a brief one-sentence caption.

Figure 19: Example prompt for the COCO dataset.

## DocVQA Layout


Image: **[Model Prefix]** answer en **[Task Prefix]** Answer the question about the image. Provide a short, direct answer. **[Taxonomy Prefix]** Answer the question based on the document. Provide a concise answer. **[Question]** What is the year mentioned at the top of the page?

Figure 20: Example prompt for the DocVQA Layout dataset.

## DocVQA Number

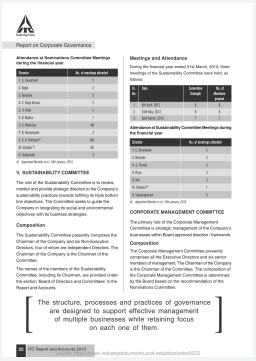


Image: **[Model Prefix]** answer en **[Task Prefix]** Answer the question about the image. Provide a short, direct answer. **[Taxonomy Prefix]** Prioritize counting objects and quantifying elements over other analysis. **[Question]** How many nomination committee meetings has S. Banerjee attended?

Figure 21: Example prompt for the DocVQA Number dataset.

## ChartQA

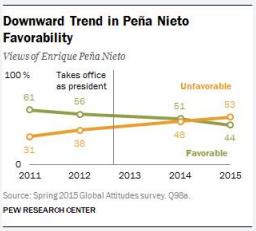



Image: **[Model Prefix]** answer en **[Task Prefix]** Answer the question based on the chart. Provide a concise answer. **[Taxonomy Prefix]** Prioritize counting objects and quantifying elements over other analysis. **[Question]** In which year the value was 51?

Figure 22: Example prompt for the ChartQA dataset.

VTABFACT

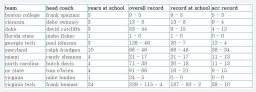


Image:

**[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]** ralph friedgen coach for 10 year at maryland

(A) Yes

(B) No

Figure 23: Example prompt for the VtabFact dataset.

## F Compute 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. The grid search in the main experiment takes ∼50 GPU hours in total.

## G Licenses

We list the licenses involved in this work as follows:

**Models:**

- `PaliGemma2` models and their backbone LLMs `Gemma2` are under the 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*.
- `InternVL3.5-1B` and `InternVL3.5-4B` models are under the *MIT License*.
- `InternViT-300M` vision encoder is under the *MIT License*.
- `Qwen3-VL-2B` and `Qwen3` language models (0.6B, 1.7B, 4B, 8B) are under the *Apache License 2.0*.
- `MiniCPM-V-4.5` (8B) model code is under the *Apache License 2.0*. Model weights require filling out a registration form for commercial use.
- `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*.

**Datasets:**

- `CV-Bench` is under the license of *Apache License 2.0*.
- `What'sUp` datasets (What'sUp-A and What'sUp-B) are under the *MIT License*.
- `BLINK` dataset is under the license of the *Apache License 2.0*.
- `CLEVR` dataset is under the *Creative Commons CC BY 4.0* license.
- `Super-CLEVR` dataset is under the *BSD License*.
- `VQAv2` dataset is under the *Creative Commons CC BY 4.0* license.
- `COCO Captions` dataset (annotations) is under the *Creative Commons CC BY 4.0* license. Images must comply with *Flickr Terms of Use*.
- `DocVQA` dataset can be downloaded from the RRC portal and requires agreement to their terms of use.
- `ChartQA` dataset is under the *MIT License*.
- `VTabFact` dataset license information should be verified from the original source.

**Other:**

- Our usage of OpenAI's models (GPT-4o and o3-mini) for prompting is under OpenAI's Terms of Service.

## H Intended Use and Compliance

All artifacts used in this work are employed consistent with their intended purposes as specified by their creators. Models are used for research evaluation of multimodal reasoning capabilities. Datasets are used for academic benchmarking and evaluation. All pre-trained components (SAEs, vision encoders, language models) are used within their documented use cases for interpretability and steering research. Our derived steering vectors are intended solely for research purposes in improving MLLM visual reasoning and should not be deployed in production systems without thorough validation.

## I Package Details

We implement our experiments using Python 3.10 with the following key packages:

- PyTorch 2.6.0 with torchvision 0.21.0 for model inference and training
- Transformers 4.51.3 (Hugging Face) for model loading and inference
- NumPy 1.26.4 for numerical computations

- Scikit-learn 1.6.1 for PCA and Linear Probing implementation
- SciPy 1.15.2 for statistical computations
- Datasets 2.21.0 (Hugging Face) for dataset loading and processing
- sae-lens 5.6.0 for Sparse Autoencoder usage