Interpretability Transfer from Language to Vision via Sparse Autoencoders

Alexey Kravets [1] **Da Li** [2] **Chuan Li** [3] **Da Chen** [1] ✉ **Vinay P. Namboodiri** [1]

Abstract

Recent advances in language model interpretability using sparse autoencoders (SAEs) have yet to effectively translate to the visual domain, mainly due to the difficulty and ambiguity of labeling visual concepts. In this paper, we introduce Visual Interpretability via SAE Transfer Alignment (VISTA), a framework that transfers interpretability from language to vision in a LLaVA-style vision-language model by constraining a visual projector to map visual tokens into an LLM's pre-existing, labeled textual SAE space. This approach enables visual interpretability without training dedicated vision SAEs. By regularizing the projector using the LLM's SAE reconstruction loss, VISTA achieves a threefold increase in the matching rate, which measures how accurately the most activating textual concepts in the SAE space correspond to semantic elements in the image. Using this framework, we further analyze spatial localization properties of different vision encoders and show that DINOv2 features have stronger localization abilities than other encoders. Leveraging this precision, we validate VISTA's cross-modal alignment through fine-grained, localized concept interventions, where specific objects are removed or replaced in the model's perception while preserving the surrounding scene. This results in improvements of 35% in object removal and 47% in object replacement tasks over vision-only baselines, providing causal evidence that visual tokens inhabit the text SAE manifold. These contributions are validated across multiple LLM architectures. Code is available on GitHub.

[1]University of Bath, UK [2]Samsung AI Centre Cambridge [3]Lambda, Inc.. Correspondence to: Da Chen <da.chen@bath.edu>.

Mechanistic Interpretability Workshop at the 43rd International Conference on Machine Learning, Seoul, South Korea, 2026.
Copyright 2026 by the author(s).

1. Introduction

Interpreting internal representations of Large Multimodal Models (LMMs) remains a challenge despite their strong visual reasoning capabilities. While sparse autoencoders (SAEs) have successfully decomposed polysemantic language features into interpretable concepts (Bricken et al., 2023), applying them to the visual modality is obstructed by limitations in the two primary solutions. The first solution involves training domain-specific SAEs on visual encoders (Lim et al., 2025; Pach et al., 2025; Joseph et al., 2025) which yields monosemantic *visual features* or *visual concepts*. However, labeling these features is difficult and ambiguous. For instance, distinguishing whether a "dog" feature represents the dog concept, a more specific breed, or even an action may be hard. Furthermore, vision-only SAEs do not reveal how visual features are interpreted once they enter the LLM's residual stream. The second potential solution is to fine-tune existing LLM-based SAEs to incorporate visual information, but this approach incurs prohibitive computational costs. For example, Gemma Scope (Lieberum et al., 2024) required ∼20% of GPT-3's total compute and would necessitate an extensive re-labeling effort (circa half a million LLM requests for Gemma-2-2B alone) to account for the shifted latent space.

To solve this, we ask the following question:

> **Q:** *Can language sparse autoencoders be reused to interpret visual concepts?*

In this paper, we answer this affirmatively through our solution **VISTA** (Visual Interpretability via SAE Transfer Alignment), that bypasses the need for new SAEs by leveraging existing, labeled feature space of text SAEs. Instead of training or fine-tuning SAEs, we introduce a training objective that forces the visual projector to map visual tokens directly into the LLM's residual stream, constrained such that they align with the semantic structure defined by the text SAEs. In this way, we achieve **interpretability transfer**, allowing visual inputs to be interpreted via pre-labeled linguistic SAE concepts with negligible additional computational cost.

VISTA is applied within the LLaVA paradigm (Liu et al., 2023). A frozen vision encoder produces visual tokens, a trainable linear projector maps them into a frozen LLM's

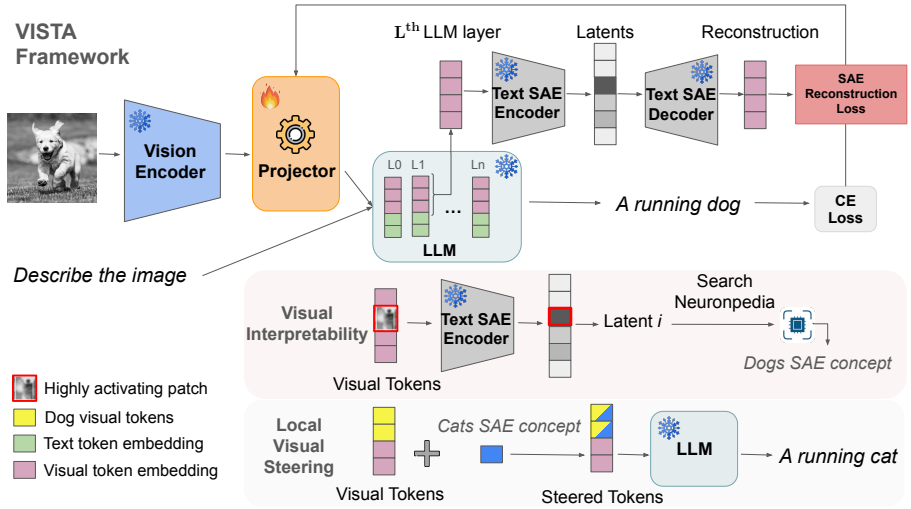


Figure 1. **VISTA Framework. Training:** A trainable projector maps tokens from a frozen vision encoder to a frozen LLM, optimized via Cross-Entropy and an auxiliary SAE Reconstruction Loss to force alignment of the visual embeddings with the manifold defined by pre-trained Text SAEs. **Visual Interpretability:** Once aligned, highly activating visual tokens can be interpreted via the Text SAE. Activating latents are searched in Neuronpedia for their linguistic labels that correspond to the visual concept in the patch, such as "Dog" concept in the patch showing the dog. **Steering:** By manipulating specific SAE concepts in localized visual tokens, model's output can be precisely steered (changing "A running dog" to "A running cat") preserving global context.

residual stream, and the LLM generates text conditioned on the resulting multimodal sequence. This design is grounded in the finding by Merullo et al. (2023) that linear mappings suffice for cross-modal alignment. VISTA augments this standard setup with a single addition of an SAE reconstruction loss constraining visual tokens to lie on the LLM's pre-trained text SAE manifold. All interpretations and interventions in this work operate on visual token representations within the LLM's residual stream.

Three metrics are applied to evaluate how well visual features align with the textual SAE space: reconstruction error, feature sparsity, and an LLM-verified matching rate (Venhoff et al., 2025). Reconstruction and sparsity dynamics (Section 4.3) determine if the projected visual tokens match the sparse distribution and reconstructibility of the LLM's text features while matching rate measures how accurately the most activating text SAE concepts activating on the visual tokens correspond to the semantic elements visible in the image (Section 4.4). VISTA constraints lead the matching rate to increase more than a **threefold** on average.

Unlike previous beliefs (Venhoff et al., 2025), we find that "delayed alignment", a phenomenon where visual tokens initially diverge from the text SAE sparse linguistic manifold, showing high reconstruction error and low feature sparsity, and only naturally converge toward the model's text-based feature space in deeper layers is not a universal characteristic of all LLM architectures. While visual tokens in Gemma SAEs eventually align with the textual patterns in later layers, this natural alignment does not occur for the LLaMA-3.1-8B-Instruct SAEs. By applying our SAE constraints during training, alignment is achieved immediately across all architectures (Section 4.3).

Furthermore, we show that precise local interpretability and steering depend heavily on the choice of the vision encoder (Section 4.2). Local interpretability requires that activated SAE concepts are spatially aligned with specific image patches. For example, a patch covering the background might incorrectly activate a concept like "cat" (Figure 2 bottom-left) that actually belongs to an object elsewhere in the scene. Our analysis reveals that while contrastive models like CLIP (Radford et al., 2021) prioritize global summaries and lead to spatial confusion, self-supervised models like DINOv2 (Oquab et al., 2024) preserve the fine-grained local information required for patch-level accuracy. Specifically, DINOv2 maintains robust spatial localization accuracy across different LLM architectures, consistently outperforming CLIP, whose localization quality varies depending on the LLM backbone but remains substantially lower. This spatial fidelity is a key requirement for fine-grained interpretability, as it allows concept activations to be reliably localized to the correct image regions. The same property enables the localized causal interventions demonstrated in Section 5. Our **contributions** are:

- We introduce a training objective that aligns visual tokens with LLM SAE features, facilitating cross-modal interpretability transfer without the computational burden of training or labeling vision-specific SAEs.
- We provide a rigorous comparison of different visual encoders enabled by VISTA's interpretability

transfer, revealing that DINOv2 achieves superior and architecture-stable spatial localization, maintaining high accuracy across all tested LLM backbones, whereas CLIP's localization accuracy is lower and varies substantially with the LLM. By tracing exactly which patch triggers which semantic SAE concept, VISTA exposes encoder-level spatial fidelity that prior interpretability methods could not measure.

- As a causal validation of our cross-modal alignment, we demonstrate that arithmetic interventions on visual tokens using text-SAE directions produce coherent, localized changes in model output, achieving **35%** improvement in object removal and **47%** in replacement tasks over best vision-only baselines.

2. Related Work

Sparse autoencoders address polysemanticity (Elhage et al., 2022) by decomposing activations into monosemantic features (Bricken et al., 2023; Huben et al., 2024). Many works in the field of mechanistic interpretability use this decomposition as a basis for further investigation, such as the causal role these features play in downstream tasks (Marks et al., 2025), the universality of features across different model scales (Wang et al., 2025), and the potential for feature steering to mitigate hallucinations (Abdaljalil et al., 2025). While foundational for LLMs, visual applications are emerging. Recent work has applied SAEs to CLIP encoders to identify interpretable features (Pach et al., 2025; Joseph et al., 2025; Daujotas, 2024), but these primarily utilize global [CLS] tokens, limiting their ability to explain or steer localized objects. While Lim et al. (2025) trained SAEs on patches for attribution, their method is primarily designed for the explanation of classification decisions rather than for the steering or control of model behavior.

The difficulty in labeling visual features suggests that a more effective path to interpretability lies in cross-modal alignment with the linguistic domain. While traditional alignment relies on contrastive losses, recent work shows that linear mappings often suffice. Merullo et al. (2023) showed that a linear projection can map frozen vision encoders to frozen LLMs. Venhoff et al. (2025) found that visual features naturally align with language features in deeper LLM layers. We extend this by enforcing alignment via SAE reconstruction. By constraining the projector to output vectors reconstructible by text SAEs, we achieve "interpretability transfer" leveraging the LLM's semantic structure to interpret visual tokens without training separate vision-based models.

This alignment enables more effective activation steering and control. Building on representation engineering, where model behavior is modulated by adding steering vectors to internal activations (Rimsky et al., 2024; Turner et al., 2023), it has been shown (Pach et al., 2025) that SAE-based directions to steer VLMs outperform traditional Difference-in-Means (Arditi et al., 2024) steering in preserving scene integrity. We demonstrate that by enforcing SAE constraints on the visual projector in VLMs, we can perform high-fidelity steering directly within the LLM's residual stream outperforming vision encoder based VLMs steering.

3. Method

In this paper, we introduce VISTA (Visual Interpretability via SAE Transfer Alignment), a framework that maps visual tokens into the pre-existing, labeled SAE feature space of a frozen LLM. By adding an SAE reconstruction loss to the training objective, we force visual features to lie in the text SAE manifold. This enables the analysis and steering of visual concepts using textual descriptors, bypassing the need for vision-specific SAE training or the fine-tuning of existing SAEs. This section details the VISTA training objective, the localized steering mechanism enabled by this alignment, and the metrics used to evaluate the precision of this transfer.

3.1. VISTA

To disentangle polysemantic LLM representations (Elhage et al., 2022), SAEs use an encoder-decoder architecture to map activations x into a high-dimensional, sparse latent space $E(x) \in \mathbb{R}^{d_{SAE}}$ (Bricken et al., 2023), where d_{SAE} is typically 10 to 100 times the size of the original input. The model is trained to generate a reconstruction $\hat{x}$ by linearly combining a small subset of features in $E(x)$. By minimizing the error $\|x - \hat{x}\|_2^2$ alongside an L_1 sparsity penalty, the SAE is forced to find a near-perfect reconstruction using only a few human-interpretable "concepts". See Section A for the full mathematical derivation.

As depicted in Figure 1, our VLM architecture follows the LLaVA paradigm (Liu et al., 2023) augmented by the VISTA alignment constraint. Let V be a frozen vision encoder (e.g., DINOv2) and L be a frozen language model (e.g., Gemma). We introduce a learnable linear projector P_θ. Given an image I, the vision encoder produces visual tokens $v = V(I) \in \mathbb{R}^{N \times D_v}$ where N is the number of visual tokens and D_v is their embedding dimension. P_θ maps these to the LLM's embedding space:

$$e_v = P_\theta(v) \in \mathbb{R}^{N \times D_{llm}} \tag{1}$$

These visual embeddings are concatenated with text embeddings e_t corresponding to the user input (e.g., a question or caption prompt) and passed to the LLM.

Traditionally, the projector P_θ is trained solely by minimizing the cross-entropy loss ($\mathcal{L}_{CE}$) on the target text se-

quences, such as answers or image descriptions (Venhoff et al., 2025). We propose augmenting this with a constraint derived from a pre-trained LLM sparse autoencoder. Let SAE be an SAE trained on the residual stream (or activations) of the language model L using only text data. The SAE consists of an encoder and a decoder, such that $\hat{x} \approx SAE(x)$. We hypothesize that if visual tokens e_v are to be interpretable to the LLM, they should be representable by the sparse features learned from text. To ensure visual tokens are interpretable within this linguistic framework, we define the SAE Reconstruction Loss ($\mathcal{L}_{SAE}$) as the Mean Squared Error between the projected visual tokens and their reconstruction by the text SAE:

$$\mathcal{L}_{SAE} = \frac{1}{|M|} \sum_{l \in M} ||SAE_l(e_v^l) - e_v^l||_2^2 \tag{2}$$

where e_v^l denote the hidden state of the visual tokens at layer l of the frozen LLM and M is the set of layers at which the SAE reconstruction loss is applied to the visual tokens. The total training objective becomes:

$$\mathcal{L}_{total} = \mathcal{L}_{CE} + \mathcal{L}_{SAE} \tag{3}$$

Minimizing this joint loss forces the projector to satisfy both the functional requirements of the multimodal task and visual token reconstruction via textual SAEs. We train the visual projector in two stages keeping the vision encoder and LLM frozen, adopting the training setup from Venhoff et al. (2025). Pre-training uses 595k image-text pairs from Conceptual Captions (Sharma et al., 2018), and the fine-tuning stage uses 665k examples from the LLaVA-1.5 pipeline (Liu et al., 2023). Additional details are in Section D.

3.2. Fine-grained Visual Steering Mechanism

The VISTA alignment forces visual tokens to inhabit the same latent space as text. This transforms the task of visual manipulation into a direct arithmetic operation within the linguistic manifold. As the training objective ensures visual tokens are represented as sparse combinations of pre-existing textual SAE features, we can modify the model's perception of specific token positions v_{source} by adding or subtracting the decoder directions (concepts) of the text SAE. This allows us to apply removal and replacement operations to specific visual tokens.

Concept Removal and Replacement For source patches v_{source}, removal subtracts a concept direction v_{remove} to suppress it, while replacement adds a target direction v_{add} to overwrite the local semantics:

$$v_{\text{steered}} = v_{\text{source}} - \beta \cdot v_{\text{remove}}, \qquad v_{\text{steered}} = v_{\text{source}} + \alpha \cdot v_{\text{add}}. \tag{4}$$

The non-negative scalars α and β control intensity, and restricting updates to selected token positions yields localized steering that leaves the rest of the scene intact.

3.3. Spatial Localization and Evaluation Framework

We evaluate SAE-constrained alignment through multiple metrics. We first conduct a spatial localization accuracy analysis to verify if visual tokens activating specific semantic latents correspond to the object's physical location in the image. We then analyze reconstruction and sparsity to determine if projected visual tokens match the sparse distribution and reconstructibility of the LLM's text features. Finally, we utilize an LLM-verified matching rate, employing an oracle LLM to confirm that activated concepts on the visual features correspond to semantic concepts present in the scene. Formal definitions for sparsity, reconstructibility and matching rate are provided in Section 4.

Spatial Localization Accuracy To evaluate the precision of the aligned visual tokens, we define *Spatial Localization Accuracy (SLA)*. This metric verifies whether the visual token that maximally activates a specific SAE latent is spatially located at the corresponding object in the image. The evaluation proceeds in two stages: we first confirm that an activated SAE latent corresponds to a recognizable object in the image, then verify that the visual token activating it covers that object. We use a subset of images sampled from COCO dataset (Lin et al., 2015). For each image, we identify the three latents with the highest activations and record their visual tokens for every layer. These latent descriptions are retrieved from Neuronpedia[1]. We employ an oracle LLM to retain only latents whose descriptions correspond to localizable objects in the image, filtering out both irrelevant concepts and abstract or broad descriptions (e.g., "building structure and layout") that do not match specific COCO classes. Using COCO ground-truth bounding boxes, we then verify whether the visual token associated with each matched concept falls within the corresponding object's bounding box.

4. Findings & Results

4.1. How do VISTA Constraints Affect Performance?

We evaluate the impact of SAE constraints on model performance across standard benchmarks, such as MME (Fu et al., 2025), GQA (Hudson & Manning, 2019), POPE (Li et al., 2023), and LLaVA-Bench (Liu et al., 2023). In smaller models like Gemma-2-2B-it, we generally observe a slight performance regression, though DINOv2 maintains or slightly improves performance on specific benchmarks like LLaVA-Bench and POPE. On the other hand, as model scale increases, the impact of these interpretability constraints becomes mixed. For the LLaMA-3.1-8B-Instruct architecture, the SAE constraint acts as a beneficial regularizer, leading to improved performance on the majority of benchmarks for every visual encoder. Conversely, the Gemma-2-9B-it back-

[1] https://www.neuronpedia.org/

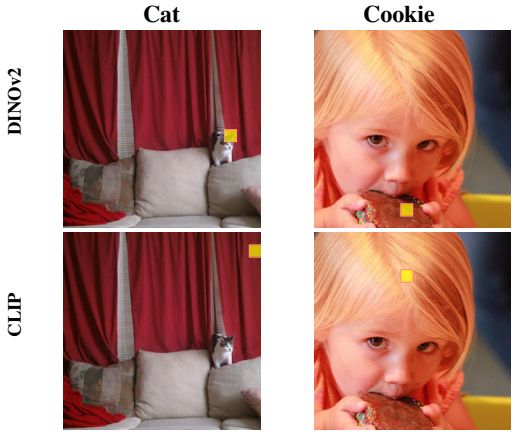


Figure 2. **Spatial Location Analysis** of activating tokens. **Top Row:** DINOv2 precisely localizes visual concepts (cat and cookie). **Bottom Row:** CLIP activations exhibit spatial confusion, often activating tokens far from the relevant object. This high spatial fidelity in DINOv2 is a prerequisite for localized concept steering.

bone shows encoder-dependent behavior where DINOv2 maintains competitive performance both with and without the SAE constraints, whereas CLIP and I-JEPA show varying degrees of performance degradation when the SAE constraints are introduced. Performance tables are provided in Section B. We determine the optimal number of SAE-constrained layers based on a performance-interpretability trade-off analysis, detailed in Section H.

4.2. Spatial Localization Accuracy: CLIP vs. DINOv2 vs. I-JEPA

We observe a large disparity in spatial activation patterns across vision encoders, suggesting that VLM interpretability is primarily influenced by the visual backbone, but also depends on the LLM architecture used for alignment. As shown in Table 1, for the Gemma-2-2B-it backbone, DINOv2 achieves superior spatial fidelity with a localization accuracy of 0.927, whereas I-JEPA and CLIP have scores of 0.702 and 0.201, respectively.

DINOv2 shows good precision; visual tokens at specific indices correspond to the semantic content of their patches. For example, as shown in Figure 2 (top-left) the highlighted visual token specifically activates the "cat" feature, which aligns perfectly with the cat's physical position in the image. This high level of spatial preservation likely stems from DINOv2's local-to-global discriminative objective, which forces the model to encode each patch's local content rather than a global summary. This patch-level locality is what enables our localized steering, allowing for arithmetic interventions on specific patch vectors without triggering unintended global hallucinations.

CLIP exhibits weaker spatial localization compared to DINOv2. Its global contrastive objective incentivizes the model to discard spatial details that do not contribute to global alignment, leading to patch-level feature collapse. Qualitatively, CLIP activations do not preserve patch location; as illustrated in Figure 2 (bottom-left) the highlighted visual token activates the "cat" feature even though the cat is physically located in different patches. I-JEPA occupies a middle ground, retaining more spatial structure than CLIP but failing to achieve the precise feature-to-patch alignment of DINOv2. Its predictive objective requires capturing high-level scene context, causing local features to be slightly smeared to remain consistent with their neighbours.

Table 1. **Spatial Localization Accuracy Scores** (Gemma-2-2B-it backbone). The score reflects the proportion of top-activating tokens that fall within the ground-truth bounding box of the corresponding concept. DINOv2 demonstrates the highest alignment.

Encoder	CLIP	I-JEPA	DINOv2
SLA	0.201	0.702	0.927

We further validate this by ablating spatial localization accuracy across different LLM backbones in Section I. We observe that CLIP's localization accuracy modestly improves with LLaMA-3.1-8B and more noticeably with Gemma-9B, suggesting that cross-modal alignment quality partially depends on the language model size as well. However, DINOv2 significantly outperforms CLIP across all tested configurations, with I-JEPA also achieving higher localization accuracy than CLIP.

4.3. Reconstruction and Sparsity Dynamics

To quantify the alignment between the projected visual tokens and the LLM's internal feature space, we follow Venhoff et al. (2025) using two primary metrics. *SAE Reconstruction Error* measures how accurately visual representations can be reconstructed by the text SAE. For a given layer l, we compute the mean squared error between the visual token $e^l_{v,i}$ and its reconstruction:

$$\text{Err}_l = \frac{1}{N}\sum_{i=1}^{N} ||e^l_{v,i} - \text{SAE}_l(e^l_{v,i})||_2^2 \tag{5}$$

where N is the number of visual tokens and SAE_l represents the SAE at layer l. Then, we analyze *SAE Feature Sparsity*, defined as the average fraction of language features activated by the visual tokens. Let $E_l(x)$ (see Section A) denote the SAE encoder at layer l. Sparsity S_l is calculated as:

$$S_l = \frac{1}{d_{SAE}}\frac{1}{N}\sum_{i=1}^{N} ||E_l(e^l_{v,i})||_0 \tag{6}$$

where $||\cdot||_0$ is the L_0 norm and d_{SAE} is the dictionary size. As textual activations are sparse (Bricken et al., 2023),

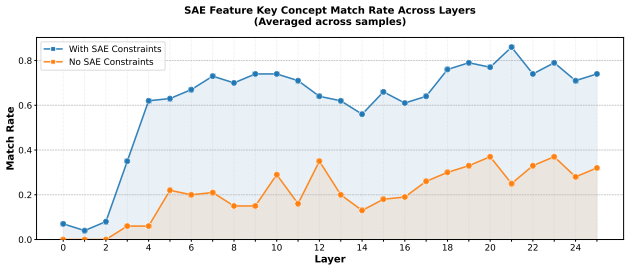


Figure 3. **DINOv2 Match Rate** for visual tokens with and without SAE constraints with Gemma-2-2B-it LLM model.

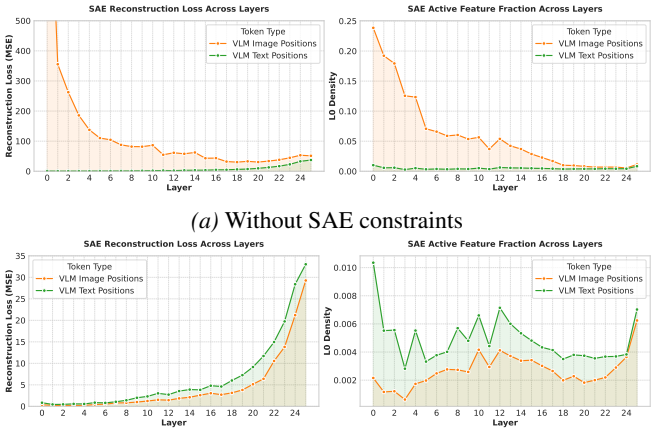
we hypothesize that visual tokens should exhibit similar sparsity.

We analyze these metrics across different LLM backbones. Our results reveal a disparity in how different architectures achieve cross-modal alignment. For the Gemma models (2B and 9B) trained without SAE constraints, we replicate the findings of Venhoff et al. (2025): visual tokens exhibit high reconstruction error and density in early layers, with alignment occurring only in the final layers (Figures 4a and 16a). This "delayed alignment" suggests that standard training struggles to map visual data to the LLM's sparse feature manifold. However, when training with SAE constraints, we observe a shift. As shown in the reconstruction graphs (Figures 4b and 16b), visual tokens achieve immediate convergence, matching the low reconstruction error and low density of text tokens from the very first layer.

When analyzing the LLaMA-3.1-8B-Instruct architecture, we observe a notable divergence in behavior. Unlike Gemma, where unconstrained visual tokens eventually align in deeper layers, the unconstrained LLaMA model fails to converge (Figure 13a); visual tokens remain dense and exhibit high reconstruction error throughout the entire network depth. This indicates that LLaMA's internal representation space is harder for the visual projector to target without guidance, making SAE constraints even more important for the visual features to be interpretable via textual SAEs. Applying our SAE constraints successfully forces alignment, reducing reconstruction error and inducing sparsity (Figure 13b).

4.4. Interpretability Matching Rate

We quantify interpretability via an LLM-Verified Matching Rate (Venhoff et al., 2025). For a given image, we first project its visual tokens through the text SAE. At each layer, we identify the three latents with the highest activation values. To verify their relevance, we employ an oracle LLM, presenting it with both the original image and the textual descriptions associated with these top latents from Neuronpedia. If the LLM confirms that at least one latent description accurately represents a concept visible in the image, we record a successful "match" for that layer.



(a) Without SAE constraints

(b) With SAE constraints

Figure 4. **DINOv2 Reconstruction and Sparsity** across layers, (a) without and (b) with SAE constraints. The gap between image-token (orange) and text-token (green) curves measures alignment where closer curves mean visual tokens inhabit the same sparse manifold as text. Without constraints, curves converge only near the end of the network; with constraints, convergence is immediate.

For Gemma, the baseline (no SAE) matching rate gradually improves in later layers as natural alignment occurs (Figure 3 and Figure 8b). This also holds for different visual encoders such as CLIP and I-JEPA (Figures 9a and 10a). In contrast, for LLaMA, the baseline matching rate remains stagnant and low across all layers, reinforcing that natural alignment is not guaranteed across all LLM architectures (Figure 8c). Adding SAE constraints not only improves the matching rate (e.g., by a threefold increase on average for DINOv2 with Gemma-2-2B-it as shown in Figure 3) but also bridges the gap between different LLMs, achieving high matching rates in LLaMA equivalent to those seen in Gemma. Full results across all visual encoders and LLM backbones are provided in Section C.

To verify that matching rate improvements generalize beyond photo-centric training data, we evaluate VISTA 500 OOD images from DomainNet (Peng et al., 2019). The matching rate is comparable to the in-distribution baseline despite the substantial distribution shift, indicating that VISTA's alignment is a structural property rather than a training-set artifact. Full results are provided in Section K.

5. Insights on Visual Steering

The interpretability transfer established in the previous section raises the question of whether visual tokens truly inhabit the text SAE manifold or whether the alignment is merely an artifact of the matching rate metric. We address this through a causal test. If the alignment is genuine then arithmetic operations using text-SAE concept directions that are mean-

Figure 5. Visual steering with DINOv2, selecting precise patches and changing model's understanding of what is contained in them.

ingful in the text domain, should steer the model's output towards the concept of intervention when applied to visual tokens. The experiments in this section serve as this causal validation. We use object removal and replacement as a controlled testbed because they admit clear success criteria (the target concept either disappears or changes). Section 5.3 (abstract concept manipulation) and Section 5.4 (image clustering by activated concepts) provide complementary downstream validations of the same alignment property.

As our first validation test, we examine whether localized arithmetic interventions on visual tokens produce spatially coherent changes in model output. Unlike standard steering, which alters the targeted concept globally across the entire image, fine-grained steering allows us to modify specific token positions while leaving the rest of the scene untouched.

5.1. Qualitative Results: Spatial Modification

As an example, we use an image of "a girl eating a cookie", first identifying the specific visual tokens (patches) where the cookie is located. We then apply the steering vectors only to those specific patches. As shown in Figure 5 (with more examples in Section G), the results are highly localized; the original description of a "girl eating a chocolate cookie" shifts to a "sandwich" after removing the "chocolate" direction, or "bread" after replacing it with the "bread" direction in the relevant patches. This precise control is uniquely enabled by DINOv2's preservation of spatial alignment. I-JEPA also supports steering, but with lower consistency than DINOv2 (see Table 2 for quantitative evaluation).

In contrast, steering with CLIP-based models is unstable. As shown in Figure 20, since CLIP does not reliably preserve spatial topology, selecting specific patches for steering often yields no effect. While steering can be achieved by applying vectors to **all** tokens, this causes the model to hallucinate. For instance, when attempting to change a "cat on a sofa" to a "dog" using global CLIP steering, the model describes unrelated concepts such as "a person in a blue jacket standing by a fence", elements not present in the image.

5.2. Quantitative Evaluation of Steering

To assess the precision of fine-grained steering, we implement a quantitative evaluation framework. We use an LLM judge to determine whether modifications are localized correctly and whether the global scene context remains intact.

5.2.1. Dataset and Patch Selection

For this evaluation, we sample 100 images from the COCO validation set, spanning diverse categories including animals, food, and vehicles. To ensure the steering task requires high spatial precision, we filter the dataset for images where the target object occupies between 2% and 30% of the total image area. We use COCO ground-truth bounding boxes to select the specific visual patch tokens corresponding to the object targeted for steering.

5.2.2. Scoring Protocol

For each steered image, the judge LLM evaluates the generated text against the original image and the baseline description (output prior to steering). We define *Source Concept* as the class initially present in the baseline and *Target Keyword* as the concept we intend to replace or remove. For removals, Source Concept corresponds to the Target Keyword. For replacements, the Target Keyword is sampled from available classes. Success is determined by the operation type:

1. **Steering Success Check:** The LLM assigns a score of 0 if the steering operation fails:
   - **Replace:** The Target Keyword is missing, **OR** the Source Concept is still present in the text (indicating a conceptual collision where the object was not fully replaced).
   - **Remove:** The Target Keyword is still present in the description.
2. **Baseline Comparison:** If the steering succeeds, the description is compared to the original baseline. If the model successfully modifies the object without adding new environmental details (e.g., background or scene layout) beyond what is present in the baseline description, it receives a Score of 2.
3. **Image Grounding:** If the steered text introduces new descriptive details not found in the baseline text, the LLM verifies them against the source image:
   - If the new details are physically present in the image, it receives a score of 2.
   - If the new details are hallucinated (not in the image), it gets a score of 1.

Table 2. Results of steering with Gemma-2-2B-it. We report the sample counts for **S2** (Strong Success), **S1** (Partial Success), and **S0** (Failure). VISTA with DINOv2 achieves best results.

		Remove Operation				Replace Operation			
Method	**Visual Encoder**	**S2**	**S1**	**S0**	**MS**	**S2**	**S1**	**S0**	**MS**
VisionSAE	CLIP Global	11	84	5	1.06	8	49	43	0.65
	CLIP Local	24	23	53	0.71	23	14	63	0.60
	CLIP Global (Rem+Rep)	—	—	—	—	0	1	99	0.01
	CLIP Local (Rem+Rep)	—	—	—	—	22	2	76	0.46
PatchSAE	CLIP Local	5	55	40	0.65	34	31	35	0.99
Ablation (No SAE)	DINOv2	10	2	88	0.22	3	0	97	0.06
	CLIP	13	20	67	0.46	16	6	78	0.38
	I-JEPA	8	18	74	0.34	20	9	71	0.49
VISTA	DINOv2	47	49	4	**1.43**	59	28	13	**1.46**
	CLIP - Gemma 2B	18	4	78	0.40	7	4	89	0.18
	I-JEPA - Gemma 2B	8	25	67	0.41	17	47	36	0.81

5.2.3. Steering Metric

We quantify steering performance using a weighted mean score, defined as:

$$\text{Mean Score} = \frac{2 \cdot N_{S2} + 1 \cdot N_{S1}}{N_{\text{total}}} \tag{7}$$

where N_{S2} and N_{S1} denote the number of samples classified as Strong Success (score 2) and Partial Success (score 1), respectively, and N_{total} is the total number of evaluation samples. Thus, the score is bounded between 0 and 2.

5.2.4. Results and Analysis

Baselines Table 2 presents the results of our evaluation. We evaluate our approach against two state-of-the-art visual steering baselines: VisionSAE (Pach et al., 2025), which utilizes SAEs trained on global CLIP tokens, and PatchSAE (Lim et al., 2025), which trains SAEs on individual CLIP patches. We specifically focus on SAE-based baselines as they have been shown to outperform traditional activation steering methods. Pach et al. (2025) demonstrated that SAE-based directions surpass the Difference-in-Means (Arditi et al., 2024) approach in both concept insertion and removal.

While effective for broad concept removal, VisionSAE lacks spatial precision. As detailed in Table 2, it frequently alters the background or global context, resulting in a higher rate of score 1 (S1), where hallucinated details are added, compared to score 2 (S2), which requires steering with no alterations beyond the targeted object. This limitation is particularly acute in replacement tasks: because the method relies on global semantic vectors, the model struggles with fine-grained swaps, often superimposing the new concept or modifying unrelated image regions. To address this, in addition to the default (Global) implementation, we evaluate two modifications to this baseline:

1. **Local Patch Steering:** We restrict steering to specific local patches. This yields a lower Mean Score.
2. **Sequential Edit (Remove + Replace):** We attempt to first remove the target concept and subsequently inject the new one. This approach also underperforms the global steering according to the Mean Score.

We also evaluate PatchSAE[2], which theoretically offers better granularity by training on CLIP's patch tokens directly. However, as shown in Table 2, even with patch-level training it underperforms our LLM-based steering which provides better control than steering the visual encoder and is particularly superior in targeted removal, where PatchSAE often hallucinates extra details (as evidenced by its significantly higher S1 rate relative to S2).

Our method achieves substantially higher overall steering quality on both replacement and removal while preserving the scene. Specifically, VISTA with DINOv2 improves the Mean Score from 1.06 to 1.43 on removal (a 35% gain over the strongest vision-only baseline, VisionSAE) and from 0.99 to 1.46 on replacement (a 47% gain over PatchSAE). Furthermore, visual token steering fails without SAE constraints as evidenced by the low Mean Scores in our ablation. The "No SAE" baseline applies the identical steering procedure to VISTA, but on a projector trained without SAE reconstruction loss. Moreover, steering is less reliable when CLIP and I-JEPA encoders are used due to their lower spatial fidelity.

For VISTA, the replace operation generally achieves a higher S2 rate than remove (Table 7). This asymmetry is intrinsic to the steering mechanism as localized subtraction does not inpaint the affected patches but leaves modified embeddings that the LLM must decode coherently. As shown in Figure 5, removing a cat from "a cat sitting on a couch" creates a semantic void at those patch positions, which the LLM fills with a plausible substitute ("a couch with a person sitting on it") to maintain scene coherence. Replacement avoids this issue as injecting a target concept gives the LLM coherent content to describe at the steered patches.

We also demonstrate the robustness of our steering mechanism by evaluating it across different LLM backbones; these ablation results are detailed in Section J. While CLIP's spatial localization accuracy increases with larger LLM backbones as shown in Table 6, this improvement is not sufficient to enable reliable fine-grained steering. As shown in Table 7, CLIP-based models continue to underperform DINOv2 in both object removal and replacement tasks.

We provide details on steering parameters and layers used in Section D. Finally, we validate that VISTA's effect stems from intervention on visual tokens (rather than linguistic hijacking) through a controlled comparison against text-only steering, detailed in Section L.

[2]We adapted the original PatchSAE to enable steering abilities.

(a) **Steering: Sadness.** *Q: Is the dog happy?* **Baseline:** "The dog is happy because it is playing with a soccer ball." **Steered:** "No, the dog is not happy."

(b) **Steering: Sleeping.** *Q: Is the dog awake or sleeping?* **Baseline:** "The dog is awake and standing on the bathroom floor." **Steered:** "The dog is sleeping on the bathroom floor."

(c) **Steering: Danger.** *Q: Is the dog safe to approach?* **Baseline:** "Yes, the dog is safe to approach." **Steered:** "No, the dog shown is not safe to approach."

Figure 6. **Qualitative results.** We steer (a) *Sadness*, (b) *Sleeping*, and (c) *Danger*. In all cases, the steering vector inverts the model's interpretation (Steered) vs the original (Baseline).

5.3. Manipulating High-Level Concepts

We also target high-level concepts. Though abstract notions like emotion or threat lack bounding boxes, they remain anchored to identifiable subjects. We select patches covering the subject exhibiting the concept (highlighted in yellow in Figure 6), and manipulate latents for emotions, physiological state, and threat level. Such concepts are binary in language but lack a fixed visual signature. For instance, "danger" can manifest through aggression, instability, or environmental hazards. This demonstrates another advantage of steering with text-aligned SAEs over visual ones.

5.4. Clustering Images by Activated SAE Concepts

We further demonstrate that the aligned SAE feature space can be utilized to cluster images based on shared semantic attributes. To perform clustering, we first select the top activating tokens from each image, and then identify the images that activate the same concept. As shown in Figure 7, this yields coherent semantic clusters, confirming that visual tokens inherit the language model's structured organization.

6. Limitations and Future Work

Dependence on Pre-trained Text SAEs VISTA requires the availability of SAE trained on the target LLM. We believe that this is becoming less of a constraint as foundational models are increasingly released with paired SAEs.

SAE Concept Coverage Text SAEs can only interpret visual concepts with corresponding features in the LLM's representation. Rare concepts may activate broader, less precise latents. VISTA inherits future improvements "for free" by swapping in larger SAEs.

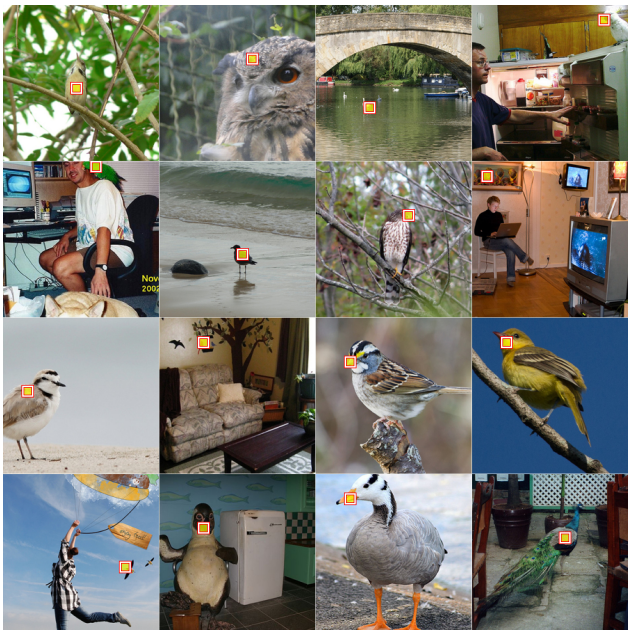


Figure 7. **Image Clustering.** Sample of images that activate the latent "terms and concepts related to birds". The patch that activates the latent is shown in yellow with red bounding box.

Future Work VISTA establishes the cross-modal alignment necessary for a broader research agenda we leave to future work: (i) feature circuit analysis (Marks et al., 2025) to trace the causal effect of visual inputs through the LLM's internal computation; (ii) hallucination detection (Abdaljalil et al., 2025) by identifying when visual tokens trigger ungrounded textual latents. Each requires substantial future work, but VISTA provides the foundational alignment they require. Also, VISTA's interpretability transfer succeeds across all encoders; only spatial precision degrades with weaker encoders. Incorporating self-supervised localized losses into contrastive encoders could mitigate this.

7. Conclusion

In this work, we demonstrated that interpretability is a transferable property across modalities. By enforcing SAE constraints during the alignment of vision encoders to LLMs, we mapped visual tokens into a pre-existing, labeled linguistic feature space, eliminating the computational and labeling costs of training vision-specific SAEs. Our evaluation revealed that the success of localized interpretability and steering is coupled with the spatial fidelity of the vision backbone; DINOv2's local objective provides the necessary precision for this, significantly outperforming contrastive models like CLIP. This "interpretability transfer" paradigm provides a scalable path for understanding and controlling complex multimodal systems.

Acknowledgements

We'd like to gratefully acknowledge Lambda Labs compute support and the support from University of Bath for the studentship.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. We specifically intend to enhance the interpretability aspect. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Abdaljalil, S., Pallucchini, F., Seveso, A., Kurban, H., Mercorio, F., and Serpedin, E. SAFE: A sparse autoencoder-based framework for robust query enrichment and hallucination mitigation in LLMs. In Christodoulopoulos, C., Chakraborty, T., Rose, C., and Peng, V. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 9335–9346, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.496.

Arditi, A., Obeso, O., Syed, A., Paleka, D., Panickssery, N., Gurnee, W., and Nanda, N. Refusal in language models is mediated by a single direction. In *NeurIPS*, 2024.

Assran, M., Duval, Q., Misra, I., Bojanowski, P., Vincent, P., Rabbat, M. G., LeCun, Y., and Ballas, N. Self-supervised learning from images with a joint-embedding predictive architecture. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629, 2023.

Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N., Anil, C., Denison, C., Askell, A., Lasenby, R., Wu, Y., Kravec, S., Schiefer, N., Maxwell, T., Joseph, N., Hatfield-Dodds, Z., Tamkin, A., Nguyen, K., McLean, B., Burke, J. E., Hume, T., Carter, S., Henighan, T., and Olah, C. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023.

Daujotas, G. Case study: Interpreting, manipulating, and controlling clip with sparse autoencoders, 2024.

Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., Hatfield-Dodds, Z., Lasenby, R., Drain, D., Chen, C., Grosse, R., McCandlish, S., Kaplan, J., Amodei, D., Wattenberg, M., and Olah, C. Toy models of superposition. *Transformer Circuits Thread*, 2022.

Fu, C., Chen, P., Shen, Y., Qin, Y., Zhang, M., Lin, X., Yang, J., Zheng, X., Li, K., Sun, X., Wu, Y., Ji, R., Shan, C., and He, R. MME: A comprehensive evaluation benchmark for multimodal large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025.

He, Z., Shu, W., Ge, X., Chen, L., Wang, J., Zhou, Y., Liu, F., Guo, Q., Huang, X., Wu, Z., et al. Llama scope: Extracting millions of features from llama-3.1-8b with sparse autoencoders. *arXiv preprint arXiv:2410.20526*, 2024.

Huben, R., Cunningham, H., Smith, L. R., Ewart, A., and Sharkey, L. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Hudson, D. A. and Manning, C. D. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6700–6709, 2019.

Joseph, S., Suresh, P., Goldfarb, E., Hufe, L., Gandelsman, Y., Graham, R., Bzdok, D., Samek, W., and Richards, B. A. Steering clip's vision transformer with sparse autoencoders. 2025.

Li, Y., Du, Y., Zhou, K., Wang, J., Zhao, X., and Wen, J.-R. Evaluating object hallucination in large vision-language models. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 292–305, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.20.

Lieberum, T., Rajamanoharan, S., Conmy, A., Smith, L., Sonnerat, N., Varma, V., Kramár, J., Dragan, A., Shah, R., and Nanda, N. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2, 2024.

Lim, H., Choi, J., Choo, J., and Schneider, S. Sparse autoencoders reveal selective remapping of visual concepts during adaptation. In *The Thirteenth International Conference on Learning Representations*, 2025.

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P. Microsoft coco: Common objects in context, 2015.

Liu, H., Li, C., Wu, Q., and Lee, Y. J. Visual instruction tuning. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 34892–34916. Curran Associates, Inc., 2023.

Marks, S., Rager, C., Michaud, E. J., Belinkov, Y., Bau, D., and Mueller, A. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Merullo, J., Castricato, L., Eickhoff, C., and Pavlick, E. Linearly mapping from image to text space. In *In The Eleventh International Conference on Learning Representations*, 2023.

Oquab, M., Darcet, T., Moutakanni, T., Vo, H. V., Szafraniec, M., Khalidov, V., Fernandez, P., HAZIZA, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.-Y., Li, S.-W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A., and Bojanowski, P. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. Featured Certification.

Pach, M., Karthik, S., Bouniot, Q., Belongie, S., and Akata, Z. Sparse autoencoders learn monosemantic features in vision-language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., and Wang, B. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1406–1415, 2019.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

Rajamanoharan, S., Lieberum, T., Sonnerat, N., Conmy, A., Varma, V., Kramár, J., and Nanda, N. Jumping ahead: Improving reconstruction fidelity with jumprelu sparse autoencoders, 2024.

Rimsky, N., Gabrieli, N., Schulz, J., Tong, M., Hubinger, E., and Turner, A. Steering llama 2 via contrastive activation addition. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.828.

Sharma, P., Ding, N., Goodman, S., and Soricut, R. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of ACL*, 2018.

Turner, A. M., Thiergart, L., Udell, D., Leech, G., Mini, U., and MacDiarmid, M. Activation addition: Steering language models without optimization, 08 2023.

Venhoff, C., Khakzar, A., Joseph, S., Torr, P., and Nanda, N. How visual representations map to language feature space in multimodal llms. 2025.

Wang, J., Ge, X., Shu, W., Tang, Q., Zhou, Y., He, Z., and Qiu, X. Towards universality: Studying mechanistic similarity across language model architectures. In *The Thirteenth International Conference on Learning Representations*, 2025.

Supplementary Material

Table of Contents

(A) SPARSE AUTOENCODERS 12

(B) PERFORMANCE 13

(C) MATCH RATE FOR DIFFERENT LLM BACKBONES AND VISUAL ENCODERS 14

(D) IMPLEMENTATION DETAILS 17

(E) RECONSTRUCTION AND SPARSITY COMPARISONS 17

(F) ACTIVE VISUAL TOKENS ACROSS LAYERS 22

(G) QUALITATIVE EXAMPLES OF VISUAL STEERING WITH CLIP AND DINOV2 23

(H) PERFORMANCE-INTERPRETABILITY TRADE-OFF 23

(I) SPATIAL LOCALIZATION WITH DIFFERENT LLM BACKBONES 25

(J) STEERING WITH DIFFERENT LLM BACKBONES 25

(K) OOD ROBUSTNESS ON DOMAINNET 26

(L) VISUAL VS TEXT-ONLY STEERING COMPARISON 26

A. Sparse Autoencoders

Internal representations in large language models are difficult to interpret because they are often polysemantic. This means a single neuron might activate for many unrelated concepts, a phenomenon known as superposition (Elhage et al., 2022; Bricken et al., 2023). To solve this, sparse autoencoders (SAEs) are used to break down these polysemantic activations into monosemantic features.

SAEs take an activation vector x from the model, such as the residual stream, and processes it through two stages - encoding and decoding. The encoder $E(\cdot)$ maps the input into a high-dimensional, sparse latent space $E(x) \in \mathbb{R}^{d_{SAE}}$. This space is "overcomplete" meaning it has many more dimensions than the original input, typically 10 to 100 times more. The encoder uses a ReLU activation, or other alternatives like JumpReLU activation (Rajamanoharan et al., 2024), to ensure that feature values are non-negative:

$$E(x) = \text{ReLU}(W_{enc}(x - b_{dec}) + b_{enc}) \tag{8}$$

where W_{enc} represents the encoder weights and b_{enc}, b_{dec} are learned biases.

The decoder $D(\cdot)$ then attempts to reconstruct the original activation x by linearly combining the discovered sparse features $E(x)$:

$$\hat{x} = D(E(x)) = W_{dec}E(x) + b_{dec} \tag{9}$$

Each column in the decoder weight matrix W_{dec} often represents a monosemantic feature that corresponds to a specific, human-interpretable concept.

The SAE is trained to minimize a dual-objective loss function consisting of a reconstruction error and an L_1 sparsity penalty:

$$\mathcal{L} = |x - \hat{x}|_2^2 + \lambda |E(x)|_1 \tag{10}$$

The first term ensures that the SAE preserves the information content of the original activations, requiring the reconstructed version $\hat{x}$ to be as close to the original x as possible. The second term, scaled by the hyperparameter λ, forces the model to represent the data using the smallest possible subset of features. By enforcing this sparsity, the SAE is forced to disentangle the overlapping concepts within the residual stream into discrete, monosemantic latent directions.

B. Performance

Table 3. Performance comparison of different visual encoders with and without SAE constraints on Gemma-2-2B-it.

		CLIP-Gemma-2-2B		DINOv2-Gemma-2-2B		I-JEPA-Gemma-2-2B	
Benchmark	**Metric**	**No SAE**	**SAE**	**No SAE**	**SAE**	**No SAE**	**SAE**
MME	Cognition	243.92	251.42	261.78	226.42	218.21	219.64
	Perception	993.73	942.89	942.13	881.30	907.13	719.31
LLaVA Bench	Conversation	53.5	47.3	52.5	59.7	49.6	41.5
	Complex	32.1	32.6	36.2	33.3	30.6	27.9
	Detailed	30.0	24.0	29.4	30.4	26.7	21.8
	All	38.6	34.5	39.2	41.1	35.6	30.3
GQA	Exact Match	42.95	41.00	43.50	42.10	40.46	39.20
POPE	Accuracy	81.6	79.3	76.63	76.33	74.1	73.57
	Precision	86.8	94.2	88.16	79.74	70.2	83.19
	Recall	74.53	62.4	61.53	70.6	83.6	59.07
	F1-score	80.2	75.1	72.48	74.89	76.4	69.08

Table 4. Performance comparison of different visual encoders with and without SAE constraints on LLaMA-3.1-8B-Instruct.

		CLIP-LLaMA-3.1-8B		DINOv2-LLaMA-3.1-8B		I-JEPA-LLaMA-3.1-8B	
Benchmark	**Metric**	**No SAE**	**SAE**	**No SAE**	**SAE**	**No SAE**	**SAE**
MME	Cognition	260.71	295.00	286.07	302.50	238.93	257.50
	Perception	996.06	1062.32	1015.27	1175.09	883.94	972.33
LLaVA Bench	Conversation	59.9	67.9	56.8	69.1	44.5	60.7
	Complex	35.9	33.5	25.9	29.5	33.6	31.5
	Detailed	27.6	31.7	27.3	30.1	19.3	31.0
	All	40.9	44.1	36.6	42.6	32.3	41.0
GQA	Exact Match	41.31	44.47	41.03	51.61	39.64	44.46
POPE	Accuracy	76.27	79.23	76.63	84.17	75.60	77.73
	Precision	76.09	78.31	72.79	87.99	71.79	77.19
	Recall	76.96	80.87	85.07	79.13	84.33	78.73
	F1-score	76.52	79.57	78.45	83.33	77.56	77.95

Table 5. Performance comparison of different visual encoders with and without SAE constraints on Gemma-2-9B-it.

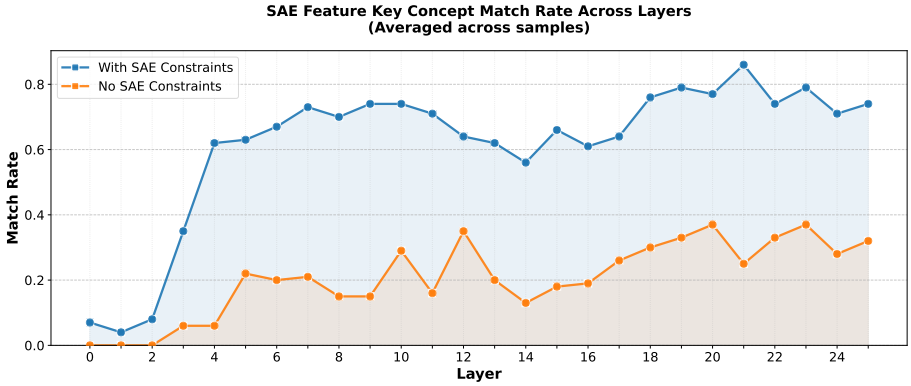
		CLIP-Gemma-2-9B		DINOv2-Gemma-2-9B		I-JEPA-Gemma-2-9B	
Benchmark	**Metric**	**No SAE**	**SAE**	**No SAE**	**SAE**	**No SAE**	**SAE**
MME	Cognition	310.71	259.29	275.36	295.36	248.57	301.07
	Perception	1198.37	1042.00	1170.90	1085.64	1063.82	1032.49
LLaVA Bench	Conversation	63.4	65.4	58.9	70.2	52.2	47.4
	Complex	34.1	34.0	30.7	32.0	33.1	34.7
	Detailed	33.8	32.4	33.1	32.3	27.8	20.8
	All	43.9	42.7	41.0	44.2	37.7	34.2
GQA	Exact Match	49.62	44.16	47.43	47.27	44.85	42.44
POPE	Accuracy	83.13	76.40	83.77	83.17	78.73	77.80
	Precision	85.55	75.19	87.32	87.95	79.40	76.44
	Recall	79.73	78.80	79.16	76.87	77.60	81.68
	F1-score	82.54	76.95	83.04	82.03	78.49	78.97

We evaluate the impact of SAE constraints on projector training across four benchmarks: MME (perception and cognition) (Fu et al., 2025), GQA (compositional reasoning) (Hudson & Manning, 2019), POPE (hallucination) (Li et al., 2023), and LLaVA-Bench (instruction following) (Liu et al., 2023). As shown in Table 3, enforcing interpretability constraints typically induces a marginal accuracy regression in smaller models. We hypothesize that forcing visual tokens to align strictly with text features taxes the limited representational capacity of smaller architectures. However, this effect is not universal; for instance, the DINOv2 encoder with SAE alignment yields slight improvements on LLaVA-Bench and POPE F1 scores.

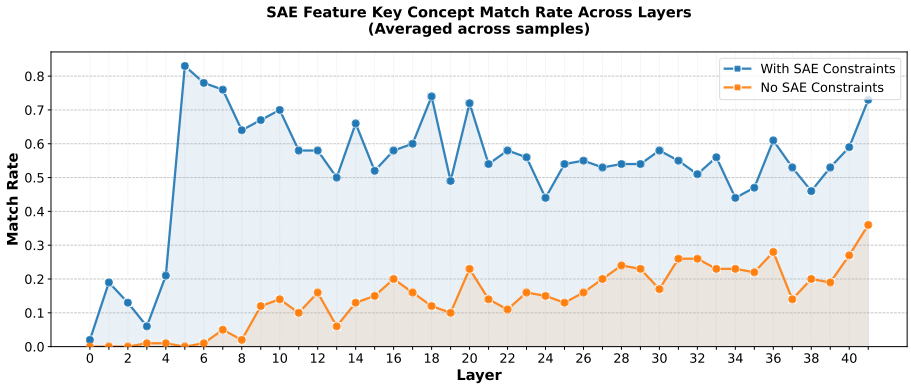
Scaling the LLM backbone fundamentally alters how the model handles these interpretability constraints (Tables 4 and 5). For the LLaMA-3.1-8B-Instruct architecture, enforcing SAE constraints results in performance gains across the majority of metrics for all visual encoders. This suggests that for certain larger architectures, SAE interpretability constraints can act as a beneficial regularizer for cross-modal alignment. Conversely, the Gemma-2-9B-it backbone demonstrates highly encoder-dependent results. While DINOv2 combined with the SAE constraint maintains competitive performance,

improving in cognition and LLaVA-Bench while slightly regressing in perception, the CLIP encoder suffers a degradation across almost all metrics when the SAE is introduced. The addition of the SAE to the I-JEPA encoder yields mixed results across the benchmarks for Gemma-2-9B-it backbone.

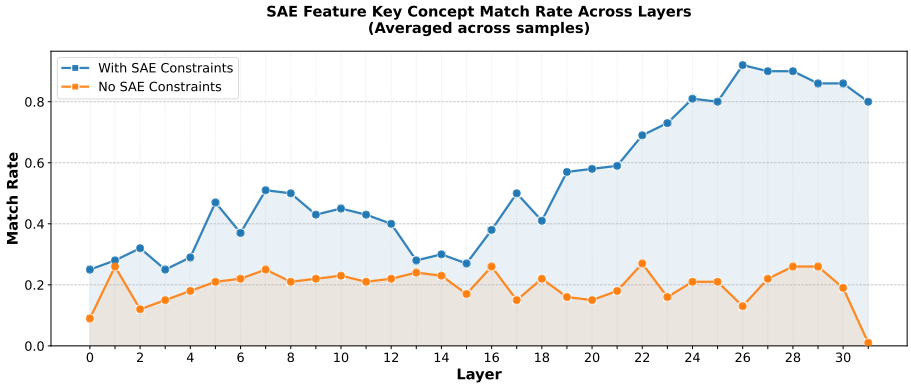
C. Match Rate for Different LLM Backbones and Visual Encoders



(a) DINOv2 match rate for visual tokens with and without SAE constraints with Gemma-2-2B-it LLM model.

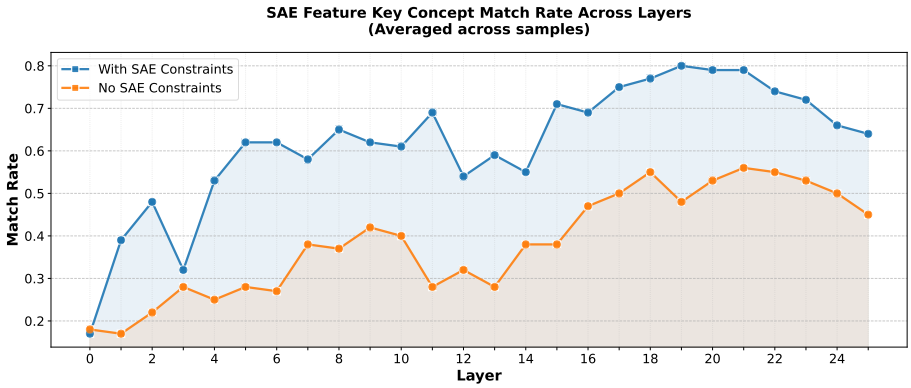


(b) DINOv2 match rate for visual tokens with and without SAE constraints with Gemma-2-9B-it model.

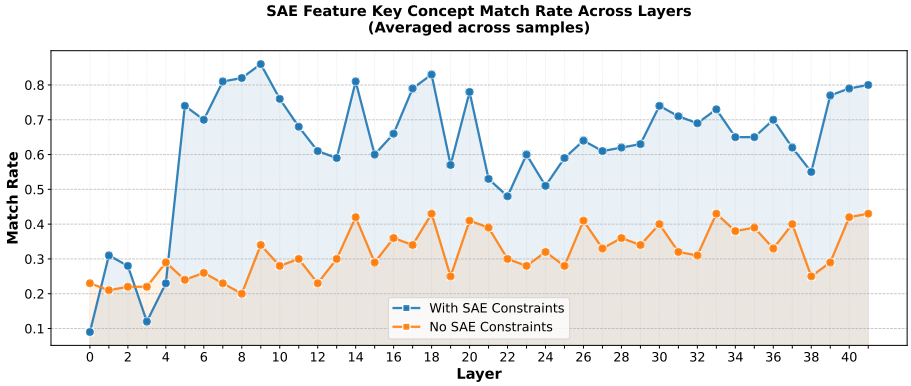


(c) DINOv2 match rate for visual tokens with and without SAE constraints with LLaMA-3.1-8B-Instruct model.

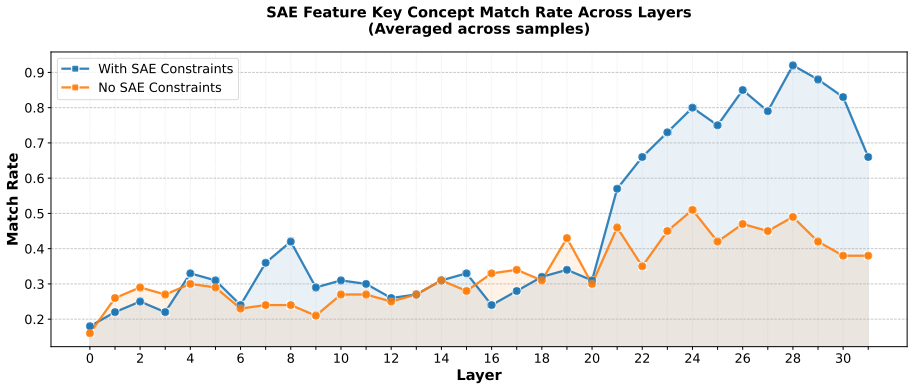
Figure 8. Matching rate for DINOv2 visual encoder.



(a) CLIP match rate for visual tokens with and without SAE constraints with Gemma-2-2B-it LLM model.

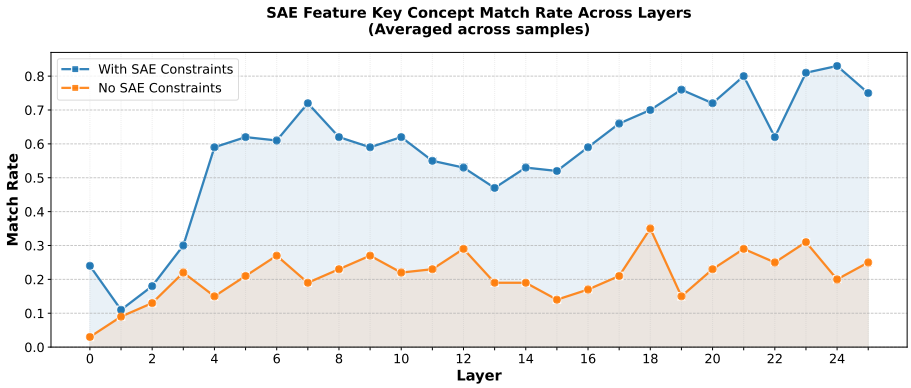


(b) CLIP match rate for visual tokens with and without SAE constraints with Gemma-2-9B-it model.

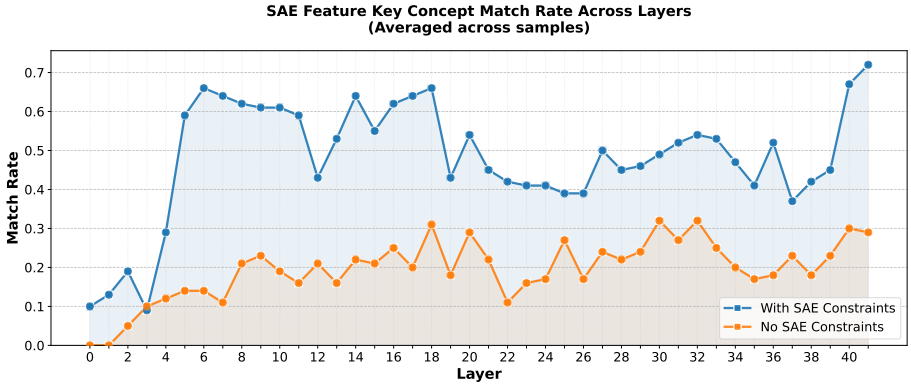


(c) CLIP match rate for visual tokens with and without SAE constraints with LLaMA-3.1-8B-Instruct model.

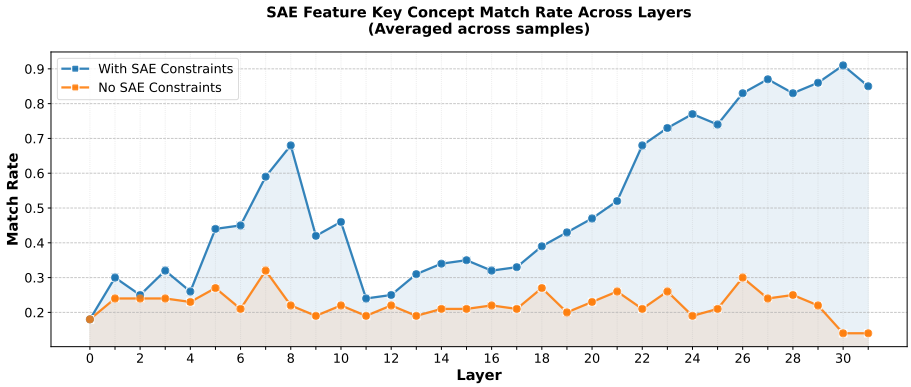
Figure 9. Matching rate for CLIP visual encoder.



(a) I-JEPA match rate for visual tokens with and without SAE constraints with Gemma-2-2B-it LLM model.



(b) I-JEPA match rate for visual tokens with and without SAE constraints with Gemma-2-9B-it model.



(c) I-JEPA match rate for visual tokens with and without SAE constraints with LLaMA-3.1-8B-Instruct model.

Figure 10. Matching rate for I-JEPA visual encoder.

D. Implementation Details

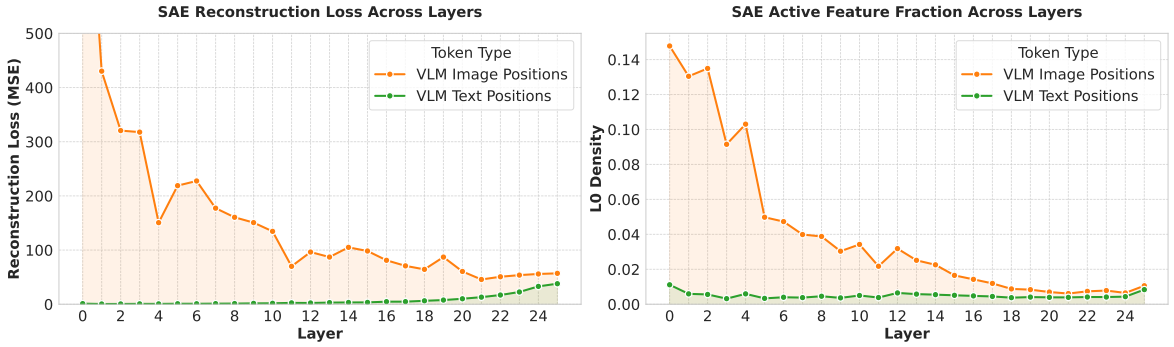
Models and Architecture Our experiments utilize pre-trained SAEs from (Lieberum et al., 2024) and LLaMA Scope (He et al., 2024). We analyze the residual stream of Gemma 2 (2B and 9B) using $16k$-width SAEs, and for LLaMA, we employ the $8\times$ expansion factor SAEs.

For the visual backbone, We employ visual encoders from three distinct model families: **CLIP-ViT-L/14** (Radford et al., 2021), **DINOv2-ViT-L/14** (Oquab et al., 2024), and **I-JEPA-ViT-H/14** (Assran et al., 2023). Each optimizes for different objectives: contrastive image-text alignment, self-supervised visual feature learning, and predictive visual modeling, respectively. All three models use a patch size of 14×14 and process 224×224 inputs into 256 visual patch tokens. While CLIP-ViT-L/14 and DINOv2-ViT-L/14 append an additional [CLS] token (resulting in 257 total tokens), I-JEPA-ViT-H/14 outputs only the 256 patch tokens. This architectural consistency enables direct comparison within our unified representation framework.

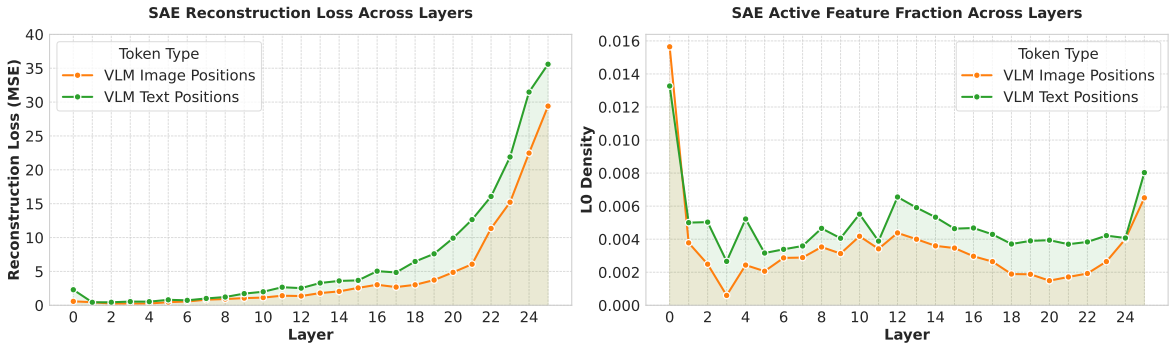
Training and Computational Cost The visual projector is trained in two stages while keeping the vision encoder and LLM frozen. The total computational requirement for full pre-training and fine-tuning is approximately 200 GPU hours for smaller LLM models and 290 GPU hours for larger architectures. We utilize A100 GPUs. For evaluations regarding Matching Rate, Reconstruction, and Sparsity, we utilize 1000 samples and employ the Gemini-2.5-flash-lite model. For Spatial Localization Accuracy calculation, we use 380 images sampled from the COCO dataset.

Steering and Intervention Parameters We perform visual steering by intervening spatially on selected visual patches and temporally at every decoding step in the first 3 layers for Gemma models and the first 2 layers of LLaMA model, found empirically most effective. For both removal and replacement operations, we use a steering coefficient of 10. When training the projector with VISTA's SAE constraints, we apply the reconstruction loss to the first 5 layers of the LLM based on the performance-interpretability trade-off analysis in Section H.

E. Reconstruction and Sparsity Comparisons

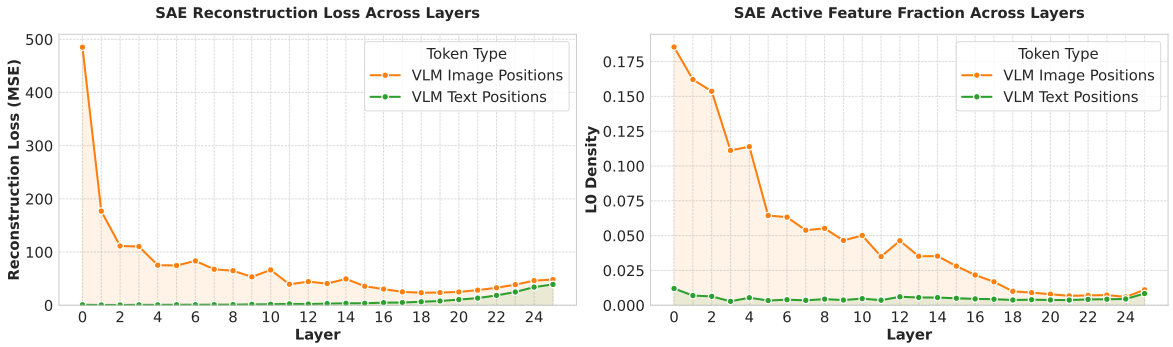


(a) CLIP with Gemma-2-2B-it reconstruction without SAE constraints.

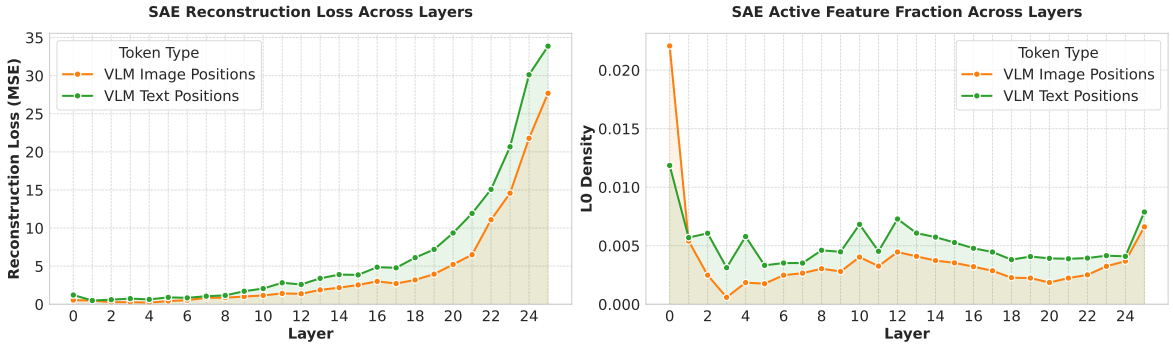


(b) CLIP with Gemma-2-2B-it reconstruction with SAE constraints.

Figure 11. CLIP with Gemma-2-2B-it reconstruction and sparsity.

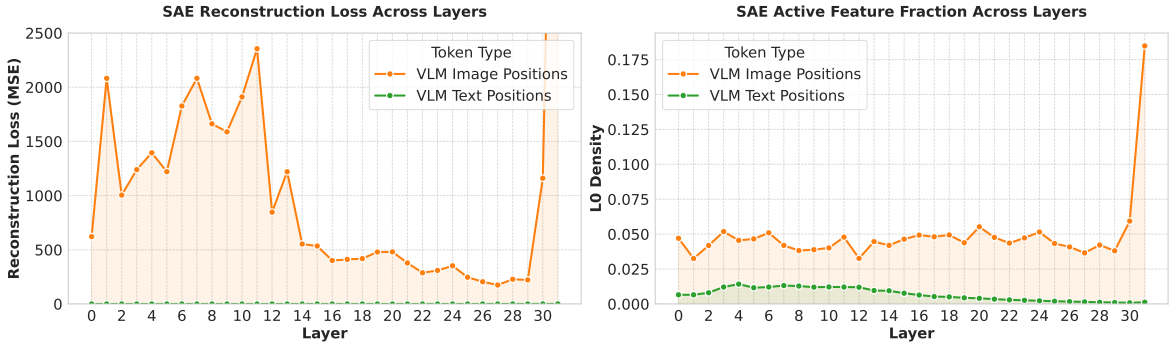


(a) I-JEPA with Gemma-2-2B-it reconstruction without SAE constraints.

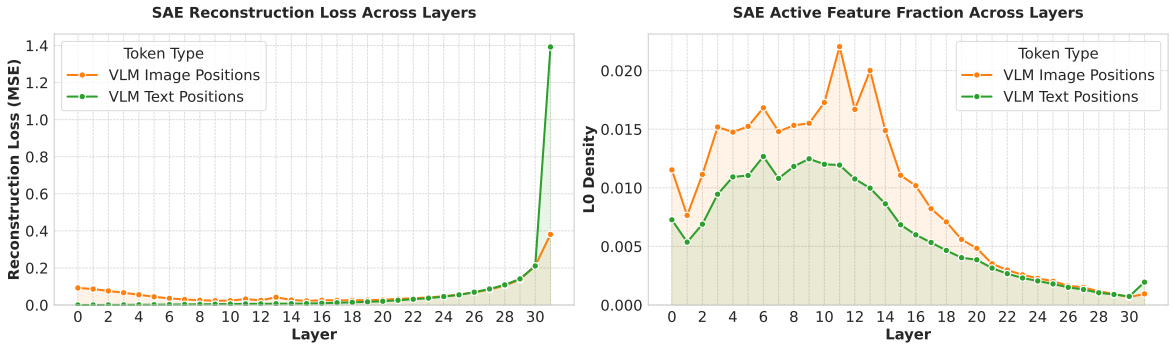


(b) I-JEPA with Gemma-2-2B-it reconstruction with SAE constraints.

Figure 12. I-JEPA with Gemma-2-2B-it reconstruction and sparsity.

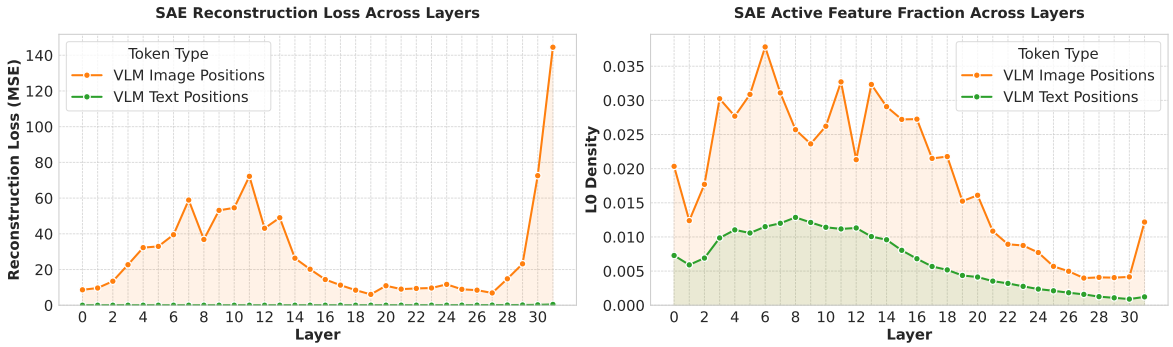


(a) DINOv2 with LLaMA-3.1-8B-Instruct reconstruction without SAE constraints

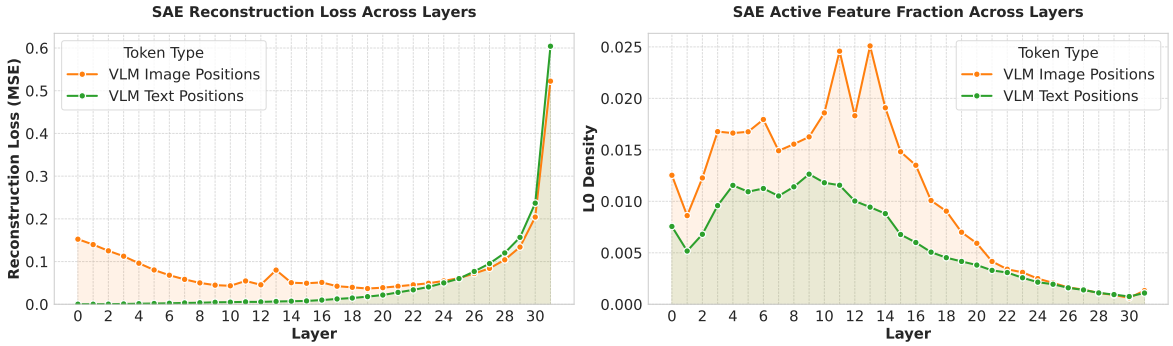


(b) DINOv2 with LLaMA-3.1-8B-Instruct reconstruction with SAE constraints.

Figure 13. DINOv2 with LLaMA-3.1-8B-Instruct reconstruction and sparsity.

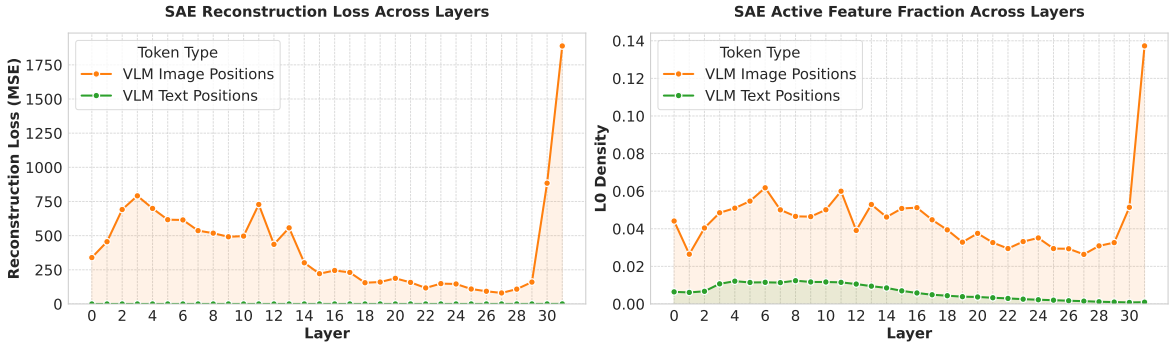


(a) CLIP with LLaMA-3.1-8B-Instruct reconstruction without SAE constraints

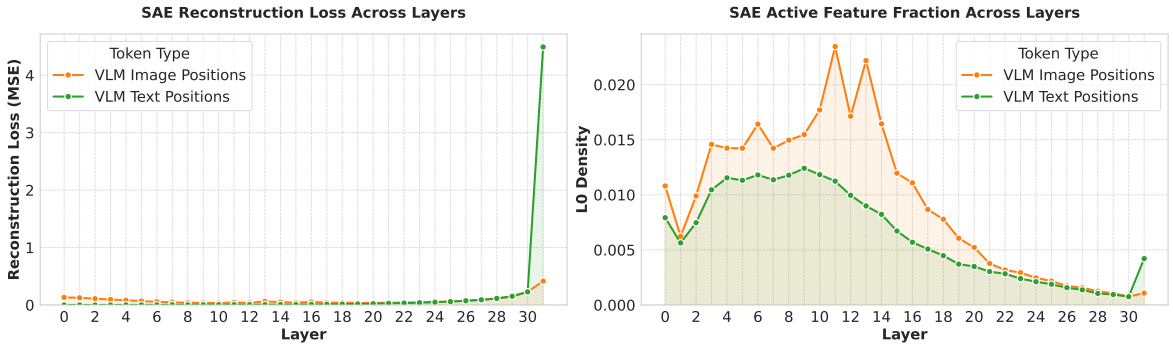


(b) CLIP with LLaMA-3.1-8B-Instruct reconstruction with SAE constraints.

Figure 14. CLIP with LLaMA-3.1-8B-Instruct reconstruction and sparsity.

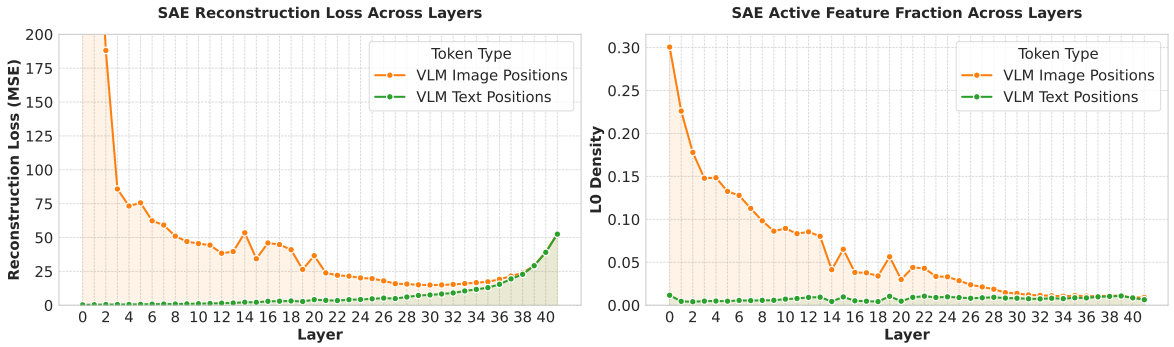


(a) I-JEPA with LLaMA-3.1-8B-Instruct reconstruction without SAE constraints

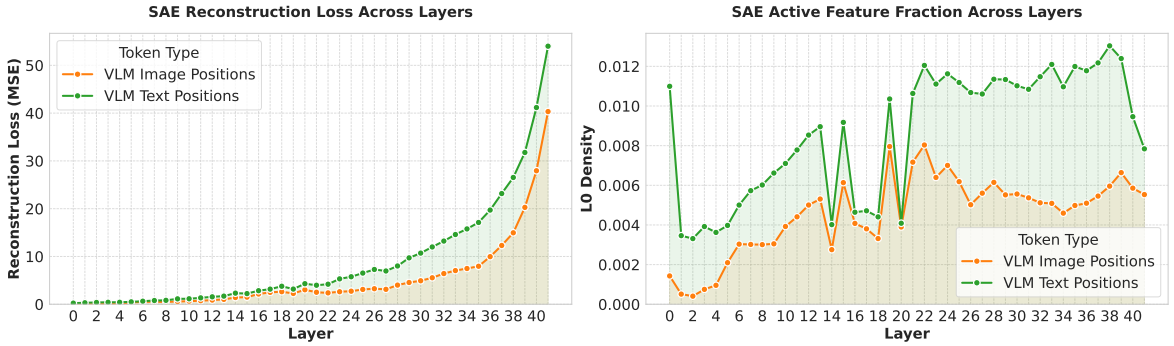


(b) I-JEPA with LLaMA-3.1-8B-Instruct reconstruction with SAE constraints.

Figure 15. I-JEPA with LLaMA-3.1-8B-Instruct reconstruction and sparsity.

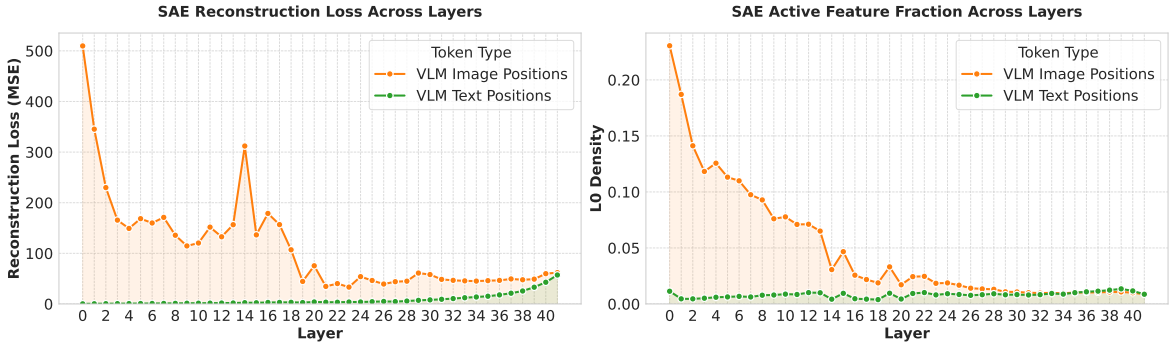


(a) DINOv2 with Gemma-2-9B-it reconstruction without SAE constraints

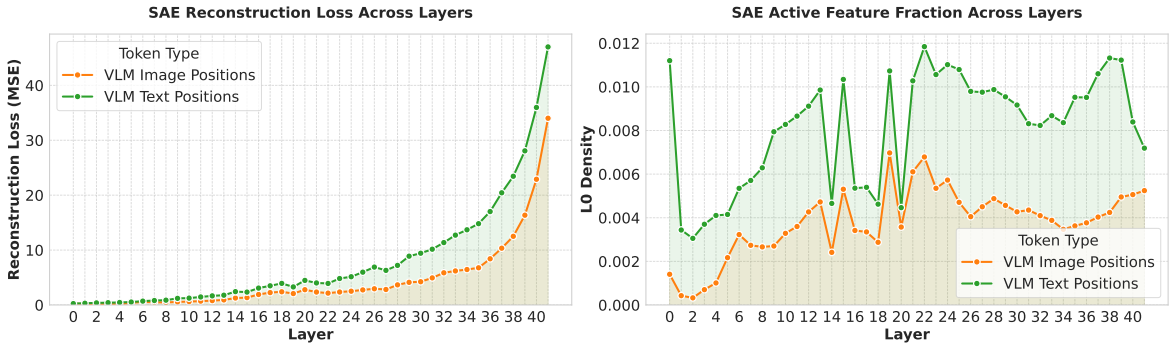


(b) DINOv2 with Gemma-2-9B-it reconstruction with SAE constraints.

Figure 16. DINOv2 with Gemma-2-9B-it reconstruction and sparsity.

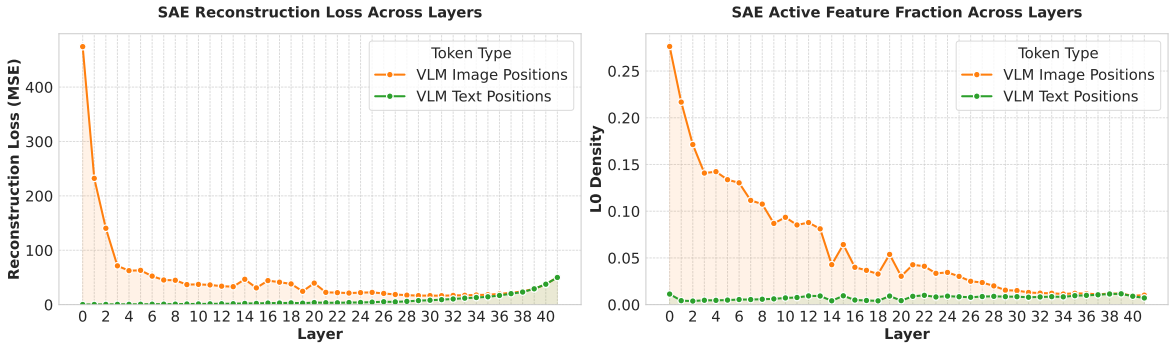


(a) CLIP with Gemma-2-9B-it reconstruction without SAE constraints

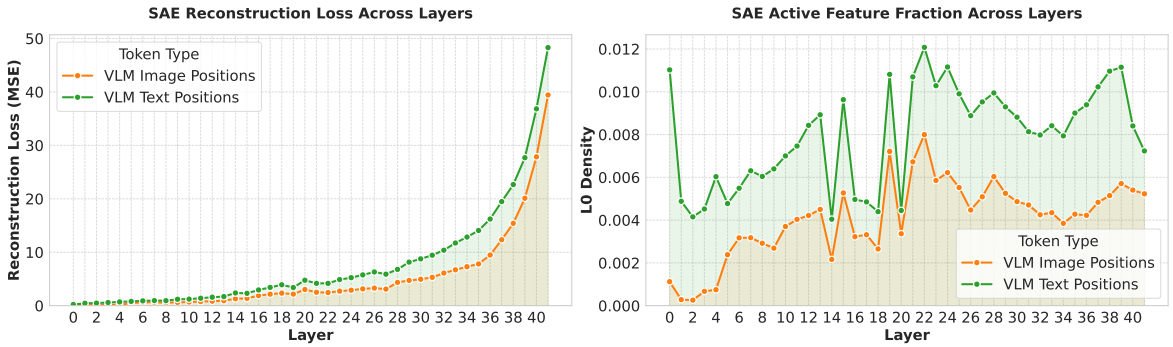


(b) CLIP with Gemma-2-9B-it reconstruction with SAE constraints.

Figure 17. CLIP with Gemma-2-9B-it reconstruction and sparsity.



(a) JEPA with Gemma-2-9B-it reconstruction without SAE constraints

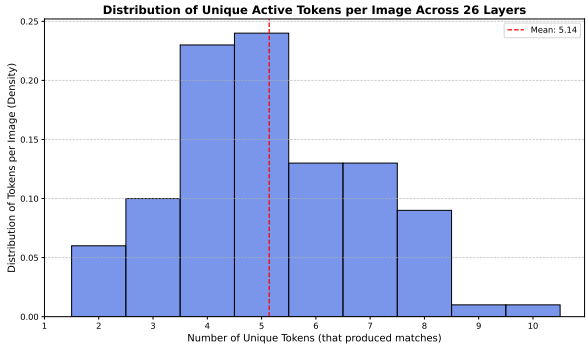


(b) JEPA with Gemma-2-9B-it reconstruction with SAE constraints.

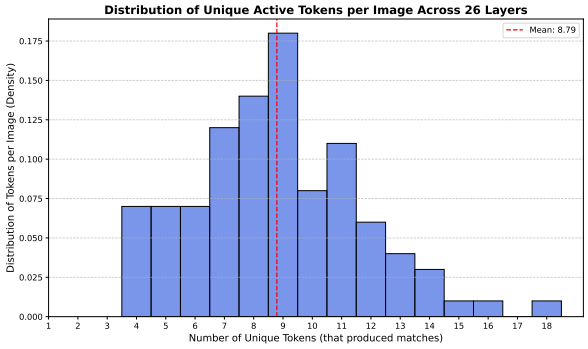
Figure 18. JEPA with Gemma-2-9B-it reconstruction and sparsity.

F. Active Visual Tokens across Layers

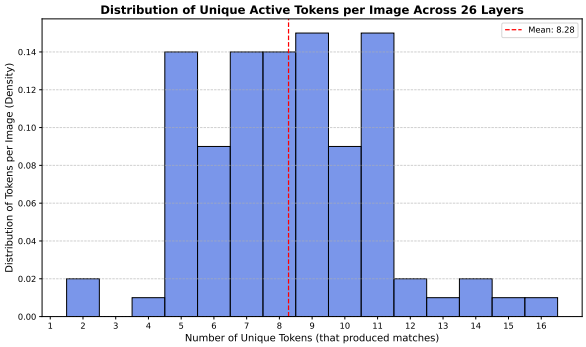
We analyse for each image whether maximally activating tokens are different or not across layers. We noticed that using CLIP visual model the number of active visual features across layers is very small, with the mean around 5. On the contrary, for I-JEPA and DINOv2 we observe that the mean of the activating tokens across layers for an image is on average around 8 and 9 respectively meaning that with I-JEPA and DINOv2 more concepts per image are activated. This is shown in Figure 19.



(a) CLIP average activating tokens across layers with Gemma-2-2B-it.



(b) DINOv2 average activating tokens across layers with Gemma-2-2B-it.



(c) I-JEPA average activating tokens across layers with Gemma-2-2B-it.

Figure 19

G. Qualitative Examples of Visual Steering with CLIP and DINOv2

Q: What is shown in this image?
Original: The image shows a cat sitting on a red and white striped curtain. The cat is looking at the camera.
Remove "cat": The image shows a cat sitting on a red and white striped curtain. The cat is looking at the camera.
Replace with "dog": The image shows a cat sitting on a red and white striped curtain. The cat is looking at the camera.

Q: Describe the time of the day in the image.
Original: The image shows a motorcycle parked on the side of the road at night..
Remove "night": The image shows a motorcycle parked on the side of the road at night.
Replace with "morning": The image shows a motorcycle parked on the side of the road at night.

Q: What is shown in this image?
Original: Bird is sitting on a branch. Bird is a green color.
Remove "bird": Bird is sitting on a branch. Bird is a green color.
Replace with "cat": Bird is sitting on a branch. Bird is a green color.

Q: What is shown in this image?
Original: A girl is eating a donut. She is sitting on a chair and has a big smile on her face.
Remove "chocolate": The image shows a little girl with a big smile and a big cupcake. She is sitting on a chair and eating a cupcake.
Replace with "bread": The image shows a girl with a big smile and a little girl with a big smile. The girl is eating a donut.

Figure 20. Qualitative example of visual steering with CLIP visual encoder and Gemma-2-2B-it LLM, selecting precise patches in the image and changing model's understanding of what is contained in them. Local steering with CLIP visual encoder is not effective.

Q: What is shown in this image?
Original: The image shows a cat sitting on a couch with a pillow. The cat is looking at the camera.
Remove "cat": The image shows a couch with a person sitting on it. The person is wearing a white shirt and has a red scarf around their neck.
Replace with "dog": The image shows a dog sitting on a couch with a pillow. The dog is a white and tan mix.

Q: Describe the time of the day in the image.
Original: The image is taken at night.
Remove "night": The time of day in the image is evening. The sun is setting in the background, and the sky is a mix of orange, red, and purple hues. Streetlights are also visible.
Replace with "morning": The image is taken in the early morning, as evidenced by the light coming from the streetlights and the fact that the sun is not yet fully risen. The time of day is around 6:00 AM.

Q: What is shown in this image?
Original: This image shows a bird sitting on a branch in a forest. The bird is looking at the camera.
Remove "bird": This image shows a fish that is hanging from a tree branch. The fish is looking down at the ground.
Replace with "cat": This image shows a cat sitting on a branch in a forest. The cat is looking at the camera.

Q: What is shown in this image?
Original: The image shows a girl eating a chocolate chip cookie. The girl is holding a cookie in her hand and is eating it.
Remove "chocolate": This image shows a girl with a large amount of food on her face. The girl is eating a sandwich.
Replace with "bread": This image shows a girl with shirt. She is holding a piece of bread in her hand.

Figure 21. Qualitative example of visual steering with DINOv2 visual encoder and Gemma-2-2B-it LLM, selecting precise patches in the image and changing model's understanding of what is contained in them. Local steering with DINOv2 visual encoder is effective.

H. Performance - Interpretability Trade-off

We compare how additional constraints on SAE layers affect both interpretability and benchmark performance. To aggregate interpretability metrics, we averaged the matching rate across all evaluated layers. For performance, we standardized all benchmarks to a 0-100 percentage scale. While GQA, POPE, and LLaVA-Bench natively occupy this range, the MME total score is normalized against its theoretical maximum of 2800 points. Furthermore, we select the POPE F1-score over accuracy to robustly penalize hallucination bias. The final score is reported as the arithmetic mean of these standardized values. We use DINOv2 with Gemma-2-2B-it for this ablation as it is our primary configuration throughout the paper.

Running the full SAE-layer sweep across every encoder - LLM combination and across all layer subsets of the 26-layer model is computationally prohibitive, so we restrict our analysis to representative subsets rather than an exhaustive search.

As shown in Figures 22 and 23 increasing the number of SAE-constrained layers improves interpretability up to a threshold. While applying SAEs to all layers causes a slight regression in interpretability compared to the peak, specifically reducing it in the last layers, it remains superior to the unconstrained baseline. Regarding performance, the configuration that includes SAE constraints on the first 5 layers yields results closest to the baseline (no SAE constraints) while preserving a good level of interpretability.

Note that this trade-off is architecture-dependent; for instance, adding SAEs in the LLaMA-based model actually improves performance compared to the slight decrease seen in Gemma-2-2B, and the gap is reduced with Gemma-2-9B as seen in Section B. We therefore adopt the five-layer configuration as a representative heuristic across architectures. While a more optimal combination of SAEs likely exists for each specific architecture, our core finding that SAE constraints substantially improve interpretability with limited performance cost is robust across the configurations we tested.

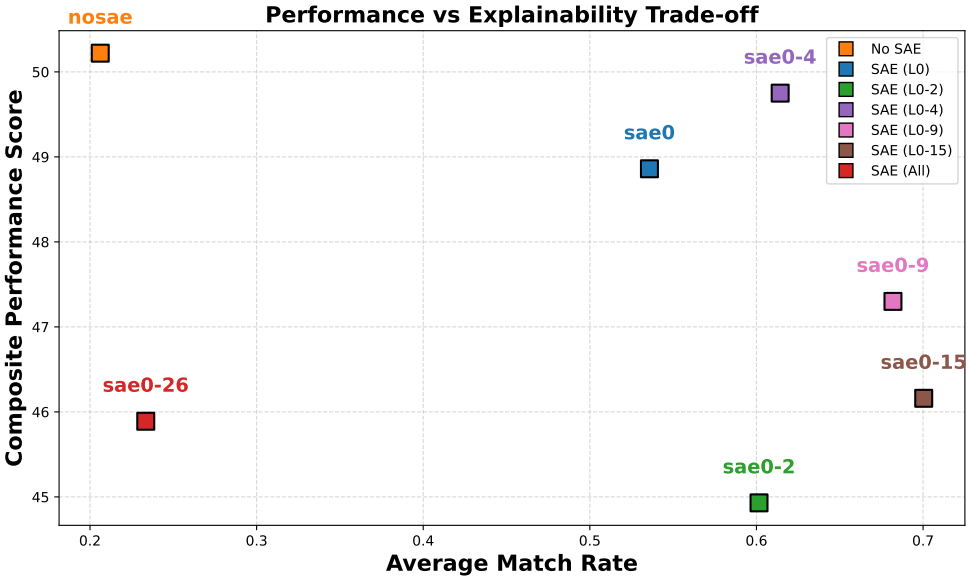


Figure 22. Performance-Interpretability Trade-off. Increasing the number of SAE-constrained layers improves interpretability up to a threshold. The closest performance to not including any SAE constraints is achieved including the SAEs from layer 0 to 4.

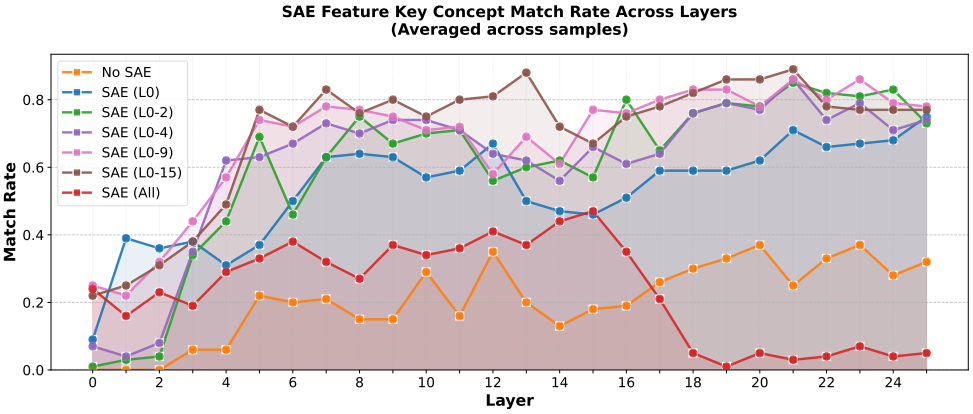


Figure 23. Matching Rate across layers for different SAE constraints. Increasing the number of SAE-constrained layers improves interpretability up to a threshold. Including all layers for SAE constraints reduces the matching rate in the last layers.

I. Spatial Localization with Different LLM Backbones

We compute the spatial localization accuracy scores across different visual encoders and LLM backbones (Table 6). DINOv2 consistently demonstrates high spatial alignment. While I-JEPA also maintains consistent alignment, its accuracy remains lower than that of DINOv2. Conversely, CLIP exhibits the lowest localization accuracy across all LLMs, with scores that fluctuate depending on the specific language backbone.

Table 6. Spatial Localization Accuracy Scores across different visual encoders and LLM backbones.

Visual Encoder	LLM Backbone	SLA
DINOv2	Gemma-2-2B	0.927
	Gemma-2-9B	0.918
	LLaMA-3.1-8B	0.913
CLIP	Gemma-2-2B	0.201
	Gemma-2-9B	0.642
	LLaMA-3.1-8B	0.261
I-JEPA	Gemma-2-2B	0.702
	Gemma-2-9B	0.717
	LLaMA-3.1-8B	0.770

J. Steering with Different LLM Backbones

We evaluate the robustness of the VISTA steering mechanism across three distinct LLMs for all visual encoders. As shown in Table 7, steering successfully works across different LLM backbones for DINOv2 model with similar Mean Score result. For the other two visual encoders I-JEPA generally performs better than CLIP for all LLMs as it maintains a higher localization accuracy.

Table 7. Quantitative ablation analysis of steering with and without SAEs across DINOv2, CLIP, and I-JEPA visual encoders using Gemma-2-2B, Gemma-2-9B, and LLaMA-3.1-8B backbones.

Method	LLM Backbone	Remove Operation				Replace Operation			
		S2	S1	S0	MS	S2	S1	S0	MS
Baseline (No SAE)									
DINOv2	Gemma-2-2B	10	2	88	0.22	3	0	97	0.06
	Gemma-2-9B	11	0	89	0.22	2	0	98	0.04
	LLaMA-3.1-8B	0	0	100	0.00	1	0	99	0.02
CLIP	Gemma-2-2B	13	20	67	0.46	16	6	78	0.38
	Gemma-2-9B	5	2	93	0.12	1	1	98	0.03
	LLaMA-3.1-8B	0	0	100	0.00	1	0	99	0.02
I-JEPA	Gemma-2-2B	8	18	74	0.34	20	9	71	0.49
	Gemma-2-9B	8	11	81	0.27	6	1	93	0.13
	LLaMA-3.1-8B	4	2	94	0.10	2	0	98	0.04
VISTA (With SAE)									
DINOv2	Gemma-2-2B	47	49	4	1.43	59	28	13	1.46
	Gemma-2-9B	54	38	8	**1.46**	62	25	13	1.49
	LLaMA-3.1-8B	47	35	18	1.29	63	30	7	**1.56**
CLIP	Gemma-2-2B	18	4	78	0.40	7	4	89	0.18
	Gemma-2-9B	13	10	77	0.36	24	14	62	0.62
	LLaMA-3.1-8B	4	0	96	0.08	6	6	88	0.18
I-JEPA	Gemma-2-2B	8	25	67	0.41	17	47	36	0.81
	Gemma-2-9B	12	19	69	0.43	28	35	37	0.91
	LLaMA-3.1-8B	26	21	53	0.73	36	36	28	1.08

K. Out-of-Distribution Robustness

Our training data (CC3M and LLaVA instruction tuning data) is predominantly photo-centric. To test whether VISTA's alignment generalizes beyond this distribution, we evaluated on 500 images from DomainNet (Peng et al., 2019), spanning visually distinct domains such as clipart, sketches, paintings, and infographics. These domains are OOD from the projector's training data.

We report reconstruction error, sparsity, and matching rate on this OOD set, comparing against in-distribution results. As shown in Tables 8 and 9, all three metrics remain consistent across distributions, indicating that VISTA's cross-modal alignment is a structural property that transfers beyond the training distribution.

Table 8. Reconstruction error and sparsity on OOD DomainNet images compared to text tokens. Metrics remain comparable, indicating visual tokens continue to inhabit the text SAE manifold under distribution shift.

Modality	**Reconstruction**	**Sparsity**
Text	6.92	0.0048
Image (OOD)	6.49	0.0042

Table 9. Matching rate on in-distribution versus OOD (DomainNet) images. VISTA preserves most of its interpretability under significant distribution shift.

Image Data	**Match Rate**
In-Distribution	0.63
OOD (DomainNet)	0.58

We attribute this robustness to two factors. First, VISTA learns a structural mapping between two frozen representation spaces rather than memorizing training examples, so it inherits the base vision encoder's zero-shot generalization. Second, because DINOv2 already clusters semantically related images (e.g., a sketch of a dog and a photo of a dog) in nearby regions of its feature space, the linear projector preserves this clustering when mapping into the text SAE space. Consequently, concepts learned on photographic training data transfer naturally to non-photographic domains.

L. Visual vs Text-Only Steering Comparison

To verify that VISTA's interventions modify the model's perception of visual content rather than hijacking the language generation pathway, we compare three intervention regimes on the "Cat on a couch" image from Figure 5, using the "-cat +dog" direction in all cases:

- **VISTA (visual tokens steered):** The model outputs *"A dog sitting on a couch with a pillow"*. Scene context is preserved because unsteered background patches remain intact.
- **Text-only steering, no image:** Without the image, the model outputs generic text such as *"A dog running in a park"* failing to describe the actual scene (couch, pillow).
- **Prompt steering with image present:** Applying the "-cat +dog" vector to the text tokens while the image is provided causes hallucinations such as *"The dog is playing in the grass with a cat next to it"*. The original concept is not removed and hallucinated content appears.

Only VISTA produces a coherent, scene-preserving modification of the perceived content. This three-way comparison provides direct causal evidence that VISTA modifies the model's perception of specific visual patches rather than steering the language pathway.