SAE-V: Interpreting Multimodal Models for Enhanced Alignment

Hantao Lou[*1] **Changye Li**[*1] **Jiaming Ji**[1] **Yaodong Yang**[1]

Abstract

With the integration of image modality, the semantic space of multimodal large language models (MLLMs) is more complex than text-only models, making their interpretability more challenging and their alignment less stable, particularly susceptible to low-quality data, which can lead to inconsistencies between modalities, hallucinations, and biased outputs. As a result, developing interpretability methods for MLLMs is crucial for improving alignment quality and efficiency. In text-only LLMs, Sparse Autoencoders (*SAE*s) have gained attention for their ability to interpret latent representations. However, extending *SAE*s to multimodal settings presents new challenges due to modality fusion and the difficulty of isolating cross-modal representations. To address these challenges, we introduce *SAE-V*, a mechanistic interpretability framework that extends the *SAE* paradigm to MLLMs. By identifying and analyzing interpretable features along with their corresponding data, *SAE-V* enables fine-grained interpretation of both model behavior and data quality, facilitating a deeper understanding of cross-modal interactions and alignment dynamics. Moreover, by utilizing cross-modal feature weighting, *SAE-V* provides an intrinsic data filtering mechanism to enhance model alignment without requiring additional models. Specifically, when applied to the alignment process of MLLMs, *SAE-V*-based data filtering methods could achieve more than 110% performance with less than 50% data. Our results highlight *SAE-V*'s ability to enhance interpretability and alignment in MLLMs, providing insights into their internal mechanisms.

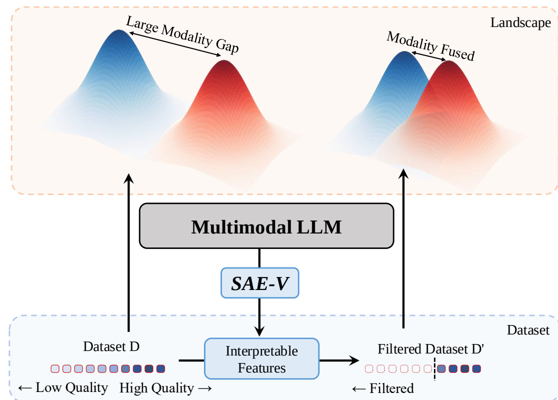


Figure 1. **Operational Dynamics of *SAE-V* Based Data Filtering Method**. *SAE-V* encodes and interprets the representation inside MLLM during alignment and inference time. Based on this representation, we could reveal the modality gap within the data, and improve the alignment process through the selection of modality-fused, high-quality data. This pipeline performs data filtering without requiring additional models, relying instead on MLLM itself to prioritize high-value data effectively.

1. Introduction

With the development and success of large language models (LLMs) (Dubey et al., 2024; Achiam et al., 2023), researchers have begun to introduce visual understanding to these models, thereby extending their operational scope from language to a mix of vision and language, resulting in the creation of powerful multimodal large language models (MLLMs) (Alayrac et al., 2022; Liu et al., 2024; Team et al., 2024; Team, 2024). To enhance the multimodal understanding capabilities of MLLMs, the research community has explored various architectures, including using individual image/text encoders to encode cross-modal information into a joint representation space (Zhang et al., 2023; Liu et al., 2024; Zhu et al., 2024; Wu et al., 2024b) and leveraging image tokenizers to transform all inputs into a unified token sequence (Team, 2024; Xie et al., 2024; Wu et al., 2024a; Wang et al., 2024). Despite the difference in the architectures of these models, their essential goal is the same: Fuse the text and image representation space into a joint multimodal semantic space.

As MLLMs continue to scale up in both size and capabil-

[*]Equal contribution [1]Institute for AI, Peking University. Correspondence to: Yaodong Yang <yaodong.yang@pku.edu.cn>.

Preliminary work. Do not widely distribute.

ity, their interpretability and controllability remain a significant challenge (Zhang & Zhu, 2018; Stan et al., 2024). Currently, mechanistic interpretability techniques such as circuit analysis (Olsson et al., 2022) and dictionary learning with sparse autoencoders (Cunningham et al., 2023) are the most widely recognized approaches to interpreting LLMs. However, their application to MLLMs, especially in the context of cross-modal integration, has been limited. There is a pressing need for specialized tools and frameworks that can unravel the intricate workings of MLLMs.

Moreover, current interpretability efforts are focused mainly on interpreting models, rather than applying this interpretability to real alignment situations, which also makes it difficult to evaluate these methods effectively. Top-down approaches, such as Representation Engineering (Zou et al., 2023) and activation steering (Turner et al., 2023; Panickssery et al., 2023), can directly evaluate the control effects of interpretability methods through control or unlearning techniques. However, for bottom-up methods like circuit analysis (Wang et al., 2022), sparse autoencoders (*SAE*s) (Cunningham et al., 2023), and cross-coders, effective evaluation methods beyond loss reduction are limited. Based on the previous discussion, *can we propose a bottom-up multimodal interpretability approach that can directly enhance the alignment process?*

In this work, we developed *SAE-V*, a mechanistic interpretability framework for MLLMs that extends the *SAE* paradigm to MLLMs. These tools are then applied to interpret the training process of transitioning from LLMs to MLLMs, as well as the process of enhancing the multimodal capabilities of MLLMs. Furthermore, utilizing the interpretable features of *SAE-V* models and their relationship to MLLM capabilities, we designed a data filtering metric based on *SAE-V*. This metric can filter out data that hinder the development of multimodal understanding, achieving stronger alignment with a smaller dataset. Overall, our work makes the following contributions:

- **Multimodal interpretability tool** We developed mechanistic interpretability tools for MLLMs based on previous attempts on LLMs and trained corresponding *SAE-V* models. We demonstrated that *SAE-V* models trained on MLLMs can effectively extract interpretable features, and *SAE-V* models can be transferred to the corresponding LLMs. Specifically, the reconstruction loss of our *SAE-V* models trained on MLLMs is 38.3% and 50.6% compared to the *SAE* model when applied to MLLMs and LLMs, respectively.

- **Interpreting Multimodal Alignment Process** We utilized *SAE-V* to study the feature distribution throughout the alignment process. We discovered that the feature distribution of *SAE-V* corresponds to the MLLM's performance on multimodal understanding tasks.

- **Filtering metric to improve alignment** Based on the previous investigation with *SAE-V*, we developed a metric to filter multimodal datasets and acquire high-quality data, therefore improving alignment quality and efficiency. Experiments demonstrate that our filtering tool achieves more than 110% performance compared to the full dataset while using 50% less data, underscoring the efficiency and effectiveness of *SAE-V*.

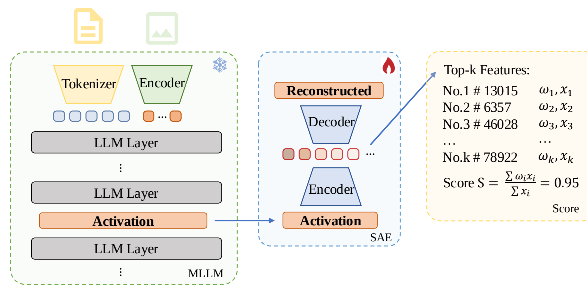


Figure 2. **The interpretability and data filtering pipeline of *SAE-V*.** *SAE-V* is trained to encode MLLM activations into sparse, interpretable features. We first acquire the cross-modal weight of these features via *SAE-V* models, then reversely score the given data by the weighted average of each feature's score. In this way, we provide an intrinsic data filtering tool by eliciting MLLM's latent representation of these data.

2. Methodology

In this section, we present our method to train, evaluate, and apply *SAE-V* to interpret MLLMs and multimodal data.

2.1. Preliminary: Sparse Autoencoder Paradigm

We adopt *SAE-V* (denoted as $\mathcal{S}_\theta$) architecture from the methodology proposed in (Bricken et al., 2023), which comprises an encoder and a feature dictionary $\mathcal{F}_\theta$: $\{\boldsymbol{f}_k\}_{k=1}^{\boldsymbol{n}}$ as a decoder. Let the input be denoted as $H \in \mathbb{R}^{\boldsymbol{l}\times\boldsymbol{m}}$, the hidden state of a specific layer of a MLLM $\mathcal{M}_\theta$. The *SAE-V* encoding operation $\mathcal{S}_\theta(\cdot)$ is

$$Z = \mathrm{ReLU}(W_{\text{enc}} H + b_{enc}), \tag{1}$$

where $Z \in \mathbb{R}^{\boldsymbol{l}\times\boldsymbol{n}}$ is the feature activation of the input. The reconstruction loss of $\mathcal{S}_\theta$ donates as

$$\mathcal{L}_R = ||H - Z \times (\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_m)^\top||_2^2. \tag{2}$$

The training loss is defined by

$$\mathcal{L} = \mathcal{L}_R + \lambda \mathcal{L}_1, \tag{3}$$

where $\mathcal{L}_1 = ||Z||_1$ adds a sparsity constraint to the learned features and λ is a hyperparameter controlling the level

Algorithm 1 Cosine similarity score Ranking

Require: Text token vocabulary: $\mathcal{T}$; vision token vocabulary: $\mathcal{V}$; multimodal dataset $\mathcal{D} = \{\boldsymbol{d}_i\}_{i=1}^{p}$; MLLM parameterized by $\mathcal{M}_\theta$; *SAE-V* model $\mathcal{S}_\theta$; features of *SAE-V* model $\mathcal{F}_\theta$: $\{\boldsymbol{f}_k\}_{k=1}^{n}$; activation bound: δ; cosine similarity function C (Equation 7)
Ensure: Ranked data $\mathcal{D}_R$
Stage 1: Collect Feature Activation Token
Initialize activated token set of features $\mathcal{A}_k \leftarrow \varnothing$
$\mathcal{D}_s \leftarrow \mathit{Sample}(\mathcal{D})$
for each $\boldsymbol{d}_i \in \mathcal{D}_s$ **do**
  $H_i \leftarrow \mathcal{M}_\theta(\boldsymbol{d}_i)$
  $Z_i \leftarrow \mathcal{S}_\theta(H_i)$
  for each $\boldsymbol{f}_k \in \mathcal{F}_\theta$ **do**
    $\mathcal{A}_k \leftarrow \mathcal{A}_k \cup \{\boldsymbol{h}_j\colon \boldsymbol{h}_j \in H_i, \boldsymbol{z}_j = \boldsymbol{e}_j Z_i, \boldsymbol{z}_j \boldsymbol{f}_k^\top > \delta\}$
  end for
end for
Stage 2: Compute Cross-modal Weight
Initialize cross-modal weight of features $\omega_k \leftarrow 0$
for each $\boldsymbol{f}_k \in \mathcal{F}_\theta$ **do**
  $\omega_k \leftarrow C(\mathit{TopK}(\mathcal{A}_k \cap \mathcal{T}), \mathit{TopK}(\mathcal{A}_k \cap \mathcal{V}))$
end for
Stage 3: Rank Dataset by Cross-modal Weight
Initialize cross-modal score of data $\boldsymbol{s}_i \leftarrow 0$
for each $\boldsymbol{d}_i \in \mathcal{D}$ **do**
  $Z_i \leftarrow \mathcal{S}_\theta(\mathcal{M}_\theta(\boldsymbol{d}_i))$
  $F_i \leftarrow \{\boldsymbol{f}_k\colon \|\boldsymbol{f}_k Z_i^\top\|_\infty > \delta\}$
  $\boldsymbol{s}_i \leftarrow \sum_{\boldsymbol{f}_k \in F_i} \omega_k$
end for
$\mathcal{D}_R \leftarrow \mathit{Sort}(\mathcal{D}, \{\boldsymbol{s}_i\}_{i=1}^{n})$

of sparsity. The training results could also quantized by incorporating an additional sparsity constraint via $1\mathcal{L}_0 = ||Z||_0$, which counts the number of nonzero elements in the learned features Z.

2.2. Interpreting Multimodal Data with *SAE-V*

It has been previously demonstrated (Gao et al., 2024; Cunningham et al., 2023) that *SAE* can be employed to interpret how LLMs encode semantic information from these models. This feature motivates us to apply *SAE-V* to assess the quality of the data and thus facilitate data filter for alignment.

We adopt a cosine similarity score ranking algorithm for data filtering (shown in Algorithm 1). Let the multimodal training dataset be donated as $\mathcal{D} = \{\boldsymbol{d}_i\}_{i=1}^{p}$, where $\boldsymbol{d}_i = \{\boldsymbol{u}_j\colon \boldsymbol{u}_j \in \mathcal{T} \vee \boldsymbol{u}_j \in \mathcal{V}\}_{j=1}^{m}$ is a set of tokens from text vocabulary $\mathcal{T}$ and tokens from vision vocabulary $\mathcal{V}$. We acquire feature activation token $\boldsymbol{z}_j$ by MLLM forward and Equation 1, *i.e.*

$$\begin{aligned} H_i &= \mathcal{M}_\theta(\boldsymbol{d_i}) && (4)\\ Z_i &= \mathcal{S}_\theta(H_i), && (5)\\ \boldsymbol{z}_j &= \boldsymbol{e}_j Z_i, && (6) \end{aligned}$$
$$\text{where } \boldsymbol{e}_j = (0, 0, \ldots, \underbrace{1}_{j\text{-th position}}, \ldots, 0).$$

We define a *SAE-V* feature $\boldsymbol{f}_k$ is activated on $\boldsymbol{z}_j$ *if* $\boldsymbol{z}_j \boldsymbol{f}_k^\top > \delta$, where δ is activation bound. Correspondingly, we state that $\boldsymbol{f}_k$ is activated on $\boldsymbol{d}_i$ *if* $\exists\, \boldsymbol{z}_j \in Z_i$ activating $\boldsymbol{f}_k$.

Our algorithm 1 consists of three stages: (1) *Collecting feature activation tokens from dataset*, (2) *Computing cross-modal weight of SAE-V features*, and (3) *Ranking dataset by cross-modal weight*.

1. **Feature Activation Token Collecting** We first sample a small subset $\mathcal{D}_S$ of the training dataset $\mathcal{D}$ and input these samples into the MLLM to obtain hidden states H. These hidden states are then fed into the *SAE-V* encoder to extract feature activations defined as Equation 4. For each feature, we collect its hidden state tokens thereby obtaining a sample of feature activation tokens across the dataset.

2. ***SAE-V* Feature Weighting** For each feature $\boldsymbol{f}_k$, we identify its top-K hidden state text tokens $\boldsymbol{t} = \mathit{TopK}(\mathcal{A}_k \cap \mathcal{T})$ and top-K hidden state vision tokens $\boldsymbol{v} = \mathit{TopK}(\mathcal{A}_k \cap \mathcal{V})$, where the top-K is ranked by the activation value $\boldsymbol{z}_j \boldsymbol{f}_k^\top$ of the token. We then compute the cosine similarity between the two lists of tokens, donating the cross-modal weight of feature $\boldsymbol{f}_k$ as

$$Cosine(\boldsymbol{t}, \boldsymbol{v}) = \frac{1}{k} \sum_{i=1}^{k} \frac{\boldsymbol{t}_i \cdot \boldsymbol{v}_i}{||\boldsymbol{t}_i||||\boldsymbol{v}_i||}, \qquad (7)$$

which represents the capability of the feature to capture multimodal information within data.

3. **Data Ranking** Using the weighted features of *SAE-V* model, we score the entire training dataset. The cosine similarity score of each piece of data is defined as the sum of the cosine similarity scores of its activating features. We rank the data set by the score and the resulting cosine similarity score order allows us to filter data that are better aligned with the structures of multimodal semantic information.

We present our experiments and results in Section 4, demonstrating that our cosine similarity score ranking method can effectively filter high-quality data from the training data set.

3. Interpretability Analysis with *SAE-V*

In this section, we conduct experiments on the *SAE-V* paradigm, aiming to demonstrate the capability and transferability of *SAE-V* model. We also performed experiment to prove the effectiveness of our *SAE-V*-based data interpreting tool from the inference side.

3.1. Training and Evaluating *SAE-V* Model

We trained a series of *SAE* and *SAE-V* models on MLLMs and their base LLMs. We evaluated the performance of these models, and the results demonstrated that *SAE-V* model is capable of interpreting MLLMs and that *SAE-V* model trained on MLLM can be effectively transferred to its original LLM.

3.1.1. Experiment Setup

Datasets For text-only and multimodal situations, we selected the Pile (Gao et al., 2020) and Obelics (Laurençon et al., 2023) datasets separately. Specifically, we sampled 100K data from each dataset as the train set and 10K data as the test set. The Pile is a diverse language modeling dataset for LLM pretraining, and Obelics is a massive interleaved image-text dataset for MLLM pretraining. These two datasets are widely recognized in various pretraining and interpretability works (Black et al., 2022; Biderman et al., 2023; Cunningham et al., 2023; Team, 2024).

Models We selected two generic MLLMs, LLaVA-NeXT-7B (Liu et al., 2024) and Chameleon-7B (Team, 2024), as our target models. These two models represent two distinct architectures, and testing our method on them can prove that our method is applicable to different architectures.

Additionally, we also studied Anole-7B (Chern et al., 2024) and Mistral-7B (Jiang et al., 2023) to compare the behavior of *SAE* and *SAE-V* models before and after fine-tuning, specifically the transitioning fine-tuning from LLM to MLLM. Anole-7B is a variant of Chameleon-7B, with its image generation capability unlocked, while Mistral-7B is the base LLM of LLaVA-NeXT-7B. [1]

Evaluation Metrics To evaluate the performance of *SAE-V* models, we use two key metrics: $\mathcal{L}_0 = ||\boldsymbol{z}||_0$ where $\boldsymbol{z}$ is defined in Equation 6 and reconstruction loss $\mathcal{L}_R$ in Equation 2. $\mathcal{L}_0$ quantifies the number of activated features, reflecting the method's ability to extract interpretable features, while reconstruction loss measures the method's activation reconstruction capability compared with the model output, indicating the method's accuracy in giving interpretations.

[1]We present the detailed training setup and hyper-parameters in *Appendix A.1*.

3.1.2. Experiment Result

Model	Method	$\mathcal{L}_0$
LLaVA-NeXT-7B	*SAE*	94.5
	SAE-V	192.5
Chameleon-7B	*SAE*	24757.6
	SAE-V	50.1
Anole-7B	*SAE*	62.1
	SAE-V	50.1

Table 1. **The $\mathcal{L}_0$ metric of *SAE* and *SAE-V* models**. $\mathcal{L}_0$ indicates the sparsity cost (average activated feature number). The results vary significantly across models due to their architectural differences. For Anole-7B and Chameleon-7B, *SAE-V* models maintain lower $\mathcal{L}_0$, suggesting more efficient feature utilization. However, LLaVA-NeXT-7B shows a contrary pattern with *SAE-V* model requiring higher feature activation than *SAE*. We propose that extra activated features of *SAE-V* model are introduced by extra vision tokens in multimodal data. Notably, Chameleon-7B with *SAE* model exhibits an unusually high sparsity cost, attributed to multiple unseen vision tokens in the inference stage.

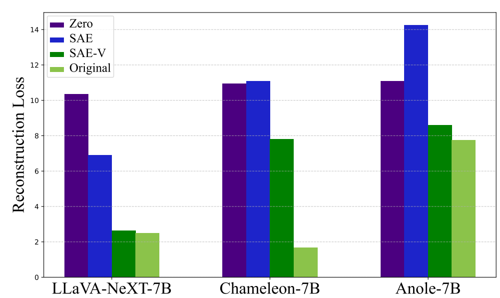


Figure 3. **Reconstruction capability of *SAE* and *SAE-V* models.** Each section compares the metrics of zero (set all activations as zero), *SAE* model, *SAE-V* model, and the Original reference state. *SAE-V* model consistently demonstrates superior reconstruction performance across all tested models. In Chameleon-7B and Anole-7B, *SAE* performs worse than the zero baseline, which indicates that *SAE* trained in text data fails to capture interpretable features in these MLLMs.

Capability of *SAE-V* Model We compare the performance of *SAE-V* and *SAE* on different multimodal models. The $\mathcal{L}_0$ (shown in Table 1) varies significantly across the three models. For LLaVA-NeXT-7B, the $\mathcal{L}_0$ of *SAE-V* is much higher than that of *SAE*. For Chameleon-7B, *SAE-V* performs normally, whereas the $\mathcal{L}_0$ of *SAE* is abnormally high, indicating that *SAE* fails to extract sparse features. We suppose that the failure is attributed to a large number of unseen vision tokens for *SAE* during the inference stage. For Anole-7B, the $\mathcal{L}_0$ of *SAE* and *SAE-V* are nearly identical. The reconstruction loss (shown in Figure 3) of *SAE-V* is lower than *SAE* and is closer to the original activation, demonstrating that *SAE-V* behaves much better at reconstructing original activation than *SAE* across all three models. The results indicate that *SAE-V* outperforms *SAE* in terms of capability.

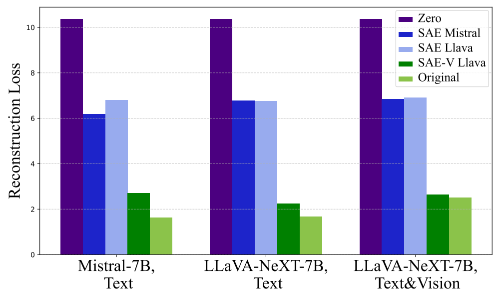


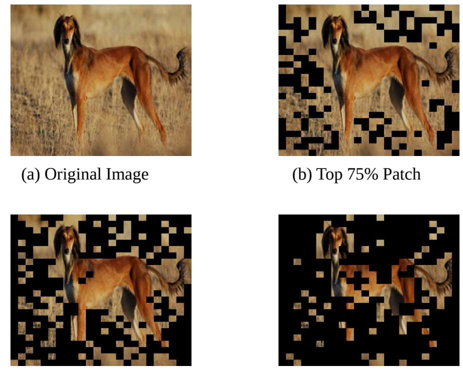
Figure 4. **Reconstruction performance of *SAE* and *SAE-V*.** The x-axis shows different models and task configurations, text indicates text-only task, and text & vision indicates multimodal task. The colored bars represent five experiment groups (zero activation, *SAE* of Mistral-7B, *SAE* of LLaVA-NeXT-7B trained with the Pile, *SAE* of LLaVA-NeXT-7B trained with Obliecs, original performance). Across various settings, *SAE-V* consistently demonstrates superior transferability compared to *SAE* and achieves reconstruction loss close to the original performance, the maximum relative gap being 67.28%. *SAE* based on Mistral-7B and *SAE* based on LLaVA-NeXT-7B achieves nearly the same loss in all tasks and models, indicating the equivalence of training *SAE* with MLLM and its base LLM.

Transferability of *SAE-V* Model We compared the reconstruction performance of *SAE-V* model trained on LLaVA-NeXT-7B and *SAE* model to prove that *SAE-V* model trained on MLLMs can generalize to its base LLM. The findings (shown in Figure 4) indicate that across different settings, *SAE-V* model consistently achieves the best performance. Moreover, when trained on both MLLM and LLM, *SAE* model exhibits nearly identical reconstruction loss values, showing its robust transferability.

These results highlight that training *SAE-V* model for MLLMs with multimodal data is effective for interpreting MLLMs, and even LLMs, as *SAE* model trained solely on textual data fail to extract and disentangle the hidden representations of MLLMs effectively. Moreover, *SAE-V* model demonstrates superior capability in reconstructing the reasoning features of MLLMs compared to the standard *SAE* model.

3.2. Apply *SAE-V* Model on Multimodal Data

In this section, we conduct an image classification task on the ImageNet dataset (Russakovsky et al., 2015) to investigate whether *SAE-V* can capture the key information within images and to validate the effectiveness of the methods proposed in Section 2.2 on multimodal data. We apply 4 methods, namely $\mathcal{L}_0$, $\mathcal{L}_1$, co-occurring $\mathcal{L}_0$, and cosine similarity score, where co-occurring $\mathcal{L}_0$ is defined as the number of features activated on at least one text and image token. The cosine similarity score is defined as the sum of cross-modal



(a) Original Image (b) Top 75% Patch

(c) Top 50% Patch (d) Top 25% Patch

Figure 5. **Case analysis of image patch filtering using $\mathcal{L}_0$ metric.** We rank and filter image patches according to the number of features activated on them. The top row shows the original image (a) and its reduced-patch versions retaining 75% (b), 50% (c), and 25% (d). In this dog image, the patches are filterd out from edge to the middle and preserved almost only dog patches, suggesting that *SAE-V* model is preserving the main semantic information of the image.

weights of features, consistent with Algorithm 1. We adopt these metrics to filter image patches, thus obtaining images that preserve 75%, 50%, and 25% patches, respectively. [2]

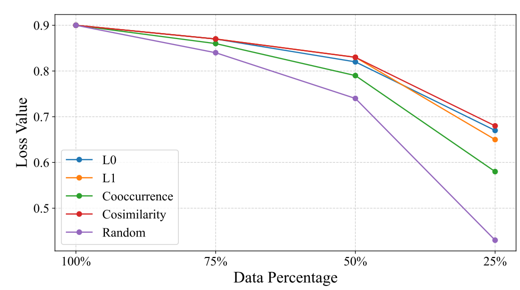


Figure 6. **The classification performance on ImageNet.** We compare the classification accuracy after filtering the image patches with $\mathcal{L}_0$ method, $\mathcal{L}_1$ method, co-occurrence $\mathcal{L}_0$ method, cosine similarity score method, and the random baseline. All methods achieve high accuracy when preserving 75% or 50% patches and $\mathcal{L}_0$ method, $\mathcal{L}_1$ method, and cosine similarity score method maintains high accuracy even in the least patches. The result shows that *SAE-V* is efficient in capturing critical information from images.

Case Analysis Figure 5 illustrates the image patches when using $\mathcal{L}_0$ metric for filtering. Even when employing the simplest $\mathcal{L}_0$ metric, *SAE-V* is still able to effectively capture the critical semantic information of the image. [3]

[2] We present complete algorithms of 4 methods in Appendix C.1.

[3] More cases and analyses are presented in Appendix C.2.

Quantized Results The quantized results are presented in Figure 6, where we observe that, for all methods, preserving 75% or 50% of the patches achieves an accuracy close to that obtained using the full image. In the challenging scenario where only 25% of the patches are retained, $\mathcal{L}_0$ method, $\mathcal{L}_1$ method, and cosine similarity score method maintain accuracy levels close to 70%, and all methods significantly surpass the accuracy obtained by the random preservation method. These results demonstrate that *SAE-V* can accurately capture critical information in images and that the methods proposed in Section 2.2 effectively utilized *SAE-V* features during inference.

4. Alignment Experiment

In this section, we adopt cosine similarity score rank- ing algorithm as a data filter (as shown in Algorithm 1) to acquire high-quality data for model alignment.

4.1. Experiment Setup

Dataset and Model Consistent with Section 3.1.1, we selected LLaVA-NeXT-7B (Liu et al., 2024) and Chameleon-7B (Team, 2024) for our alignment experiment. Since the LLaVA-NeXT-7B model is rather powerful in multimodal capabilities, we selected the Align-Anything (Ji et al., 2024) text-image-to-text dataset for our experiment. Align-Anything is a 400K multimodal preference dataset containing fine-grained annotated multimodal input-output preference data, and we used the 40K subset of text-image input and text output in our experiment.

Algorithm We adopt the cosine similarity score ranking algorithm (shown in Algorithm 1) as a filter to exclude data with low scores. In addition, we also adopt two algorithms, the $\mathcal{L}_0$ ranking, and the co-occurrence ranking. [4]

Evaluation To evaluate the efficiency of our methods, we applied Direct Preference Optimization (DPO) to the model using the filtered datasets. [5] We then evaluate the multimodal capabilities of the model using LLaVA-Bench (Liu et al., 2024) benchmarks.

4.2. Experiment Results

We performed *SAE-V*-based data filter with different filtering ratios on the LLaVA-NeXT-7B model and Align-Anything dataset. The filtered datasets were then used to fine-tune MLLMs, which were evaluated on LLaVA-Bench. The results (shown in Figure 7) demonstrate that our *SAE-V*-based filtering method effectively enhances the alignment of LLaVA-NeXT-7B, even with reduced data. Since most of the data in Align-Anything contribute positively to model alignment, the performance of the model is higher than the base model without any fine-tuning in most cases. At any data filter proportion, the *SAE-V*-based data filtering method outperforms the random selection baseline, with the best result being 108.17 (115% of the full dataset's performance) achieved using 50% filtered data from the cooccurrence filter, and 104.20 (108% of the full dataset's performance) achieved using 20% filtered data from the cosine similarity filter. However, as the dataset inevitably contains some low-quality data, the performance is optimal with a moderate data proportion and shows a downward trend as the data proportion increases.

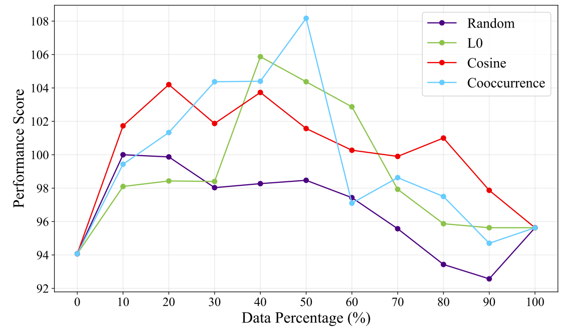


Figure 7. **The performance of *SAE-V*-based data filter method** *SAE-V*-based data filter method significantly outperforms the random selection baseline. Specifically, the cosine similarity filter demonstrated the most stable performance, consistently surpassing the random filter across all data percentages and achieving 108% of the full dataset's performance with only 20% of the data. The co-occurrence filter peaked at 50% of the data, reaching a score of 108.17, (115% of the full dataset's performance). As a more straightforward utilization of *SAE-V* model, the L0 filter also generally outperforms the random selection baseline.

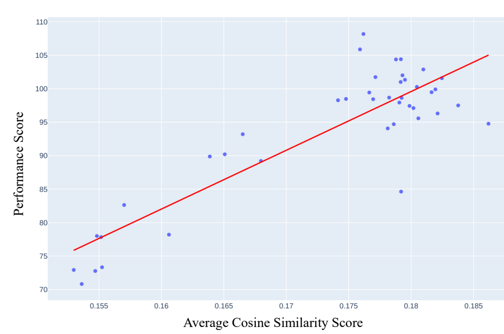


Figure 8. **The relationship between average cosine similarity score and MLLM performance.** We measure the average cosine similarity score of models in Section 4.2, and fit a linear relationship between model performance and average cosine similarity score. The correlation coefficient of the correlation reaches 0.84, suggesting that higher similarity scores on *SAE-V* features correspond to enhanced MLLM performance.

[4]Detailed descriptions of these ablation algorithms and their corresponding hyperparameters are provided in Appendix B.

[5]We present detailed training parameters in Appendix B.2.

4.3. Relationship between MLLM Capability and *SAE-V* Features

In the previous section, we demonstrated the effectiveness of utilizing the cosine similarity score for data filters in model training. To further investigate the relationship between model performance and cross-modal similarity, as measured by cosine similarity of *SAE-V* features, we further measure the average cosine similarity score of these models. Given a dataset, we apply the cosine similarity score ranking algorithm (shown in Algorithm 1) to the MLLM, and we define the MLLM's average cosine similarity score as the mean score of all non-zero cross-modal weight *SAE-V* features.

We calculated the average cosine similarity scores for the models discussed in Section 4.2. The result (shown in Figure 8) revealed a positive correlation between the average cosine similarity score of *SAE-V* feature and the performance of MLLM, suggesting that higher similarity scores of *SAE-V* features correspond to enhanced MLLM performance.

4.4. Ablation Study

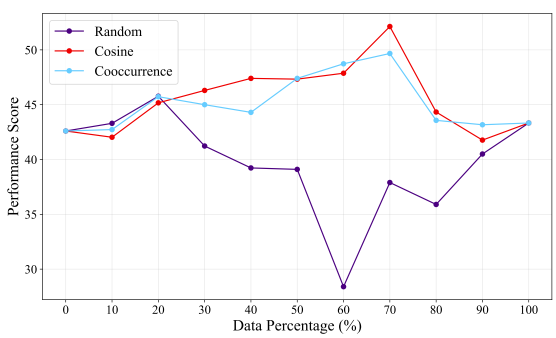


Figure 9. **The performance of *SAE-V*-based data filter method on Chameleon-7B** We replicate *SAE-V*-based data filter on Chameleon-7B model and Align-Anything dataset. The result shows that on Chameleon-7B model, *SAE-V*-based data filter also achieved significant performance gain compared to the random filter. Specifically, both cosine similarity and co-occurrence filters perform better than random filters at almost every data percentage, and the cosine similarity filter achieved a score of 52.13 (120% of the full dataset's performance) with 70% data.

Ablation on Models To prove that *SAE-V*-based data filter method could generalize to distinct model architectures, we replicate *SAE-V*-based data filter on Chameleon-7B model and Align-Anything dataset. The result (shown in Figure 9) demonstrates that although Chameleon-7B performs worse than LLaVA-NeXT-7B on LLaVA-Bench, the *SAE-V*-based filter method still shows its effectiveness. When using a smaller data proportion, the performance is strongly correlated with the data quantity, and thus the differences between methods are minimal. However, with a larger data proportion, the *SAE-V*-based filter method significantly surpasses the random filter, achieving a peak of 52.13 (120% of the full dataset's performance) with 70% of the data. The largest performance gap is observed in the 50-70% data range, while the differences converge again as the data proportion approaches 100%. This proves that *SAE-V*-based data filter is effective on architectures other than CLIP-based MLLM, and shows its potential to generalize across a wide range of models.

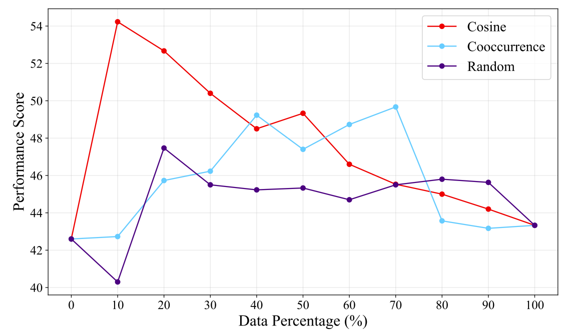


Figure 10. **The performance of *SAE-V*-based data filter method on RLAIF-V dataset.** We performed an ablation study with the RLAIF-V dataset and Chameleon-7B model to prove that *SAE-V*-based data filter method can generalize across datasets. Both cosine similarity and co-occurrence filters generally outperform the random filter on the RLAIF-V dataset. Specifically, the cosine similarity filter achieves the highest score of 54.23 (125% of the full dataset's performance) with only 10% data, demonstrating its supreme efficiency.

Ablation on Datasets We also performed an ablation study on the datasets to be filtered. Since the LLaVA-NeXT-7B model is highly capable, most datasets fail to further enhance its multimodal abilities. Therefore, we selected the RLAIF-V dataset and the relatively weaker Chameleon-7B model for the dataset ablation study. The results (shown in Figure 10) further confirm that *SAE-V*-based data filter is working across different datasets. Moreover, on RLAIF-V, the cosine similarity filter could achieve a score of 54.23 (125% of the full dataset's performance) by using only 10% of the data, demonstrating exceptional efficiency.

Comparation with Other Filtering Methods To validate the effectiveness of our *SAE-V*-based data filtering method, we conducted an ablation study comparing it with other similar data filtering approaches. Since there are currently no widely recognized data filtering methods specifically designed for multimodal data, we adapted the IFD metric (Li et al., 2023b) method to the multimodal setting. The result (shown in Figure 11) suggests that our data filter method achieves a performance comparable to the IFD metric. However, considering that the IFD metric needs to train an additional *cherry model*, our *SAE-V*-based data filter could directly fit various datasets, demonstrating greater generalizability and efficiency.

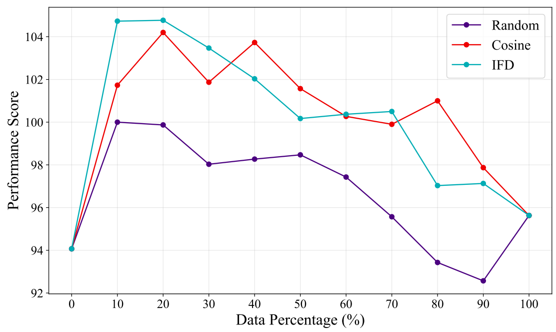


Figure 11. **The performance of *SAE-V*-based data filter and IFD metrics** We replicate the IFD metric on MLLMs and compare the result with our cosine similarity filter. The results show that although the peak performance of *SAE-V*-based data filtering is a little lower compared to IFD metric, our method could achieve a generally similar performance compared to IFD metric without introducing additional models and training process.

5. Related Work

Multimodal Large Language Model MLLM is a type of LLM integrated with multimodal modules that incorporate multimodal information to deal with multimodal tasks. Based on the method of integrating vision features into the model, most MLLMs can be categorized into three types:

- *CLIP-based MLLMs*: These models encode images with CLIP (Radford et al., 2021) and use MLP to project visual features. Examples include LLaVA (Liu et al., 2024) series and NExT-GPT (Wu et al., 2023).
- *Early-Fusion MLLMs*: These models directly tokenize visual features for input. Examples include Chameleon (Team, 2024) and Janus (Wu et al., 2024a) series.
- *Q-Former-based MLLMs*: These models use a structure similar to Q-Former (Li et al., 2023a) to extract visual representations, represented by Qwen-VL (Bai et al., 2023) and MiniGPT-4 (Zhu et al., 2024).

Our study focuses on the CLIP-based and early-fusion MLLMs. Specifically, we select LLaVA-NeXT-7B and Chameleon-7B as the target models.

Mechanistic interpretability with Sparse Autoencoder Mechanistic interpretability seeks to uncover and explain the internal mechanisms that enable models to understand input data and generate responses (Rai et al., 2024). Specifically, most current mechanistic interpretability methods focus on analyzing features, smaller units that contribute to performing explainable semantic tasks, within models (Olah et al., 2020).

Sparse Autoencoder (*SAE*) aims to learn sparse and interpretable features from polysemantic model representations (Yun et al., 2021; Bricken et al., 2023; Sharkey et al., 2022; Peigné, 2023; Elhage et al., 2022). By introducing sparsity constraints, the activation values in the hidden layers of *SAE* are mostly zero, allowing *SAE* to encode polysemantic features in LLM to monosemantic ones.

In this paper, we extended the scope of *SAE* to MLLMs, thereby building *SAE-V*. We further demonstrated *SAE-V*'s capability and transferability on MLLMs, and built a data filter tool based on *SAE-V* to enhance multimodal alignment.

Data Filter in Alignment Data filtering ensures that only relevant high-quality data are used during the alignment of LLM or MLLM s, thus reducing the quantity of data while achieving greater performance (Zhou et al., 2023; Chen et al., 2023; Du et al., 2023; Li et al., 2023c;b; Tu et al., 2024). For example, LIMA (Zhou et al., 2023), ALPAGASUS (Chen et al., 2023), and IFD (Li et al., 2023b) use human annotation, API annotation and train a new model for annotation to score data separately. Our method, *SAE-V*-based data filter, provides a self-guided and interpretable metric to evaluate the similarity of multimodal data, which indicates their qualities. The method is stable and efficient for models of different architectures.

6. Conclusion

This work introduced *SAE-V*, a framework that extends *SAE* to MLLMs and improves their alignment. Through experiments on LLaVA-NeXT-7B and Chameleon-7B, we demonstrated that *SAE-V* model demonstrates excellent capability and transferability in interpreting MLLM and multimodal data, and *SAE-V*-based data filtering methods could achieve more than 110% performance with less than 50% data. These results highlight *SAE-V*'s potential to enhance multimodal model interpretability and alignment efficiently.

Limitation While *SAE-V* introduces significant advancements in interpreting multimodal models and enhancing alignment through mechanistic analysis, several limitations remain unaddressed and warrant further exploration: (1) Although *SAE-V* demonstrates superior interpretability and data filtering efficiency compared to *SAE*, the theory behind *SAE-V*, especially the mathematical relationship between image-text similarity metrics, cross-modal co-occurrence features, and model performance, is not fully revealed. (2) Due to resource constraints, *SAE-V* is primarily evaluated on text and vision modalities, leaving its effectiveness on other modalities such as audio, video, and embodied AI systems unexplored. Our future work will focus on establishing a comprehensive theoretical foundation for *SAE-V* and extending its application to additional modalities, such as audio, video, and embodied AI systems, to broaden its utility and impact.

Impact Statement

The source code and checkpoints of *SAE-V* mentioned in this paper will be released under the CC BY-NC 4.0 license. This research has several potential risks that must be considered. The interpretability tools introduced in this work, while beneficial for alignment, could also be leveraged to manipulate or reverse-engineer model behaviors in unintended ways. Additionally, while *SAE-V* provides a self-guided filtering mechanism, it remains dependent on the initial dataset quality, meaning biases in the dataset could still propagate into the final model. We strongly condemn any malicious use of the *SAE-V* code and checkpoints and advocate for its responsible and ethical use.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

Bai, J., Bai, S., Yang, S., Wang, S., Tan, S., Wang, P., Lin, J., Zhou, C., and Zhou, J. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023.

Biderman, S., Schoelkopf, H., Anthony, Q. G., Bradley, H., O'Brien, K., Hallahan, E., Khan, M. A., Purohit, S., Prashanth, U. S., Raff, E., et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pp. 2397–2430. PMLR, 2023.

Black, S., Biderman, S., Hallahan, E., Anthony, Q., Gao, L., Golding, L., He, H., Leahy, C., McDonell, K., Phang, J., et al. Gpt-neox-20b: An open-source autoregressive language model. *arXiv preprint arXiv:2204.06745*, 2022.

Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N., Anil, C., Denison, C., Askell, A., et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2, 2023.

Chen, L., Li, S., Yan, J., Wang, H., Gunaratna, K., Yadav, V., Tang, Z., Srinivasan, V., Zhou, T., Huang, H., et al. Alpagasus: Training a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701*, 2023.

Chern, E., Su, J., Ma, Y., and Liu, P. Anole: An open, autoregressive, native large multimodal models for interleaved image-text generation. *arXiv preprint arXiv:2407.06135*, 2024.

Cunningham, H., Ewart, A., Riggs, L., Huben, R., and Sharkey, L. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

Du, Q., Zong, C., and Zhang, J. Mods: Model-oriented data selection for instruction tuning. *arXiv preprint arXiv:2311.15653*, 2023.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., Hatfield-Dodds, Z., Lasenby, R., Drain, D., Chen, C., et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., et al. The Pile: An 800GB dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Gao, L., la Tour, T. D., Tillman, H., Goh, G., Troll, R., Radford, A., Sutskever, I., Leike, J., and Wu, J. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*, 2024.

Ji, J., Zhou, J., Lou, H., Chen, B., Hong, D., Wang, X., Chen, W., Wang, K., Pan, R., Li, J., et al. Align anything: Training all-modality models to follow instructions with language feedback. *arXiv preprint arXiv:2412.15838*, 2024.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Laurençon, H., Saulnier, L., Tronchon, L., Bekman, S., Singh, A., Lozhkov, A., Wang, T., Karamcheti, S., Rush, A. M., Kiela, D., Cord, M., and Sanh, V. Obelics: An open web-scale filtered dataset of interleaved image-text documents, 2023.

Li, J., Li, D., Savarese, S., and Hoi, S. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023a.

Li, M., Zhang, Y., Li, Z., Chen, J., Chen, L., Cheng, N., Wang, J., Zhou, T., and Xiao, J. From quantity to quality: Boosting llm performance with self-guided data selection for instruction tuning. *arXiv preprint arXiv:2308.12032*, 2023b.

Li, Y., Hui, B., Xia, X., Yang, J., Yang, M., Zhang, L., Si, S., Liu, J., Liu, T., Huang, F., et al. One shot learning as instruction data prospector for large language models. *arXiv preprint arXiv:2312.10302*, 2023c.

Liu, H., Li, C., Wu, Q., and Lee, Y. J. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

Olah, C., Cammarata, N., Schubert, L., Goh, G., Petrov, M., and Carter, S. Zoom in: An introduction to circuits. *Distill*, 5(3):e00024–001, 2020.

Olsson, C., Elhage, N., Nanda, N., Joseph, N., DasSarma, N., Henighan, T., Mann, B., Askell, A., Bai, Y., Chen, A., et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

Panickssery, N., Gabrieli, N., Schulz, J., Tong, M., Hubinger, E., and Turner, A. M. Steering llama 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06681*, 2023.

Peigné, P. Taking features out of superposition with sparse autoencoders more quickly with informed initialization, Sep 2023. URL https://www.lesswrong.com/posts/YJpMgi7HJuHwXTkjk/taking-features-out-of-superposition-with-sparse.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Rai, D., Zhou, Y., Feng, S., Saparov, A., and Yao, Z. A practical review of mechanistic interpretability for transformer-based language models. *arXiv preprint arXiv:2407.02646*, 2024.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Sharkey, L., Braun, D., and Millidge, B. Taking features out of superposition with sparse autoencoders, Dec 2022. URL https://www.alignmentforum.org/posts/z6QQJbtpkEAX3Aojj/interim-research-report-taking-features-out-of-superposition. [Interim research report].

Stan, G. B. M., Rohekar, R. Y., Gurwicz, Y., Olson, M. L., Bhiwandiwalla, A., Aflalo, E., Wu, C., Duan, N., Tseng, S.-Y., and Lal, V. Lvlm-intrepret: An interpretability tool for large vision-language models. *arXiv preprint arXiv:2404.03118*, 2024.

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

Team, G., Reid, M., Savinov, N., Teplyashin, D., Dmitry, L., Lillicrap, T., Alayrac, J., Soricut, R., Lazaridou, A., Firat, O., et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. in arxiv [cs. cl]. arxiv, 2024.

Tu, Z., Meng, X., He, Y., Yao, Z., Qi, T., Liu, J., and Li, M. Resofilter: Rine-grained synthetic data filtering for large language models through data-parameter resonance analysis. *arXiv preprint arXiv:2412.14809*, 2024.

Turner, A. M., Thiergart, L., Leech, G., Udell, D., Vazquez, J. J., Mini, U., and MacDiarmid, M. Activation addition: Steering language models without optimization. *arXiv e-prints*, pp. arXiv–2308, 2023.

Wang, K., Variengien, A., Conmy, A., Shlegeris, B., and Steinhardt, J. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small, 2022. URL https://arxiv.org/abs/2211.00593.

Wang, X., Zhang, X., Luo, Z., Sun, Q., Cui, Y., Wang, J., Zhang, F., Wang, Y., Li, Z., Yu, Q., et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024.

Wu, C., Chen, X., Wu, Z., Ma, Y., Liu, X., Pan, Z., Liu, W., Xie, Z., Yu, X., Ruan, C., et al. Janus: Decoupling visual encoding for unified multimodal understanding and generation. *arXiv preprint arXiv:2410.13848*, 2024a.

Wu, S., Fei, H., Qu, L., Ji, W., and Chua, T.-S. Next-gpt: Any-to-any multimodal llm. *arXiv preprint arXiv:2309.05519*, 2023.

Wu, S., Fei, H., Qu, L., Ji, W., and Chua, T.-S. NExt-GPT: Any-to-any multimodal LLM. In *Forty-first International Conference on Machine Learning*, 2024b. URL https://openreview.net/forum?id=NZQkumsNlf.

Xie, J., Mao, W., Bai, Z., Zhang, D. J., Wang, W., Lin, K. Q., Gu, Y., Chen, Z., Yang, Z., and Shou, M. Z. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024.

Yun, Z., Chen, Y., Olshausen, B. A., and LeCun, Y. Transformer visualization via dictionary learning: contextualized embedding as a linear superposition of transformer factors. *arXiv preprint arXiv:2103.15949*, 2021.

Zhang, H., Li, X., and Bing, L. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023.

Zhang, Q.-s. and Zhu, S.-C. Visual interpretability for deep learning: a survey. *Frontiers of Information Technology & Electronic Engineering*, 19(1):27–39, 2018.

Zhou, C., Liu, P., Xu, P., Iyer, S., Sun, J., Mao, Y., Ma, X., Efrat, A., Yu, P., Yu, L., et al. Lima: Less is more for alignment. corr abs/2305.11206 (2023). *arXiv preprint arXiv:2305.11206*, 10, 2023.

Zhu, D., Chen, J., Shen, X., Li, X., and Elhoseiny, M. MiniGPT-4: Enhancing vision-language understanding with advanced large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=1tZbq88f27.

Zou, A., Phan, L., Chen, S., Campbell, J., Guo, P., Ren, R., Pan, A., Yin, X., Mazeika, M., Dombrowski, A.-K., et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023.

A. Details of Interpretability Experiment

A.1. Hyperparameter of *SAE* and *SAE-V* Models Training

Hyper-parameters	*SAE* and *SAE-V* of LLaVA-NeXT/Mistral	*SAE* and *SAE-V* of Chameleon/Anole
Training Parameters		
total training steps	30000	30000
batch size	4096	4096
LR	5e-5	5e-5
LR warmup steps	1500	1500
LR decay steps	6000	6000
adam beta1	0.9	0.9
adam beta2	0.999	0.999
LR scheduler name	constant	constant
LR coefficient	5	5
seed	42	42
dtype	float32	float32
buffer batches num	32	64
store batch size prompts	4	16
feature sampling window	1000	1000
dead feature window	1000	1000
dead feature threshold	1e-4	1e-4
***SAE* and *SAE-V* Parameters**		
hook layer	16	8
input dimension	4096	4096
expansion factor	16	32
feature number	65536	131072
context size	4096	2048

Table 2. **Hyperparameters of training *SAE* and *SAE-V* models.**

The hyperparameters of the training *SAE* and *SAE-V* are shown in Table A.1. The differences in training parameters arise because the LLaVA-NeXT-7B model requires more GPU memory to handle vision input, so fewer batches can be cached. For the *SAE* and *SAE-V* parameters, we set different hook layers and context sizes based on the distinct architectures of the two models. We also experimented with different feature numbers on both models, but found that only around 30,000 features are actually activated during training. All training runs were conducted until convergence. All *SAE* and *SAE-V* training is performed on 8×A800 GPUs. We ensured that the variations in the parameters did not affect the experiment results.

B. Details of alignment experiment

We present details of alignment experiment in this section, including algorithms and hyperparameters of algorithms and model training.

B.1. Algorithms in alignment experiment

The complete algorithm of $\mathcal{L}_0$-based Ranking and co-occurring $\mathcal{L}_0$-based Ranking are shown in Algorithm 2 and Algorithm 3. These two algorithms serve as ablation variants of the cosine similarity score Ranking (shown in Algorithm 1). The $\mathcal{L}_0$-based Ranking represents a straightforward algorithm that selects data by directly computing the sum of $\mathcal{L}_0$ for each data point. The co-occurring $\mathcal{L}_0$-based Ranking takes an initial step toward cross-modal consideration by only counting features that are activated across both modalities. Building upon these algorithms, we further developed the cosine similarity score Ranking approach.

Algorithm 2 $\mathcal{L}_0$-based Ranking

Require: multimodal dataset $\mathcal{D} = \{\boldsymbol{d}_i\}_{i=1}^p$; MLLM $\mathcal{M}_\theta$; *SAE-V* $\mathcal{S}_\theta$; features of *SAE-V*; activation bound: δ; $\mathcal{F}_\theta$: $\{\boldsymbol{f}_k\}_{k=1}^n$;
Ensure: Ranked data $\mathcal{D}_R$
for each $\boldsymbol{d}_i \in \mathcal{D}$ **do**
  $Z_i \leftarrow \mathcal{S}_\theta(\mathcal{M}_\theta(\boldsymbol{d}_i))$
  $F_i \leftarrow \{\boldsymbol{f}_k : \|\boldsymbol{f}_k Z_i^\top\|_\infty > \delta\}$
  $\mathcal{L}_{0,i} \leftarrow |F_i|$
end for
$\mathcal{D}_R \leftarrow Sort(\mathcal{D}, \{\mathcal{L}_{0,i}\}_{i=1}^n)$

Algorithm 3 Co-ocurring $\mathcal{L}_0$-based Ranking

Require: Text token vocabulary: $\mathcal{T}$; vision token vocabulary: $\mathcal{V}$; multimodal dataset $\mathcal{D} = \{\boldsymbol{d}_i\}_{i=1}^p$; MLLM $\mathcal{M}_\theta$; *SAE-V* $\mathcal{S}_\theta$; features of *SAE-V* $\mathcal{F}_\theta$: $\{\boldsymbol{f}_k\}_{k=1}^n$; activation bound: δ;
Ensure: Ranked data $\mathcal{D}_R$
Initialize coocurrence feature set of data $F_i \leftarrow \varnothing$
for each $\boldsymbol{d}_i \in \mathcal{D}$ **do**
  Initialize activated token set of features $\mathcal{A}_k \leftarrow \varnothing$
  $H_i \leftarrow \mathcal{M}_\theta(\boldsymbol{d}_i)$
  $Z_i \leftarrow \mathcal{S}_\theta(H_i)$
  for each $\boldsymbol{f}_k \in \mathcal{F}_\theta$ **do**
    $\mathcal{A}_k \leftarrow \mathcal{A}_k \cup \{\boldsymbol{h}_j : \boldsymbol{h}_j \in H_i, \boldsymbol{z}_j = \boldsymbol{e}_j Z_i, \boldsymbol{z}_j \boldsymbol{f}_k^\top > \delta\}$
    if $\mathcal{A}_k \cap \mathcal{T} \neq \varnothing \wedge \mathcal{A}_k \cap \mathcal{V} \neq \varnothing$ **then**
      $F_i \leftarrow F_i \cup \{\boldsymbol{f}_k\}$
    end if
  end for
end for
$\mathcal{D}_R \leftarrow Sort(\mathcal{D}, \{|F_i|\}_{i=1}^n)$

Hyperparameters of Algorithms 1,2,3 The hyperparameters of Algorithms 1,2,3 are shown in Table 3. We ensure that all parameters are the same to ensure a fair comparison between the algorithms.

Hyper-parameters	Cosine similarity	Coocurrence	$\mathcal{L}_0$
top-K	5	5	5
text token vocabulary size	32000	32000	32000
vision token vocabulary size	64	64	64
activation bound	1	1	1
sample data size	1000	1000	1000

Table 3. **Hyper-parameters of Algorithm 1,2,3.**

B.2. Hyperparameter of Model Training

In this section, we list out the hyperparameters used for model training through SFT and DPO (shown in Table B.2). All *SAE* training is performed on 8×A800 GPUs. To ensure fair comparison between algorithms, we maintained consistent parameter settings across all experiments.

Hyper-parameters	SFT	DPO
max length	4096	4096
per device train batch size	8	8
per device eval batch size	8	8
gradient accumulation steps	4	4
LR scheduler type	cosine	cosine
LR	1e-6	1e-6
warmup steps	10	10
eval steps	50	50
epochs	3	3
val size	0.1	0.1
bf16	True	True

Table 4. **Hyperparameters of SFT training and DPO training.**

C. Details of Applying *SAE-V* on Multimodal Data

In this section, we present implementation details of the *SAE-V* application experiments. We enumerate 4 image patch selection algorithms employed in this study and provide additional case analyses. These comprehensive results further demonstrate the robust inference capabilities of *SAE-V*.

C.1. Algorithm

The complete algorithms of $\mathcal{L}_0$, $\mathcal{L}_1$, co-occurring $\mathcal{L}_0$, and cosine similarity score methods are shown in Algorithm 4, Algorithm 5, Algorithm 6 and Algorithm 7.

Algorithm 4 $\mathcal{L}_0$ patch filter

Require: Vision token vocabulary: $\mathcal{V}$; image V; fixed Prompt T; MLLM $\mathcal{M}_\theta$; *SAE-V* $\mathcal{S}_\theta$; features of *SAE-V* $\mathcal{F}_\theta$:$\{\boldsymbol{f}_k\}_{k=1}^n$; activation bound δ; mask rate γ;
Ensure: Filtered image V'
Initialize score of each patch $p_i \leftarrow 0$
$H \leftarrow \mathcal{M}_\theta(T, V)$
$Z \leftarrow \mathcal{S}_\theta(H)$
for each $\boldsymbol{h}_i \in H$ **do**
  if $\boldsymbol{h}_i \in \mathcal{V}$ **then**
    $p_i = \sum_j \mathbf{1}(z_{ij} > \delta)$
  end if
end for
$K \leftarrow \lfloor \gamma |I| \rfloor$
$V' \leftarrow$ *TopK*$(v_i \in V)$ sorted by p_i

Algorithm 5 $\mathcal{L}_1$ patch filter

Require: Vision token vocabulary: $\mathcal{V}$; image V; fixed Prompt T; MLLM $\mathcal{M}_\theta$; *SAE-V* $\mathcal{S}_\theta$; features of *SAE-V* $\mathcal{F}_\theta$:$\{\boldsymbol{f}_k\}_{k=1}^n$; activation bound δ; mask rate γ;
Ensure: Filtered image V'
Initialize score of each patch $p_i \leftarrow 0$
$H \leftarrow \mathcal{M}_\theta(T, V)$
$Z \leftarrow \mathcal{S}_\theta(H)$
for each $\boldsymbol{h}_i \in H$ **do**
  if $\boldsymbol{h}_i \in \mathcal{V}$ **then**
    $p_i = \sum_j (z_{ij})$
  end if
end for
$K \leftarrow \lfloor \gamma |I| \rfloor$
$V' \leftarrow$ *TopK*$(v_i \in V)$ sorted by p_i

These algorithms take images as input and produce masked images, where the masking proportion is determined by the mask rate γ. All algorithms utilize the activation patterns of *SAE-V* features for patch filtering, with their primary distinctions lying in their methods of computing feature activation ($\mathcal{L}_0$, $\mathcal{L}_1$) and measuring cross-modal similarity (co-occurring $\mathcal{L}_0$, cosine similarity score).

Algorithm 6 Co-occuring $\mathcal{L}_0$ patch filter

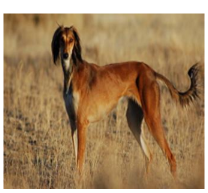
Require: Text token vocabulary $\mathcal{T}$; vision token vocabulary: $\mathcal{V}$; image V; fixed Prompt T; MLLM $\mathcal{M}_\theta$; *SAE-V* $\mathcal{S}_\theta$; features of *SAE-V* $\mathcal{F}_\theta$:$\{\boldsymbol{f}_j\}_{j=1}^n$; activation bound δ; mask rate γ;
Ensure: Filtered image V'
Initialize score of each patch $p_i \leftarrow 0$, co-occuring feature set $F \leftarrow \varnothing$ and activated token set of features $\mathcal{A}_j \leftarrow \varnothing$
$H \leftarrow \mathcal{M}_\theta(T, V)$
$Z \leftarrow \mathcal{S}_\theta(H)$
for each $\boldsymbol{f}_j \in \mathcal{F}_\theta$ **do**
  $\mathcal{A}_j \leftarrow \mathcal{A}_j \cup \{\boldsymbol{h}_i\text{: } \boldsymbol{h}_i \in H, \boldsymbol{z}_i = \boldsymbol{e}_i Z, \boldsymbol{z}_i \boldsymbol{f}_j^\top > \delta\}$
  if $\mathcal{A}_j \cap \mathcal{T} \neq \varnothing \wedge \mathcal{A}_j \cap \mathcal{V} \neq \varnothing$ **then**
    $F \leftarrow F \cup \{\boldsymbol{f}_j\}$
  end if
end for
for each $\boldsymbol{h}_i \in H$ **do**
  if $\boldsymbol{h}_i \in \mathcal{V}$ **then**
    $p_i = \sum_j \mathbf{1}(z_{ij} > \delta \ \wedge \ \boldsymbol{f}_j \in F)$
  end if
end for
$K \leftarrow \lfloor \gamma |I| \rfloor$
$V' \leftarrow$ *TopK*$(v_i \in V)$ sorted by p_i

Algorithm 7 Cosine similarity score patch filter

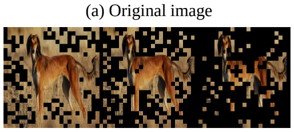
Require: Text token vocabulary $\mathcal{T}$; vision token vocabulary: $\mathcal{V}$; image V; fixed Prompt T; MLLM $\mathcal{M}_\theta$; *SAE-V* $\mathcal{S}_\theta$; features of *SAE-V* $\mathcal{F}_\theta$:$\{\boldsymbol{f}_j\}_{j=1}^n$; activation bound δ; mask rate γ;cosine similarity weight $\{\omega_j\}_{j=1}^n$
Ensure: Filtered image V';
Initialize score of each patch $p_i \leftarrow 0$, co-occuring feature set $F \leftarrow \varnothing$ and activated token set of features $\mathcal{A}_j \leftarrow \varnothing$
$H \leftarrow \mathcal{M}_\theta(T, V)$
$Z \leftarrow \mathcal{S}_\theta(H)$
for each $\boldsymbol{f}_j \in \mathcal{F}_\theta$ **do**
  $\mathcal{A}_j \leftarrow \mathcal{A}_j \cup \{\boldsymbol{h}_i\colon \boldsymbol{h}_i \in H, \boldsymbol{z}_i = \boldsymbol{e}_i Z, \boldsymbol{z}_i \boldsymbol{f}_j^\top > \delta\}$
  if $\mathcal{A}_j \cap \mathcal{T} \neq \varnothing \wedge \mathcal{A}_j \cap \mathcal{V} \neq \varnothing$ **then**
    $F \leftarrow F \cup \{\boldsymbol{f}_j\}$
  end if
end for
for each $\boldsymbol{h}_i \in H$ **do**
  if $\boldsymbol{h}_i \in \mathcal{V}$ **then**
    $p_i = \sum_j \mathbf{1}(z_{ij} > \delta \ \wedge \ \boldsymbol{f}_j \in F)\, \omega_j$
  end if
end for
$K \leftarrow \lfloor \gamma |I| \rfloor$
$V' \leftarrow \mathit{TopK}(v_i \in V)$ sorted by p_i

C.2. Case Analysis

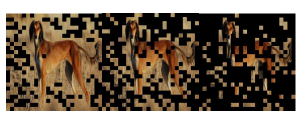
We present 4 cases in Figure 12, corresponding to each of our metric in Section 3.2. The cases intuitively show that $\mathcal{L}_0$ method and cosine similarity score method are more capable of identifying significant patches in images compared to other methods, which aligns with the quantized results shown in Figure 6.



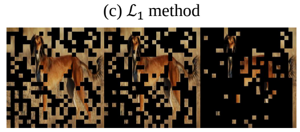
(a) Original image



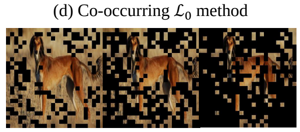
(b) $\mathcal{L}_0$ method



(c) $\mathcal{L}_1$ method



(d) Co-occurring $\mathcal{L}_0$ method



(e) Cosine similarity score method

Figure 12. **Case Analysis of all image patch filtering methods in Section 3.2.** We present the original image (a) and 4 case for methods, $\mathcal{L}_0$ (b), $\mathcal{L}_1$ (c), co-occurring $\mathcal{L}_0$ (d) and cosine similarity score (e). Each case contains 3 images as preserving top 75% patches, top 50% patches and top 25% patches.

We present 5 cases filtered with the cosine similarity score method in Figure 13. The results show that *SAE-V* model performs excellently in capturing critical patches in images.

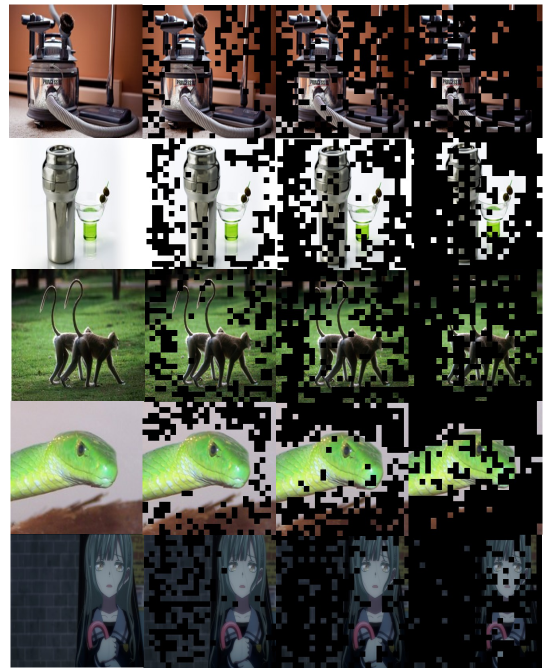


Figure 13. **Case Analysis of cosine similarity score method in Section 3.2.** 5 cases filtered with cosine similarity score method are shown in the Figure. Each case contains contains 4 images as original image, preserving top 75% patches, top 50% patches and top 25% patches.