

Analyzing Fine-tuning Representation Shift for Multimodal LLMs Steering

Pegah Khayatan^{*1} Mustafa Shukor^{*1} Jayneel Parekh^{*1} Arnaud Dapogny¹ Matthieu Cord^{1,2}

¹ISIR, Sorbonne Université, Paris, France ²Valeo.ai, Paris, France

Abstract

*Multimodal LLMs (MLLMs) have reached remarkable levels of proficiency in understanding multimodal inputs. However, understanding and interpreting the behavior of such complex models is a challenging task, not to mention the dynamic shifts that may occur during fine-tuning, or due to covariate shift between datasets. In this work, we apply concept-level analysis towards MLLM understanding. More specifically, we propose to map hidden states to interpretable visual and textual concepts. This enables us to more efficiently compare certain semantic dynamics, such as the shift from an original and fine-tuned model, revealing concept alteration and potential biases that may occur during fine-tuning. We also demonstrate the use of shift vectors to capture these concepts changes. These shift vectors allow us to recover fine-tuned concepts by applying simple, computationally inexpensive additive concept shifts in the original model. Finally, our findings also have direct applications for MLLM steering, which can be used for model debiasing as well as enforcing safety in MLLM output. All in all, we propose a novel, training-free, ready-to-use framework for MLLM behavior interpretability and control. Our implementation is publicly available.*¹

1. Introduction

With the rapid progress in Large Language Models (LLMs) [7, 10, 28, 44, 65], Multimodal LLMs (MLLMs) [3, 12, 34, 39, 61] have recently demonstrated remarkable capabilities in addressing complex multimodal tasks such as image captioning and visual question-answering.

MLLMs are typically composed of a visual encoder, an LLM, and a connector. Following initial unimodal pretraining—and, in many cases, multimodal pretraining on large datasets—these models can be further specialized by training on multimodal datasets [1, 33]. Given the high compu-

tational cost of training these models, recent research has proposed more efficient approaches, such as creating diverse, high-quality instruction-tuning datasets [39] or keeping the LLM frozen and fine-tuning small amounts of parameters, like the connector [41, 57, 58, 67]. These approaches take advantage of the ability of frozen LLMs to generalize to multimodal data [56]. Despite the different efficient tuning methods, training these models still incurs significant costs.

While substantial progress has been made in developing high-performing MLLMs, relatively few studies aim to understand them [4, 47, 55, 56, 59, 61, 73]. Existing work typically conducts post-hoc analyses of MLLMs in isolation, overlooking the internal changes due to fine-tuning. Research by [56] addresses this gap to some extent by examining the internal multimodal alignment as it evolves during training.

In this work, we apply concept-level analysis to provide a readable understanding of MLLM behavior and, in particular, semantic dynamics that may occur due to fine-tuning, or due to covariate shift when considering different datasets.

In the first case, we find that fine-tuning on a specific task potentially reshapes learned latent concepts, with some adjusting subtly to align with the task, and others emerging or disappearing altogether (see Fig. 1). Notably, we find that most fine-tuned concepts can be reconstructed from the original model by translating its original concepts in the direction of specific *concept shift vectors*, reducing the need for additional training and its associated costs. Furthermore, we explore the implications of the proposed analysis for MLLMs steering, demonstrating how model outputs can be modified inexpensively without additional training. Our key findings are summarized as follows:

- We apply latent concept-level analysis to provide readable understanding on MLLMs’ behavior ; in particular, we show that fine-tuning can introduce significant alteration in the original concepts.
- We show that we can control the MLLM’s behavior w.r.t. certain concepts by simply manipulating shift vectors.
- Lastly, our findings also have direct applications for steering MLLM outputs, which find use for model debiasing as well as safety control.

^{*}First authors

¹Project page and code: <https://pegah-kh.github.io/projects/lmm-finetuning-analysis-and-steering/>

In a nutshell, we propose a novel, ready-to-use framework (including code) for MLLM behavior interpretability and control, debiasing and steering, which, we believe, will pave the way for future research.

2. Related Work

Concept-based explainability. Concept-based explainability methods have emerged as an alternative to traditional feature attribution based methods, that are capable of extracting key semantic features from the model internal representations. Most post-hoc concept-based approaches are based on the idea of concept activation vectors (CAV) [30], which represent concepts as vectors in the activation space. Instead of relying on human annotations, recent works have proposed methods to automatically discover concepts via clustering [21, 71] or matrix decomposition [18], which can be viewed as instances of a dictionary learning problem [17]. Initially focusing on understanding vision models, dictionary learning for concept extraction has been extended to LLMs e.g. using sparse autoencoders [27, 52]. However, none of the prior approaches have been applied to understand MLLMs, with the exception of recently proposed CoX-LMM [47].

MLLMs and Explainability. Multimodal LLMs [3, 34, 39, 67] have recently garnered significant interest. They typically adopt a late fusion architecture, and consist of an image encoder [19, 51, 72], a connector, and an LLM [28, 63, 65]. This family of models has inspired extensive research to better understand them and explain their behavior. For example, studies like [26, 45, 55] seek to identify multimodal neurons within LLMs or analyze modality-specific sub networks [56]. Some methods leverage the fact that these models are text-generative to simply generate textual explanations for model outputs [8, 20, 59, 70]. MLLMs benefit from in-context learning capabilities, which have been examined for limitations, including biases [4] and links to hallucinations [59], as well as the factors that may enhance their in-context learning performance [9, 49]. Related to our approach, CoX-LMMs [47] employs dictionary learning to extract multimodal semantic concepts from model representations. However, these studies typically assess models only in their final trained states, overlooking the dynamic changes that occur during training. Only limited works, such as [56], have investigated explaining changes due to fine-tuning, focusing specifically on implicit alignment between image and text modalities. In this work, we investigate how multimodal concepts within the model evolve throughout fine-tuning and explore the implications of these shifts on model steering.

Steering models with feature editing. In contrast to editing model weights, representation or feature editing methods [62, 66, 69] aim to modify model outputs without altering the model’s weights. A prominent approach within this family involves identifying steering vectors, or directions in the feature space (often within the residual stream), that are

linked to contrasting concepts. These methods have been applied to language models for various purposes, such as enhancing factuality or reducing hallucinations [46], inducing sentiment shifts or detoxification [64, 66], improving refusals to harmful requests [2], promoting truthfulness by modifying the output of attention heads [35], and erasing specific concepts or biases [6, 53]. However, their application to MLLMs is yet to be explored. Another set of approaches related to steering methods are based on In-context learning (ICL) [15, 16, 29], where prompts are carefully designed to induce desired behavior. Yet, ICL requires predefined demonstrations and lacks interpretability at a concept level. In contrast, our method offers lightweight steering, extending such capabilities to MLLMs without requiring any training, for instance as in ReFT [69].

3. Methodology Overview

Our framework is summarized in Fig. 1. We apply concept-level analysis of MLLM latent space (Section 3.1) to define and compare concepts between different setups. This allows us to monitor conceptual changes occurring during fine-tuning and manipulate MLLM behavior at a conceptual level (Section 3.2). Lastly, our framework finds applications for MLLM steering for e.g. debiasing and safety control (3.3) with negligible computational burden.

MLLM setup. A generic MLLM consists of a visual encoder f_V , a trainable connector C , and a language model f_{LM} . We assume that the model is pretrained on a multimodal (e.g. captioning) dataset $\mathcal{S} = \{(x_i, y_i)\}_i$, where $x_i \in \mathcal{X}$ represents images and $y_i \subset \mathcal{Y}$ are the associated captions specified as sequence of tokens from token vocabulary space \mathcal{Y} . The model is trained to generate the next text tokens, conditioned on text and images. The input to f_{LM} is a sequence of tokens that includes the concatenation of: (1) N_V visual tokens extracted from the image x via the visual encoder and connector ($C(f_V(x))$), and (2) linearly embedded textual tokens corresponding to the text instruction and previously predicted tokens. This can be expressed as:

$$\hat{y}^p = f_{LM}(h^1, \dots, h^{N_V}, \dots, h^p),$$

where $h^1, \dots, h^{N_V} = C(f_V(x))$, and $h^p = \text{Emb}(\hat{y}^{p-1})$, with Emb representing the token embedding layer. During generation, the output token \hat{y}^p is derived by normalizing the last layer (L) tokens $h_{(L)}^p$, then applying the unembedding layer W_U and a softmax operation. The model keeps predicting the next token until the end of the sentence token to obtain the generated response $\hat{y} = \{\hat{y}^p\}_{p > N_V + N_I}$, where N_I corresponds to the text instruction.

3.1. Retrieval and comparison of latent concepts

To understand the internal representations of any given MLLM f , we leverage the approach introduced in [47].

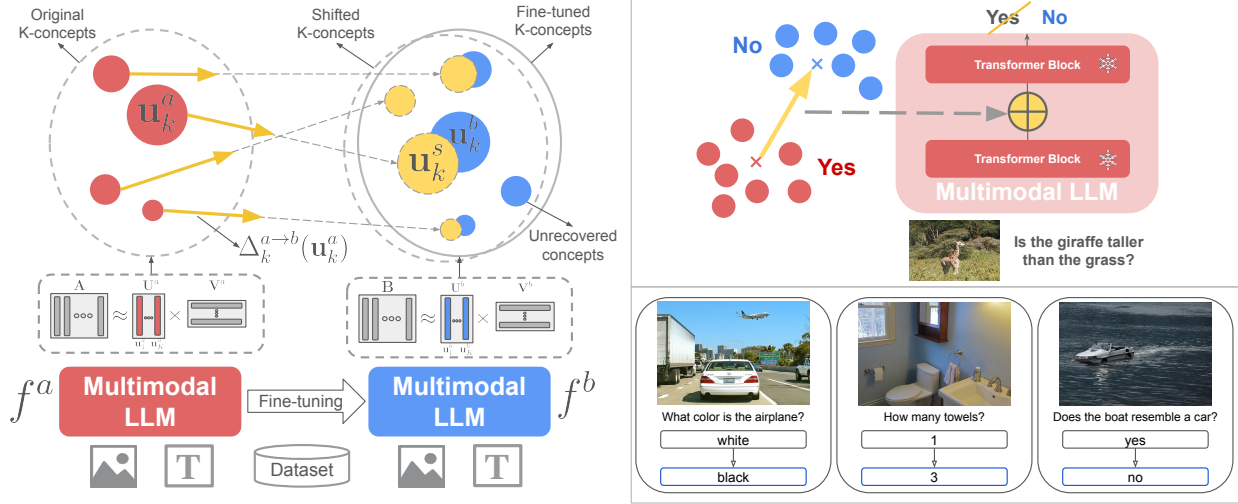


Figure 1. **Framework overview.** We apply concept-level analysis for MLLM behavior monitor and control, for (left) understanding and manipulating (through *shift vectors*) concept changes due to fine-tuning, as well as (right) MLLM steering for debiasing or safety control.

Specifically, given a set of M images $\{x_1, \dots, x_M\}$ we extract a set of residual stream representations from some layer l of the MLLM f . These representations $z_m = f_l(x_m) \in \mathbb{R}^D$ (one per image) are collected in a feature matrix $\mathbf{Z} \in \mathbb{R}^{D \times M}$. Typically, the set of images and extracted representations correspond to a particular token of interest *TOI* (e.g., ‘Dog’, ‘Cat’, ‘Person’, etc.) in the predicted caption. However, the extraction can also be performed for a larger set of target tokens. This feature matrix \mathbf{Z} is then decomposed as $\mathbf{Z} \approx \mathbf{U}\mathbf{V}$ to recover the concepts in the latent embedding space. Here, $\mathbf{U} \in \mathbb{R}^{D \times K}$ is the matrix of K concepts and $\mathbf{V} \in \mathbb{R}^{K \times M}$ represents the coefficients/activations of the samples projected onto these concepts. Different decompositions of the matrix K result in various concepts, inheriting the properties of the decomposition (such as low grounding overlap with PCA). We employ K -Means to learn our concept dictionaries. This is motivated by K -Means’ simplicity, and straightforward arithmetic manipulation of the clusters/concepts it allows. Each column $\mathbf{u}_k \in \mathbf{U}$ corresponds to a concept, while each column of \mathbf{V} encodes the activation of these concepts for a given sample. Note that any given representation $f_l(x)$ can be projected on \mathbf{U} to obtain its activation vector $\mathbf{v}(x) \in \mathbb{R}^K$, i.e. $f_l(x) \approx \mathbf{U}\mathbf{v}(x)$. Each extracted concept is then interpreted through grounding in both image and text spaces. Specifically, the top N_{MAS} that activates concept \mathbf{u}_k the most represent its image grounding:

$$\mathbf{X}_{\text{MAS}}(\mathbf{u}_k) = \arg \max_{\hat{X} \subset \mathbf{X}_t, |\hat{X}|=N_{\text{MAS}}} \sum_{x \in \hat{X}} |\mathbf{v}_k(x)|, \quad (1)$$

where $\mathbf{v}_k(x)$ refers to the the activation of \mathbf{u}_k for image x . For text grounding, we decode the features using the unembedding matrix of the language model W_U [5, 32, 43, 54]. Specifically, the operation $W_U \mathbf{u}_k \in \mathbb{R}^{|\mathcal{V}|}$ produces

logits over the vocabulary, and the top $N_{\text{grounding}}$ words with highest logits are extracted:

$$\mathbf{T}_{\text{words}}(\mathbf{u}_k) = \arg \max_{\text{Top-}N_{\text{grounding}}} (W_U \mathbf{u}_k). \quad (2)$$

Finally, to quantify the similarity of two concepts (i.e. the columns of \mathbf{U} we define the *Text Grounding Overlap* as:

$$\text{T-Overlap}(\mathbf{u}, \mathbf{u}') = 100 \times \frac{|\mathbf{T}_{\text{words}}(\mathbf{u}) \cap \mathbf{T}_{\text{words}}(\mathbf{u}')|}{|\mathbf{T}_{\text{words}}(\mathbf{u})|}. \quad (3)$$

Now that we have defined a generic framework, we present two subcases for extracting and manipulating concepts.

3.2. Evolution of concepts through fine-tuning.

Setup overview. An original model f^a is typically fine-tuned to produce a specialized model f^b for a particular task—or, specifically, for a set of target concepts. This fine-tuning can be conducted on samples that include a set of words $\{w_1, \dots, w_m\}$ associated with these target concepts. For instance, if we fine-tune a image captioning model to emphasize colors in the image, the set of words will simply be these colors. Efficient fine-tuning is typically achieved using Low-Rank Adaptation (LoRA) [25, 34, 39]. Fine-tuning can selectively alter certain representations, leading to shifts in the conceptual space encoded by the model. Using the interpretability framework discussed in Section 3.1 we can study these shifts at a readable conceptual level.

Concept recovery via shift vectors. To study the change from an original model f^a to a finetuned model f^b , we fix the dataset $S^{(1)} = S^{(2)}$, and obtain two sets of embeddings from f^a, f^b respectively, i.e. $A \approx U^a V^a, B \approx U^b V^b$, where $U^a, U^b \in \mathbb{R}^{D \times K}$ are K concepts extracted from each model. We propose to characterize the concept changes

from an original to fine-tuned model as linear directions in embedding space or *concept shift vectors*. To do so, we first associate each original concept $\mathbf{u}_k^a \in \mathcal{U}^a$ with a subset of samples where \mathbf{u}_k^a is the most activated concept:

$$\mathbf{A}_k = \{m \mid k = \arg \max_i |\mathbf{v}_i^a(x_m)|\}.$$

For each sample $x_m, m \in \mathbf{A}_k$ we define $\delta_m^{a \rightarrow b} = \mathbf{b}_m - \mathbf{a}_m$ as the change in its representation from f^a to f^b . To compute the concept shift vector $\Delta_k^{a \rightarrow b}(\mathbf{u}_k^a)$ associated with \mathbf{u}_k^a , we aggregate shifts of its associated samples specified by \mathbf{A}_k :

$$\Delta_k^{a \rightarrow b}(\mathbf{u}_k^a) = \frac{1}{|\mathbf{A}_k|} \sum_{m \in \mathbf{A}_k} \delta_m^{a \rightarrow b} = \frac{1}{|\mathbf{A}_k|} \sum_{m \in \mathbf{A}_k} (\mathbf{b}_m - \mathbf{a}_m)$$

The concept shift vector is used to shift each concept in the original model \mathbf{u}_k^a to obtain the shifted concept \mathbf{u}_k^s :

$$\mathbf{u}_k^s = \mathbf{u}_k^a + \alpha \Delta_k^{a \rightarrow b}(\mathbf{u}_k^a), \quad (4)$$

where α is a coefficient to control the shift magnitude. Unless otherwise stated, we use $\alpha = 1$ as the default magnitude of shift. It is worth noting that given the concept shift vectors, the computation of shifted concepts does not rely on accessing the fine-tuned model. Practically speaking, this means that we can "push" the original model towards the concepts of a fine-tuned one with very little overhead, by simply shifting its latent representation in the direction of the shift vector, as it will be illustrated in the experiments.

3.3. Concept evolution across datasets and applications to model steering

Concept comparison between different datasets. Another interesting subcase of the proposed framework consists in evaluating the shift from one dataset $S^{(1)}$ to another $S^{(2)}$, using the same model and encoding h . Beyond interpretability of a model behavior, it also find applications for model steering. Model steering (see Fig. 1 (right)) refers to guiding the model outputs towards desired outcomes by modifying the features without altering the model weights.

Coarse-grained model steering. In coarse-grained or global steering, the objective is to adjust the model outputs \hat{y} to generally align with a set of target samples (e.g., changing answers type). Given input-output samples, we first extract the answer representations $\mathbf{B} = \mathbf{b}_1, \dots, \mathbf{b}_N$ at layer l from the target set. Similarly, we obtain representations for the original set $\mathbf{A} = \mathbf{a}_1, \dots, \mathbf{a}_M$ (e.g. randomly drawn from the train set). We then compute the coarse steering vector \mathbf{s}_c as:

$$\mathbf{s}_c = \frac{\sum_i^N \mathbf{b}_i}{N} - \frac{\sum_i^M \mathbf{a}_i}{M}, \quad (5)$$

\mathbf{s}_c is applied to all the samples in the validation set. For instance, the activations $f_i(x_i)$ of a sample x_i become:

$$\tilde{f}_i(x_i) = f_i(x_i) + \alpha \mathbf{s}_c \quad (6)$$

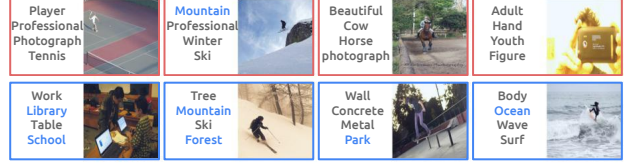


Figure 2. **Concepts extracted from original and fine-tuned models.** concepts from the original f^a (top), and model fine-tuned to focus more on places f^b (bottom), for $TOI = person$. The concepts from f^b exhibit a stronger association with places.

where α controls the steering strength and it is set to 1 (we study α in App. B.5.3). Thus, in this setup, all examples are coarsely steered in the direction of the steering vector \mathbf{s}_c at layer l , before passing $\tilde{f}_i(x_i)$ through the rest of layers and $\tilde{f}_i(x_i)$ becomes the input to the next layer $l + 1$.

Fine-grained steering. Unlike global steering, fine-grained steering consists in finding and editing directions that adjust only certain concepts to other ones. To do this, we decompose the hidden states of a set of samples into a set of concepts \mathcal{U} as previously explained. We then compute a set of fine-grained steering vectors $\mathbf{s}^f = \mathbf{s}_{11}^f, \dots, \mathbf{s}_{NN}^f$, with $\mathbf{s}_{ij}^f = \mathbf{s}_{ij}^f = \mathbf{u}_j - \mathbf{u}_i$ the steering vector from concept \mathbf{u}_i to \mathbf{u}_j . However, not all steering vector are meaningful: options for finding the relevant ones include proximity matching, as well as identifying vectors that have the strongest impact on guiding the model towards generating specific answers or concepts (e.g. producing significantly more target answers). This is more detailed in App. B.3. Various applications of MLLM steering can be explored, such as gender debiasing and safety alignment, as it will be shown below.

4. Experiments

4.1. Fine-tuning experiments

In this section, we study how fine-tuning introduces changes in the overall structure of the learned concepts in MLLMs (with the architecture described in Section 3).

Implementation details. The main paper covers experiments on the popular LLaVA [38] model comprising a CLIP image encoder, a two-layer MLP connector, and a 7B Vicuna-1.5 LLM. More experiments on a different multimodal model can be found in App. A. We conduct our study in a controlled setup that consists in specializing the model on a target dataset. Specifically, we apply fine-tuning on three different subsets of the Visual Genome dataset [31], related to places, colors, and sentiments (more details in App. A.2).

Impact of fine-tuning on learned concepts. Fig. 2 depicts this concept change. Through the concepts groundings for an exemple $TOI (person)$ we see that the fine-tuned model puts a stronger emphasis on places, which is expected and serves as a sanity check for our method.

Matched concepts. To further analyze how each concept changes after fine-tuning, we focus on each concept and its

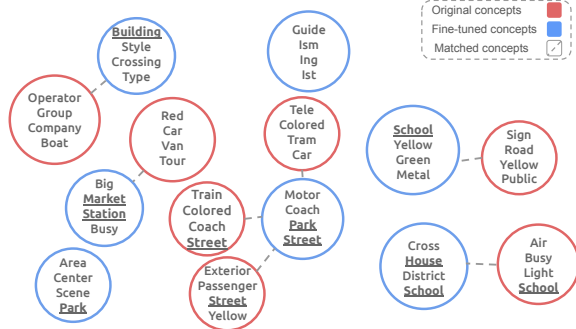


Figure 3. **Concepts text grounding change after fine-tuning.** Text grounding for concepts ($TOI = \text{bus}$) from f^a and their match from f^b , (fine-tuned to focus more on places). Emerging concepts may include grounding words not explicitly included in the fine-tuning vocabulary for place (e.g., "District", "Crossing"), while others evolve more smoothly (e.g., "Street").

match. Specifically, we define a matching function $m : i \rightarrow j^*$ which associates each concept vector \mathbf{u}_i^a in set U^a to its closest vector $\mathbf{u}_{j^*}^b$ in set U^b based on cosine similarity, i.e. $m(i) = \arg \max_{\mathbf{u}_j^b \in U^b} \cos(\mathbf{u}_i^a, \mathbf{u}_j^b)$.

Fig. 3 shows the text groundings for various concepts, displaying the words with a frequency lower than 5 across concepts (e.g., filtering out high-frequency terms like bus, vehicle, etc.). We observe the emergence of place-related terms across the identified elements, while the overall thematic structure remains consistent. Note that certain concepts may converge toward the same fine-tuned concept. We also analyze how the distances between matched concepts evolve during fine-tuning in App. A.3.

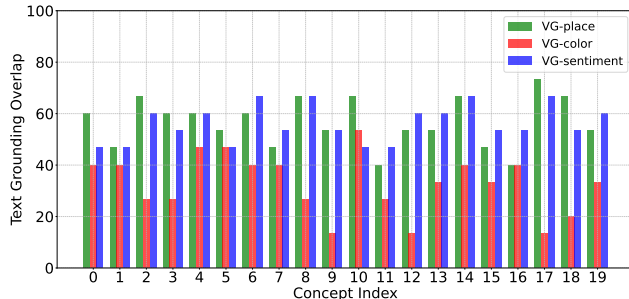


Figure 4. **Text grounding overlap (T-Overlap) between original and fine-tuned model concepts.** Different concepts change to different extents depending on the fine-tuning.

Concept evolution. To quantify how much a concept $\mathbf{u}_i^a \in U^a$ is changed after fine-tuning, we compute the overlap between its grounding words and those of its closest matching concept from the fine-tuned model U^b . Specifically, we compute $\text{T-Overlap}(\mathbf{u}_i^a, \mathbf{u}_{m(i)}^b)$ (Eq. (3)) for all the concepts $i \in \{1, \dots, K\}$, and visualize them for different fine-tunings in Fig. 4. We observe varying rates of change across different concepts and fine-tunings. This might be due to the difference in the fine-tuning dataset size, complexity,

or similarity to the original dataset. It also highlights 2 main behaviors, detailed as follows:

- *Concepts that are refined.* This group contains the concepts that slightly change to be more specialized towards the fine-tuning task (Fig. 5 top, middle rows). These concepts exhibit a relatively high ($\text{T-Overlap}(\mathbf{u}_i^a, \mathbf{u}_{m(i)}^b)$).
- *Concepts that change completely.* This group contains the concepts that emerge or, to a certain extent, disappear (Fig. 5 bottom row) in the fine-tuned model. New concepts emerge during fine-tuning likely due to the introduction of novel patterns or relationships. These concepts exhibit a relatively low ($\text{T-Overlap}(\mathbf{u}_i^a, \mathbf{u}_{m(i)}^b)$).

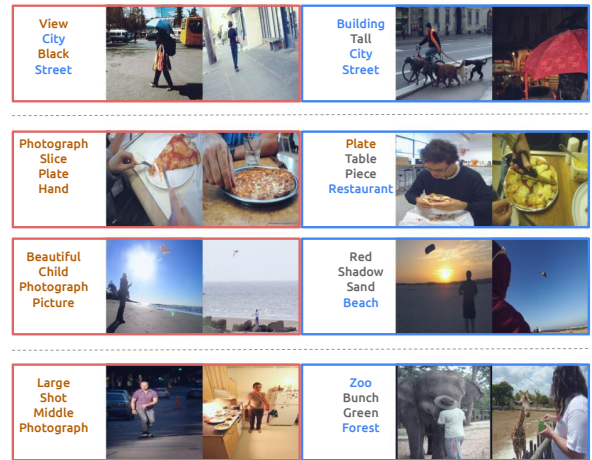


Figure 5. **Concepts evolve differently due to fine-tuning.** Left: concepts extracted from the original model f^a . Right: matched concept extracted from f^b , fine-tuned to focus on places. Each concept is grounded in image and text. We observe different levels of adaptation across concepts: some concepts specialize further by adding to place-related words (e.g., Top), some concepts introduce place-related words while staying aligned with the original concept in textual or visual groundings (e.g., Middle), while others undergo complete transformation, diverging significantly from their original meaning to fully embrace place-related elements (e.g., Bottom).

We also notice that T-Overlap decreases with the number of training iterations, indicating that fine-tuning leads to deviation from the original concepts (more details in App. A.3).

Evaluating fine-tuned concept recovery. To study if the fine-tuned concepts U^b can be recovered from the original ones U^a , we first establish a matching ($m : i \rightarrow j$) between the set of original $\{\mathbf{u}_i^a\}_{i=1}^K$ and fine-tuned concepts $\{\mathbf{u}_j^b\}_{j=1}^K$. For systematic evaluation of recovery of all fine-tuned concepts, we constrain m to be bijective using an optimal transport algorithm detailed in App. A.1. Finally, we evaluate how well a shifted concept \mathbf{u}_k^s (Equ. (4)) is similar to its match $\mathbf{u}_{m(k)}^b$ using the aforementioned T-Overlap metric. Fig. 6 shows the results of recovering the fine-tuned concepts for models fine-tuned on different subsets of the VG

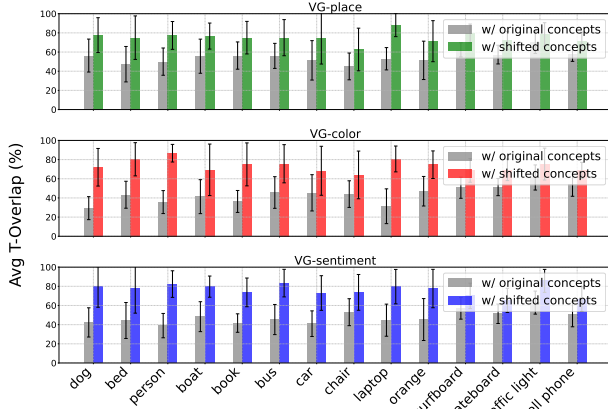


Figure 6. **Recovering fine-tuned concepts.** Across different finetunings (places (top), colors (middle) and sentiments (bottom)), we compute the average text grounding overlap of original and shifted concepts with the matched fine-tuned ones. Shifting the original concepts result in partially recovering the fine-tuned ones.

dataset (place, color, sentiment). We report the T-Overlap between the shifted u_k^s and fine-tuned concepts $u_{m(k)}^b$ for various different tokens of interest. For each target token and finetuning we extract $K = 20$ concepts and report the mean and standard deviation over them. We use the overlap between the original concepts u_k^a and the fine-tuned ones as a baseline. We observe that most shifted concepts show higher overlap than the original ones: this demonstrates that fine-tuned concepts can be efficiently recovered from the original model with concept shift vectors.

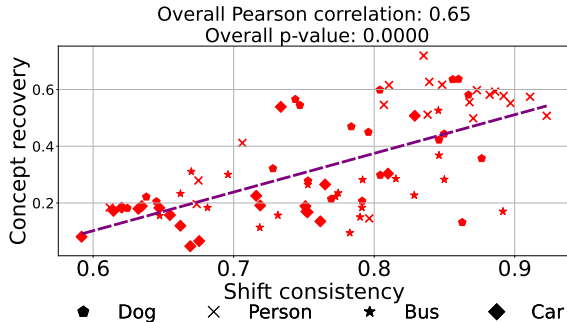


Figure 7. **Correlation between shift consistency and concept recovery (Color finetuning).** The more consistent and aligned the individual representation shifts associated with a concept, the better the recovery of the fine-tuned concept.

Which concepts are recovered better? We hypothesize that if the representation shift for individual samples associated with a concept u_k^a , $\{\delta_m^{a \rightarrow b} \mid m \in \mathcal{A}_k\}$, is *consistently aligned* with the concept shift vector, the resulting $\Delta_k^{a \rightarrow b}(u_k^a)$, should be more effective at recovering the fine-tuned concept. We quantify the consistency through mean cosine similarity of $\{\delta_m^{a \rightarrow b}\}_{m \in \mathcal{A}_k}$ with the concept shift vector $\Delta_k^{a \rightarrow b}(u_k^a)$. In other words, this quantifies the alignment

of individual shifts and their mean, $\Delta_k^{a \rightarrow b}(u_k^a)$:

$$\text{Consistency}(u_k^a) = \frac{1}{|\mathcal{A}_k|} \sum_{m \in \mathcal{A}_k} \cos(\delta_m, \Delta_k^{a \rightarrow b}(u_k^a)),$$

We measure the recovery of concept k , CR_k , as the improvement in similarity between the matched fine-tuned $u_{m(k)}^b$ and shifted concept u_k^s , relative to the original one u_k^a :

$$\text{CR}_k = \frac{\cos(u_{m(k)}^b, u_k^s) - \cos(u_{m(k)}^b, u_k^a)}{\cos(u_{m(k)}^b, u_k^a)} \quad (7)$$

We plot the consistency and recovery for concepts extracted across four tokens of interest for color finetuning in Fig. 7. Plots for other finetunings are in App. A.5. Crucially, we observe a positive and statistically significant correlation between the two quantities across all the finetuning tasks. This supports our hypothesis that a better concept shift recovery is related to consistency of individual shifts of original concept. More ablation studies about concept recovery are available in App. A.4, analyzing the influence of steering strength α , number of concepts K , and extraction layer l .

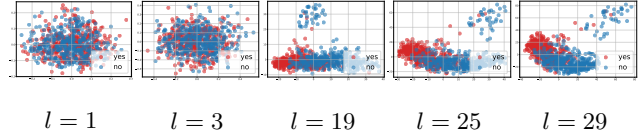


Figure 8. **Linear separability of concepts features in MLLMs.** We visualize the features related to the concepts "yes" and "no" after PCA projections across MLLM layers.

In summary, we demonstrated the feasibility of recovering target fine-tuned concepts by applying simple per-concept shift of the original model features. This supposes that the features related to different concepts are almost linearly separable, which empirically seems to hold at least for the last MLLM layers as pictured on Fig. 8, and as previously studied for LLMs in [42, 48]. This motivates the following investigations on using a similar methodology for simple, computationally inexpensive, yet efficient MLLM steering.

4.2. Multimodal model steering

We perform steering by applying a steering vector v to the residual stream features Z of the MLLM, without changing its parameters. We first evaluate the MLLM steering capabilities in a visual question-answering (VQA) setup. Then, we show the applicability of our MLLM steering to control captioning styles. Lastly, we present two steering applications: gender debiasing, aiming to mitigate biases in model outputs, and safety alignment, *i.e.* ensuring that the model refuses to generate harmful information. We discuss technical details for each of these applications in App. C and App. D.

Setup. For VQA tasks, each query consists of a question about an input image, and the model generates an answer.

To measure the effectiveness of our approach in directing the model towards specific answers or answer types, we report the number of generated answers that align with the target output or answer type. Additionally, we aim for targeted steering, ensuring that only specific answer types are influenced. For example, when altering answers from “yes” to “no” within the “yes/no” category, responses to other question types should remain unaffected. This specificity is assessed by tracking accuracy across answer types and number of answers from each type.

Implementation details. Experiments in the main paper are primarily conducted on LLaVA [39] for conciseness. However, we show in App. B.2 that our method is generally applicable to other popular MLLMs. We experiment on VQAv2 [24], a visual question-answering corpus with image-question-answer triplets and annotated answer types (“yes/no”, “number”, and “other”). Steering vectors are derived from a subset of the train set, with model performance evaluated on the val set. As steering becomes more effective in deeper layers (see Fig. 8 and App. B), we apply it on the last layer. Additional experiments can be found in App. B.

Target Type	Answers Type		
	yes/no	number	other
N/A	366	122	494
yes/no	557	96	288
number	327	201	390
other	364	115	501

Table 1. **Steering MLLMs answers type.** Number of target answers type increases significantly after model steering.

Coarse and Fine-Grained Model Steering for VQA. We explore both coarse and fine-grained steering in VQA tasks. Specifically, coarse steering aims to alter the distribution of answers, while fine-grained steering targets specific responses. For coarse-grained steering, we direct the model’s answers toward a particular category: yes/no, numbers, or other (e.g., colors, objects). We compute a steering vector for each target answer type. As shown in Table 1, applying steering significantly increases the proportion of the targeted answer type. For fine-grained steering, we first assess the feasibility of identifying such steering vectors. Fig. 9 illustrates examples of these vectors. We derive them from three sets of sample answers corresponding to “yes/no,” “number,” and “other” categories. Interestingly, some vectors distinctly align with specific answers such as “No,” “4,” and “Red.” This confirms the potential to identify fine-grained steering vectors capable of guiding the model toward a precise response. Building on these insights, we seek to steer the model toward a user-specified answer. For each original/target answer pair (e.g., yes/no), we collect some samples and compute the corresponding (coarse) steering vector. We then apply these vectors to all validation set samples. In

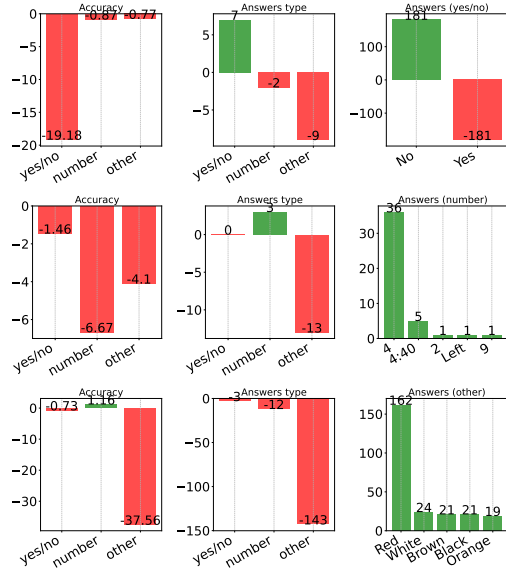


Figure 9. **Discovering meaningful steering directions.** Each line corresponds to a fine-grained steering direction to steer the model answer to: “No” (yes/no), “4” (number) and “Red” (other). Some steering directions are targeted (e.g., “No”) as there is slight change in both the accuracy and number of answers types on other types (e.g., number, other). We show the relative scores compared to a baseline with no steering.

Table 2, we report evaluation metrics when steering at the last layer. The results show that steering effectively increases the occurrence of target answers, while accuracy on other answer types remains largely stable.

Steering	Accuracy (%)			Answer Types			Answers	
	Yes/No	Number	Other	Yes/No	Number	Other	Original	Target
N/A	90.82	58.47	71.10	1861	687	2349	0	0
Yes → No	69.03	56.82	68.99	1884	695	2294	-828	+828
1 → 3	90.71	54.52	71.12	1861	670	2350	-215	+144
White → Black	90.40	58.42	58.36	1861	671	2312	-98	+441

Table 2. **Steering MLLMs answers.** Steering answers from “Yes” (Yes/No), “1” (Number), “White” (Other) to “No”, “3”, “Black” respectively. The number of original/target answer counts decreases/increases significantly, while the accuracy on other answer types changes only slightly, and the number of answer type counts remains almost constant.

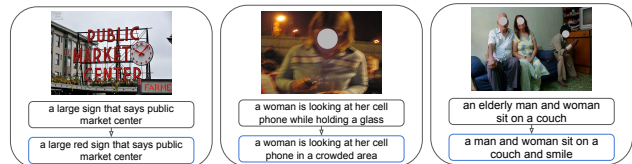


Figure 10. **Steering MLLMs captions style.** Captions steered to focus more on colors (left), places (middle) and sentiments (right).

Steering image caption styles. We previously applied steering on relatively brief answers from the VQAv2 dataset. Here, we extend this approach to longer, descriptive outputs

using the COCO captioning dataset [37]. Given that multiple captions can effectively describe an image by emphasizing various aspects such as the main object, surroundings, actions, or events, we aim at modifying captions to align with a specific target style. Here, we learn a coarse steering vector between samples with predicted captions in the target style and random samples. Qualitative examples of this steering for LLaVA are in Fig. 10. Results in Table 3 demonstrate that captions can be effectively steered towards a target style even when considering tasks with longer responses. We provide more captioning experiments in App. B.4.

Target Style	Captions Style		
	places	colors	sentiments
N/A	430	1309	2
places	796	1077	1
colors	488	2561	1
sentiments	393	1040	48

Table 3. **Steering MLLMs captions style.** Each line corresponds to a different steering vector. Steering towards a target style increases the number of captions with that style.

Gender debiasing captions. We perform gender debiasing on COCO test set, aiming at mitigating biases for any gendered nouns when captioning. We experiment with both coarse and fine-grained steering. The coarse steering vector is computed between sets of samples with a gendered/neutral noun in the caption. For fine-grained steering, we steer each concept to its closest (as defined by its cosine similarity) counterpart among the neutral concepts, as explained in Section 3.3. The results are reported in Table 4. Interestingly, both strategies are capable of converting many gendered captions to neutral ones, with fine-grained steering being significantly more effective than coarse steering.

Safety alignment. Pure text LLMs often exhibit stronger safety alignment compared to MLLMs [11]. Empirical evidence for this can be found in App. D. Using this insight, we construct two sets of samples, categorized as safe and unsafe, by evaluating the model’s response to identical malicious content presented in different modalities: one conveyed through text and the other through text + image. We use these to compute a safety guard steering vector. This steering vector gives the model a higher level of safety, without affecting its usefulness for safe tasks (Table 5). We evaluate the safety of the model by ASR (attack success rate) metric, described in detail in App. D.

5. Discussion

Limitations. The effectiveness of our method relies on the fact that the concepts are represented by linear directions in latent space. However, recent work [14] has found that not all features are captured as such: thus, when analyzing recovery

Model	Total	Method	Gendered → Neutral
LLaVA-1.5	794	coarse	232
		fine-grained	632
Idefics2	815	coarse	237
		fine-grained	315
Qwen2-VL-Instruct	926	coarse	134
		fine-grained	300

Before Steering	After Steering
<i>A young boy with curly hair is playing a video game.</i>	<i>A child with curly hair is playing a video game.</i>
<i>A man riding a dirt bike on a beach.</i>	<i>A person riding a dirt bike on a beach.</i>

Table 4. **Gender debiasing results:** number of occurrences of gendered terms converted to neutral terms across different models and methods, after steering with $\alpha = 1$. Below, qualitative samples illustrate changes in descriptions before and after applying steering.

Model	Before steering	After steering
Qwen2-VL-Instruct	45/100	5/100

Table 5. **Enhancing MLLM Safety Through Steering.** We evaluate safety using the ASR metric, quantifying the proportion of responses that do not refuse to provide harmful instructions. Our assessment is conducted on a portion of MM-SafetyBench [40] dataset. A lower ASR is desired when the prompt requires harmful instructions.

of fine-tuned concepts, it can be interesting to explore more sophisticated similarity measures and matching algorithms.

Conclusion. In this work, we introduced a novel concept-based analysis framework for monitoring and controlling MLLM behavior, offering new insights into how latent representations evolve during fine-tuning and across datasets. To address the former, we proposed *concept shift vectors*, an efficient method for recovering and interpreting concepts in fine-tuned models relative to their original counterparts. This approach naturally led us to explore the latter case, where we demonstrated the ability to *steer model behavior* by modifying features – without requiring additional training. Our results show that this technique effectively modifies MLLM answers, enabling applications such as gender debiasing, safety control, and enhanced caption generation that highlight different aspects of an image. By releasing our code, we hope our framework will benefit the community and encourage research towards better understanding of MLLMs, as well as their broader applications in domains such as physical and digital agents [50, 60].

Acknowledgements

This work has been partially supported by ANR grant VISA DEEP (ANR-20-CHIA-0022), HPC resources of IDRIS under the file A0160614966 allocated by GENCI, and Cluster PostGenAI@Paris (ANR-23-IAEL-0007, FRANCE 2030).

References

- [1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022. 1
- [2] Andy Arditi, Oscar Obeso, Aaqib Syed, Daniel Paleka, Nina Rimsky, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *arXiv preprint arXiv:2406.11717*, 2024. 2
- [3] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023. 1, 2
- [4] Folco Bertini Baldassini, Mustafa Shukor, Matthieu Cord, Laure Soulier, and Benjamin Piwowarski. What makes multimodal in-context learning work? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1539–1550, 2024. 1, 2
- [5] Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor V. Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *ArXiv*, abs/2303.08112, 2023. 3
- [6] Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, and Stella Biderman. Leace: Perfect linear concept erasure in closed form. *Advances in Neural Information Processing Systems*, 36, 2024. 2
- [7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:1877–1901, 2020. 1
- [8] Shi Chen and Qi Zhao. Rex: Reasoning-aware and grounded explanation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15586–15595, 2022. 2
- [9] Shuo Chen, Zhen Han, Bailan He, Mark Buckley, Philip Torr, Volker Tresp, and Jindong Gu. Understanding and improving in-context learning on vision-language models. In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2024. 2
- [10] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022. 1
- [11] Yi Ding, Bolian Li, and Ruqi Zhang. Eta: Evaluating then aligning safety of vision language models at inference time. *ArXiv*, abs/2410.06625, 2024. 8, 10, 11
- [12] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023. 1
- [13] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1, 2021. 1
- [14] Joshua Engels, Eric J Michaud, Isaac Liao, Wes Gurnee, and Max Tegmark. Not all language model features are linear. *arXiv preprint arXiv:2405.14860*, 2024. 8
- [15] Brandon Huang et al. Multimodal task vectors enable many-shot multimodal in-context learning. In *NeurIPS*, 2024. 2
- [16] Yingzhe et al. LIVE: Learnable in-context vector for visual question answering. In *NeurIPS*, 2024. 2
- [17] Thomas Fel, Victor Boutin, Louis Béthune, Rémi Cadène, Mazda Moayeri, Léo Andéol, Mathieu Chalvidal, and Thomas Serre. A holistic approach to unifying automatic concept extraction and concept importance estimation. *Advances in Neural Information Processing Systems*, 36, 2023. 2
- [18] Thomas Fel, Agustin Picard, Louis Bethune, Thibaut Boissin, David Vigouroux, Julien Colin, Rémi Cadène, and Thomas Serre. Craft: Concept recursive activation factorization for explainability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2711–2721, 2023. 2
- [19] Enrico Fini, Mustafa Shukor, Xiujun Li, Philipp Dufter, Michal Klein, David Haldimann, Sai Aitharaju, Victor Guilherme Turrisi da Costa, Louis Béthune, Zhe Gan, et al. Multimodal autoregressive pre-training of large vision encoders. *arXiv preprint arXiv:2411.14402*, 2024. 2
- [20] Jiaxin Ge, Sanjay Subramanian, Trevor Darrell, and Boyi Li. From wrong to right: A recursive approach towards vision-language explanation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1185, 2023. 2
- [21] Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 9277–9286, 2019. 2
- [22] Yichen Gong, Delong Ran, Jinyuan Liu, Conglei Wang, Tianshuo Cong, Anyu Wang, Sisi Duan, and Xiaoyun Wang. Fig-step: Jailbreaking large vision-language models via typographic visual prompts. *ArXiv*, abs/2311.05608, 2023. 11
- [23] Yunhao Gou, Kai Chen, Zhili Liu, Lanqing Hong, Hang Xu, Zhenguo Li, Dit-Yan Yeung, James T. Kwok, and Yu Zhang. Eyes closed, safety on: Protecting multimodal llms via image-to-text transformation. In *European Conference on Computer Vision*, 2024. 11
- [24] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 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, 2017. 7, 3
- [25] J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685, 2021. 3, 1

- [26] Kaichen Huang, Jiahao Huo, Yibo Yan, Kun Wang, Yutao Yue, and Xuming Hu. Miner: Mining the underlying pattern of modality-specific neurons in multimodal large language models. *arXiv preprint arXiv:2410.04819*, 2024. **2**
- [27] Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*, 2024. **2**
- [28] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023. **1, 2**
- [29] Yuchu Jiang, Jiale Fu, Chenduo Hao, Xinting Hu, Yingzhe Peng, Xin Geng, and Xu Yang. Mimic in-context learning for multimodal tasks. *arXiv preprint arXiv:2504.08851*, 2025. **2**
- [30] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018. **2**
- [31] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *Int. J. Comput. Vis.*, 123(1):32–73, 2017. **4, 1**
- [32] Anna Langedijk, Hosein Mohebbi, Gabriele Sarti, Willem Zuidema, and Jaap Jumelet. Decoderlens: Layerwise interpretation of encoder-decoder transformers. *ArXiv*, abs/2310.03686, 2023. **3**
- [33] Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander Rush, Douwe Kiela, et al. Obelics: An open web-scale filtered dataset of interleaved image-text documents. *Advances in Neural Information Processing Systems*, 36, 2024. **1**
- [34] Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. What matters when building vision-language models? *arXiv preprint arXiv:2405.02246*, 2024. **1, 2, 3**
- [35] Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36, 2024. **2**
- [36] Lijun Li, Bowen Dong, Ruohui Wang, Xuhao Hu, Wangmeng Zuo, Dahua Lin, Yu Qiao, and Jing Shao. Salad-bench: A hierarchical and comprehensive safety benchmark for large language models. In *Annual Meeting of the Association for Computational Linguistics*, 2024. **10**
- [37] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. **8, 1, 3, 9**
- [38] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2023. **4**
- [39] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024. **1, 2, 3, 7**
- [40] Xin Liu, Yichen Zhu, Jindong Gu, Yunshi Lan, Chao Yang, and Yu Qiao. Mm-safetybench: A benchmark for safety evaluation of multimodal large language models. In *European Conference on Computer Vision*, 2023. **8, 10**
- [41] Oscar Mañas, Pau Rodriguez Lopez, Saba Ahmadi, Aida Nematzadeh, Yash Goyal, and Aishwarya Agrawal. Mapl: Parameter-efficient adaptation of unimodal pre-trained models for vision-language few-shot prompting. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2523–2548, 2023. **1**
- [42] Neel Nanda. Actually, othello-gpt has a linear emergent world model, 2023. **6, 8**
- [43] Nostalgebraist. Interpreting gpt: The logit lens. <https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens>, 2020. Accessed: [date of access]. **3**
- [44] OpenAI. Gpt-4 technical report. *arXiv*, 2023. **1**
- [45] Haowen Pan, Yixin Cao, Xiaozhi Wang, and Xun Yang. Finding and editing multi-modal neurons in pre-trained transformer. *arXiv preprint arXiv:2311.07470*, 2023. **2**
- [46] Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering llama 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06681*, 2023. **2**
- [47] Jayneel Parekh, Pegah Khayatan, Mustafa Shukor, Alasdair Newson, and Matthieu Cord. A concept-based explainability framework for large multimodal models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. **1, 2**
- [48] Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Forty-first International Conference on Machine Learning*, 2024. **6, 8**
- [49] Libo Qin, Qiguang Chen, Hao Fei, Zhi Chen, Min Li, and Wanxiang Che. What factors affect multi-modal in-context learning? an in-depth exploration. *arXiv preprint arXiv:2410.20482*, 2024. **2**
- [50] Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, Shihao Liang, Shizuo Tian, Junda Zhang, Jiahao Li, Yunxin Li, Shijue Huang, et al. Ui-tars: Pioneering automated gui interaction with native agents. *arXiv preprint arXiv:2501.12326*, 2025. **8**
- [51] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. **2**
- [52] Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, János Kramár, Rohin Shah,

- and Neel Nanda. Improving dictionary learning with gated sparse autoencoders. In *Advances in Neural Information Processing Systems*, 2024. 2
- [53] Shauli Ravfogel, Michael Twiton, Yoav Goldberg, and Ryan D Cotterell. Linear adversarial concept erasure. In *International Conference on Machine Learning*, pages 18400–18421. PMLR, 2022. 2
- [54] Mansi Sakarvadia, Arham Khan, Aswathy Ajith, Daniel Grzenda, Nathaniel Hudson, André Bauer, Kyle Chard, and Ian T. Foster. Attention lens: A tool for mechanistically interpreting the attention head information retrieval mechanism. *CoRR*, abs/2310.16270, 2023. 3
- [55] Sarah Schwettmann, Neil Chowdhury, Samuel Klein, David Bau, and Antonio Torralba. Multimodal neurons in pretrained text-only transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2862–2867, 2023. 1, 2
- [56] Mustafa Shukor and Matthieu Cord. Implicit multimodal alignment: On the generalization of frozen llms to multimodal inputs. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 1, 2
- [57] Mustafa Shukor and Matthieu Cord. Skipping computations in multimodal llms. *arXiv preprint arXiv:2410.09454*, 2024. 1
- [58] Mustafa Shukor, Corentin Dancette, and Matthieu Cord. epalm: Efficient perceptual augmentation of language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22056–22069, 2023. 1
- [59] Mustafa Shukor, Alexandre Rame, Corentin Dancette, and Matthieu Cord. Beyond task performance: evaluating and reducing the flaws of large multimodal models with in-context-learning. In *The Twelfth International Conference on Learning Representations*, 2024. 1, 2
- [60] Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, et al. Smolvla: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025. 8
- [61] Mustafa Shukor, Enrico Fini, Victor Guilherme Turrise da Costa, Matthieu Cord, Joshua Susskind, and Alaaeldin El-Nouby. Scaling laws for native multimodal models. *arXiv preprint arXiv:2504.07951*, 2025. 1
- [62] Nishant Subramani, Nivedita Suresh, and Matthew E Peters. Extracting latent steering vectors from pretrained language models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 566–581, 2022. 2
- [63] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024. 2
- [64] Curt Tigges, Oskar John Hollinsworth, Atticus Geiger, and Neel Nanda. Linear representations of sentiment in large language models. *arXiv preprint arXiv:2310.15154*, 2023. 2
- [65] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 1, 2
- [66] Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Activation addition: Steering language models without optimization. *arXiv preprint arXiv:2308.10248*, 2023. 2
- [67] Théophane Vallaey, Mustafa Shukor, Matthieu Cord, and Jakob Verbeek. Improved baselines for data-efficient perceptual augmentation of llms. *arXiv preprint arXiv:2403.13499*, 2024. 1, 2
- [68] Yu Wang, Xiaogeng Liu, Yu Li, Muhao Chen, and Chaowei Xiao. Adashield: Safeguarding multimodal large language models from structure-based attack via adaptive shield prompting. In *European Conference on Computer Vision*, 2024. 10
- [69] Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D Manning, and Christopher Potts. Ref: Representation finetuning for language models. *arXiv preprint arXiv:2404.03592*, 2024. 2
- [70] Dizhan Xue, Shengsheng Qian, and Changsheng Xu. Few-shot multimodal explanation for visual question answering. In *ACM Multimedia 2024*, 2024. 2
- [71] Chih-Kuan Yeh, Been Kim, Sercan O Arik, Chun-Liang Li, Pradeep Ravikumar, and Tomas Pfister. On concept-based explanations in deep neural networks. *arXiv preprint arXiv:1910.07969*, 2019. 2
- [72] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023. 2
- [73] Xiaofeng Zhang, Chen Shen, Xiaosong Yuan, Shaotian Yan, Liang Xie, Wenxiao Wang, Chaochen Gu, Hao Tang, and Jieping Ye. From redundancy to relevance: Enhancing explainability in multimodal large language models. *arXiv preprint arXiv:2406.06579*, 2024. 1