

Enhancing Few-Shot Vision-Language Classification with Large Multimodal Model Features

Chancharik Mitra^{1*} Brandon Huang^{2*} Tianning Chai² Zhiqiu Lin¹ Assaf Arbelle³
 Rogerio Feris⁴ Leonid Karlinsky⁴ Trevor Darrell² Deva Ramanan¹ Roei Herzig^{2,4}

¹Carnegie Mellon University ²University of California, Berkeley
³IBM Research ⁴MIT-IBM Watson AI Lab

Abstract

Generative Large Multimodal Models (LMMs) like LLaVA and Qwen-VL excel at a wide variety of vision-language (VL) tasks. Despite strong performance, LMMs’ generative outputs are not specialized for vision-language classification tasks (i.e., tasks with vision-language inputs and discrete labels) such as image classification and multiple-choice VQA. One key challenge in utilizing LMMs for these tasks is the extraction of useful features from generative LMMs. To overcome this, we propose an approach that leverages multimodal feature extraction from the LMM’s latent space. Toward this end, we present **Sparse Attention Vectors (SAVs)**—a finetuning-free method that leverages sparse attention head activations (fewer than 5% of the heads) in LMMs as strong feature representations. With only few-shot examples, SAVs demonstrate state-of-the-art performance compared to a variety of few-shot and finetuned baselines on a collection of vision-language classification tasks. Our experiments also imply that SAVs can scale in performance with additional examples and generalize to similar tasks, establishing SAVs as both effective and robust multimodal feature representations.

1. Introduction

Generative Large Multimodal Models (LMMs) such as GPT-4V [73], LLaVA [59, 60], and QwenVL [3] demonstrate state-of-the-art performance on open-ended vision-language (VL) tasks like image captioning [54, 107], visual question answering [2, 32, 43], and language grounding [38, 65]. However, despite their remarkable performance on generative tasks, these models struggle on vision-language classification tasks, where responses are a discrete set of labels [8, 111]. Indeed, LMMs with billions of parameters and trained on trillions more tokens of data underperform smaller encoder VLMs like CLIP and SigLIP [8, 111]

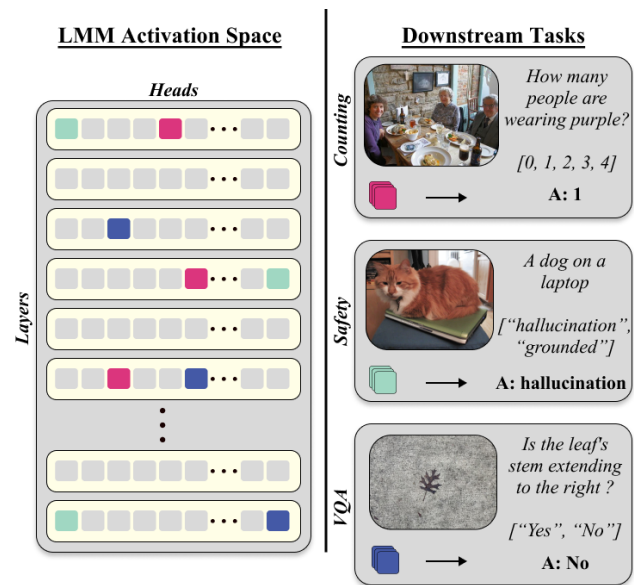


Figure 1. **Sparse Attention Vectors (SAVs) Overview.** We develop a method for extracting features from a generative LMM without finetuning. We first extract a sparse set of attention vectors for each task given a set of few-shot examples, and then, we utilize these attention vectors directly as features for downstream vision-language classification tasks.

and even classical machine learning methods [6] on image classification tasks. One reason that CLIP-like models excel at classification is the ease of extracting visual or textual features from modality-specific encoders. However, such models cannot process *joint* vision-language inputs, as LMMs can. Our goal is to extract compact features from generative LMMs, enabling downstream discrete labeling tasks such as classification and multiple-choice VQA.

Feature extraction is a well-explored field in both vision-only [15, 83, 94] and language-only encoder models [13, 81], but such is not the case with generative models. Most

current methods for extracting features from generative models require carefully constructed prompts [41, 67, 104], specialized architectures [51], few-shot prompting [7, 11, 112], and finetuning [8, 64, 111]. Recent work [111] shows that prompting approaches fail to close the gap with encoder VLMs, while finetuning requires training-scale data for *every* unseen task, which is inefficient. However, as we observe in our work, generative models still offer the promise of more flexible, truly multimodal features as compared to modality-specific features from CLIP-like models.

A source of inspiration for our method is long-standing work in neuroscience that suggests certain areas of the brain are reserved for specific tasks [17, 37] (i.e. functional specificity). Motivated by this idea, we refer to recent interpretability research that has focused on identifying specific heads in transformer-based models that correspond to particular tasks [72]. The most prominent of these methods is a line of work that looks to enhance vision-language capabilities using task vectors [24, 27, 30, 93], which are compact implicit representations of tasks encoded in the activations of a transformer model. While promising, these methods ultimately use these representations to augment a model’s generative capabilities. On the other hand, we seek to use feature representations directly as classifiers. Nevertheless, this intuition from interpretability informs our work on Sparse Attention Vectors (SAVs), which are sparse features in an LMMs activation space that can be directly exploited for few-shot VL classification.

Our method has three steps: First, we extract features (i.e., attention vectors) from the output of each head of the LMM for some few-shot labeled examples (≈ 20 per label). Second, we average these attention vectors over the examples in each class and evaluate their accuracy as centroids in a class centroid classifier. We then select the top 20 heads by classification accuracy as our SAVs. In this way, we identify a very *sparse* set of attention vectors (less than 5% of the total number of heads) that can be used for discriminative tasks. Finally, we perform inference on the given task by doing a majority vote across this sparse set of attention vectors for each new query. This approach requires only few-shot examples at test-time to extract effective multimodal embeddings. An overview is shown in Figure 1.

We summarize our main contributions as follows: (i) We introduce a novel method that yields a sparse set of attention vectors to serve as highly effective features for individual VL classification tasks; (ii) Our method surpasses zero-shot, few-shot, and LoRA fine-tuned baselines across multiple tasks (+7% improvement on average over LoRA on challenging benchmarks like BLINK [18], VLGuard [114], and NaturalBench [44]); (iii) We establish several advantageous properties of our approach, including strong generalization capabilities and favorable scaling characteristics.

2. Related Works

Controllable Generation for Classification. Controllable text generation in LMMs guides model outputs to meet specific constraints. For classification tasks with generative LMMs, several approaches exist: test-time hard prompting [7, 104] uses prompt engineering or few-shot examples to elicit class label outputs [11, 88, 102, 105, 111]; direct probability analysis of generated class labels [55, 58] for image-text retrieval; and soft prompting methods that finetune learnable tokens [42, 49]. Other techniques include instruction finetuning [103] on labeled data [6, 111] and preference modeling like DPO and RLHF [16, 74, 79]. Our method, however, is finetuning-free and directly selects class labels without preference data. Most related work shows that internal transformer representations called task vectors [23, 27, 30, 92] can encapsulate ICL example tasks. Beyond previous approaches, we use a sparse set of attention vectors directly as VL classification task features.

Vision-Language Features. Feature extraction seeks useful representations for diverse downstream tasks. Early embedding techniques including autoencoders [4, 39, 62, 63, 82], Word2Vec [66] and GloVe [77] transformed inputs into vector representations, followed by advances in NLP [13, 19, 68, 81] and vision [15, 83, 94]. Recent methods like CLIP and SigCLIP [46, 47, 50, 78, 108, 109] explore multimodal correlations through contrastive learning or sigmoid loss. These representations offer flexibility across tasks [12, 25, 33, 40, 80, 95] and domains [53, 101, 110].

Interestingly, extracting features from generative models poses unique challenges in identifying optimal extraction points. Some approaches finetune encoder VLMs on LLM-generated data [48, 97] or finetune LMMs directly on specific tasks [31, 111]. More efficient methods finetune encoder-decoder representations for better alignment [36, 64, 69, 70]. Finetuning-free approaches use distillation prompts to extract representations from model weights or activations [34, 35, 41, 61], while other methods employ mixture-of-experts [52], expert models [99], or embedding reranking [21].

Thus, current SOTA faces the following challenges: (1) modality-specific rather than multimodal features, (2) requiring finetuning, (3) limited flexibility due to specialized prompts, and (4) dependence on multiple models. Our approach provides effective multimodal embeddings without any gradient-based finetuning and flexibly apply to various VL classification tasks without additional models.

3. Methods

In this section, we outline our approach for using sparse attention vectors from the activation space of a transformer-based large multimodal model (LMM) as features for any VL classification task. The method consists of three main

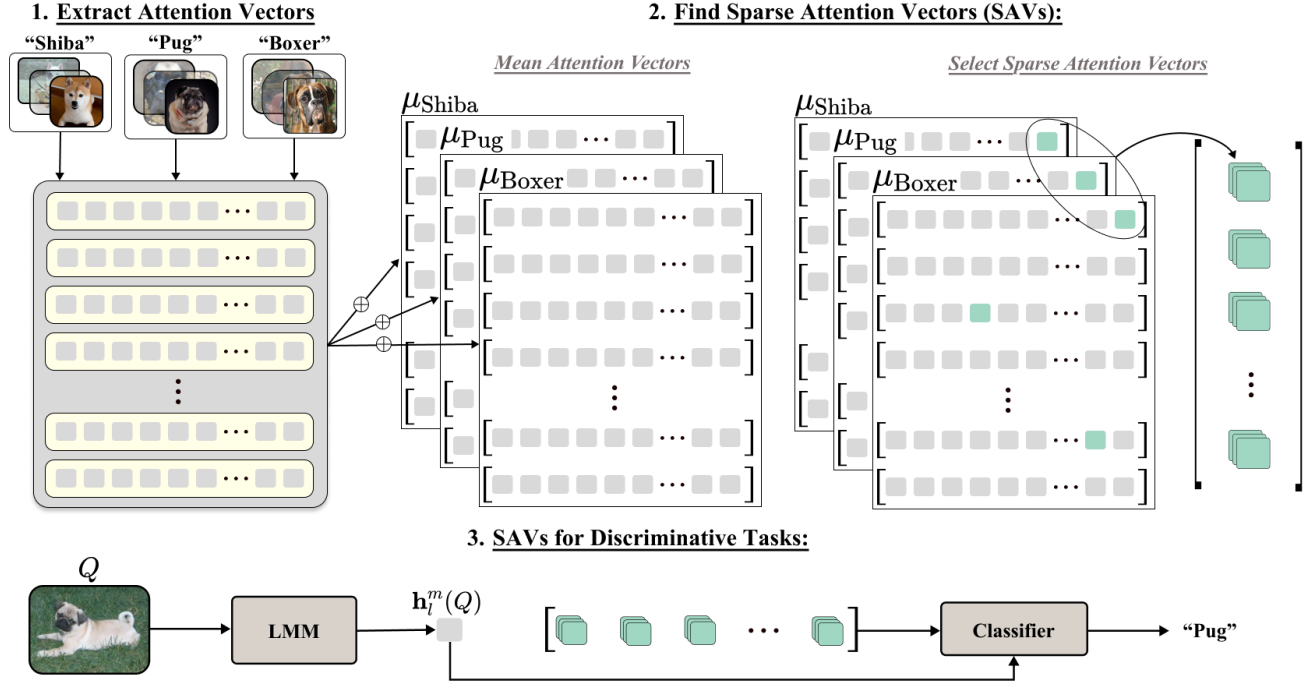


Figure 2. **Sparse Attention Vectors (SAVs) Detailed View.** Our method is broken into the following three parts: (1) Given a set of few shot examples to be classified by a frozen LLM, we extract attention vectors across all heads and layers. (2) These attention vectors are averaged across the set of examples for each class. For each head, we use these mean attention vectors as centroid classifiers, which are then used to select a sparse set of k heads with the highest classification accuracy. (3) Finally, we use these sparse attention vectors to directly classify new inputs via majority vote (denoted by the “Classifier” block).

steps: (i) extracting the attention vectors from all attention heads in the model, (ii) identifying a sparse set of vectors based on their ability to consistently return the correct label for some support set of examples, and (iii) using these sparse features to classify new queries. We begin with a formal description of the transformer decoder LLM and its attention mechanism, followed by the detailed methodology for sparse attention vector selection and classification. A detailed view of our method is shown in Figure 2.

3.1. Preliminaries

A transformer-based large language model (LLM) with L layers and H attention heads per layer processes input sequences through multi-head self-attention mechanisms. Each layer combines multiple attention heads to capture different aspects of the input sequence, followed by feed-forward networks for further processing.

Multi-Head Attention. Let $x = \{x_1, x_2, \dots, x_T\}$ represent a sequence of input tokens, where x_i is the i^{th} token. For each layer $l \in \{1, \dots, L\}$, the input sequence is projected into queries, keys, and values for each attention head $m \in \{1, \dots, H\}$. Each head performs the following scaled dot-product attention:

$$\mathbf{h}_l^m(x_i) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_m}} \right) V$$

where Q , K , and V are the query, key, and value matrices respectively, and the dimensionality of each head d_m which is given by $\frac{d}{H}$ (the embedding dimension divided by the number of heads). We denote $\mathbf{h}_l^m(x_i)$ as an *attention vector* for head m in layer l .

The outputs of all heads are concatenated and projected to form the layer output:

$$\text{MultiHead}(x_i) = \text{Concat}(\mathbf{h}_l^1(x_i), \dots, \mathbf{h}_l^H(x_i))W^O$$

where W^O is the output projection matrix.

In our work, we look to leverage attention vectors for the purpose of vision-language classification tasks. Specifically, the attention vectors are used as latent representations of the inputs to both find attention heads in an LLM suited for a classification task and then perform downstream inference using those selected attention heads. We describe our method in detail in the sections that follow.

3.2. Sparse Attention Vectors

Our key insight is that within the many attention heads and transformer layers of an LLM, there exists a sparse subset

that can serve as effective features for vision-language classification tasks. We present a three-step method to identify and utilize these features to build lightweight classifiers.

Step 1: Extracting Attention Vectors. Given a frozen LMM and few-shot examples of sequence-label pairs $\{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ we first extract the attention vectors for each sequence x_i . Specifically, we compute the attention vector $\mathbf{h}_l^m(x_i^T)$ for head m from layer l for the final token x_i^T . This yields a set of attention vectors $\{\mathbf{h}_l^m(x_i^T) \mid i = 1, \dots, N\}$ for each head m and layer l .

Step 2: Identifying Relevant Vectors. The central question is how to identify which attention vectors are naturally suited for the classification task at hand. We evaluate each vector’s ability as a *local classifier* by computing its performance under a nearest class centroid classifier.

Specifically, for each class $c \in \mathcal{C}$, compute its centroid (or mean) attention vector across the few shot examples:

$$\mu_c^{l,m} = \frac{1}{|N_c|} \sum_{j:y_j=c} \mathbf{h}_l^m(x_j^T)$$

where $N_c = \{j : y_j = c\}$ is the set of indices of examples with label c . For each input x_i , we compute its cosine similarity to each class centroid head:

$$s_{l,m}(x_i, c) = \frac{\mathbf{h}_l^m(x_i^T) \cdot \mu_c^{l,m}}{\|\mathbf{h}_l^m(x_i^T)\| \|\mu_c^{l,m}\|}, \quad \forall c \in \mathcal{C}$$

Next, we measure the classification accuracy of each head (i.e. local classifier):

$$\text{score}(l, m) = \sum_{i=1}^N \mathbf{1}[\hat{y} = y_i]$$

where the nearest class centroid label is given as $\hat{y} = \arg \max_{c \in \mathcal{C}} s_{l,m}(x_i, c)$, and $\mathbf{1}[\cdot]$ is the indicator function that evaluates to 1 when the condition is true (and 0 otherwise). We denote the set of k top-scoring heads as \mathcal{H}_{SAV} :

$$\mathcal{H}_{\text{SAV}} = \{(l, m) \mid \text{score}(l, m) \text{ is among } k \text{ highest scores}\}$$

Step 3: Classification with Sparse Attention Vectors.

Given a query sequence Q to classify, we leverage our sparse set of heads \mathcal{H}_{SAV} for prediction. For each head $(l, m) \in \mathcal{H}_{\text{SAV}}$, we compute the class centroid $\mu_c^{l,m}$ closest to the query as follows:

$$\hat{y}_{l,m} = \arg \max_{c \in \mathcal{C}} s_{l,m}(Q^T, c)$$

where $s_{l,m}(\cdot, \cdot)$ is defined as in Step 2. Our final class prediction is that of a *global classifier* that counts the majority vote across all local classifiers (heads) in \mathcal{H}_{SAV} :

$$\arg \max_{y \in \mathcal{C}} \sum_{(l,m) \in \mathcal{H}_{\text{SAV}}} \mathbf{1}[\hat{y}_{l,m} = y]$$

This approach reveals a surprising capability of LMMs: with just a few carefully selected attention heads ($|\mathcal{H}_{\text{SAV}}| \ll LH$), we can transform a generative language model into a lightweight vision-language classifier. This finding suggests that classification-relevant features naturally emerge within specific attention heads during model pretraining. In theory, one could replace our local and global classifiers with more complex variants (e.g. linear models, MLPs), something we explore in 3a.

4. Evaluation

We apply SAVs to two state-of-the-art LMMs—LLaVA-OneVision [45] and Qwen2-VL [98]. We also do a rigorous comparison of our method to strong few-shot and finetuning baselines on a variety of different image-text and image-only vision-language classification tasks covering safety, VQA, image-text retrieval, and simple classification.

4.1. Implementation Details

We implemented our approach in PyTorch [76]. We use the official implementations of each model, and all of our experiments can be run on a single NVIDIA A100 GPU. We use the default hyperparameters provided by the original models’ (LLaVA-OV and Qwen2-VL) codebases. These hyperparameters are detailed in Table 8. More details of the implementation is included in the supplementary in Section 8.

4.2. Benchmarks

Safety. (1) LMM-Halucination [9] is a dataset which evaluates the hallucinations of the models when answering multimodal tasks. We report the raw classification accuracy of our method. Thus, the set of class labels for this task is given by $\mathcal{C} = \{\text{“hallucinating”}, \text{“not hallucinating”}\}$. (2) VGuard [114] is a dataset focusing on vision-language safety which identifies 4 main categories of harmful content: Privacy, Risky Behavior, Deception and Hateful Speech. VGuard proposes Attack Success Rate (ASR) for evaluating unsafe inputs and Helpfulness for evaluating safe inputs. We simply reformat it as a classification task, where the set of class labels is given by $\mathcal{C} = \{\text{“safe”}, \text{“unsafe”}\}$.

VQA Datasets. In our work, we evaluate on VQA benchmarks, many of which can be formulated as a classification task. (1) BLINK [18] contains many tasks that are intuitive for humans but complicated for multimodal models such as multi-view reasoning, and visual similarity comparison. Since potential answers in BLINK are multiple choice, the class labels would be given as $\mathcal{C} = \{\text{“A”}, \text{“B”}, \text{“C”}, \text{“D”}\}$ (note: the number of labels depends on the possible number of options allowed for a task). (2) NaturalBench-VQA [44] is a compositional dataset collected from natural image-text corpora but validated with human filtering. Each sample of the dataset contains two questions on compositional

Model	Image-Text Tasks											Image-Only Tasks				
	Safety		VQA					I-T Retrieval				Classification				
	MHalu	VLGuard	BLINK	VizWiz	NB (T)	NB (U)	NB (G)	NB-r (T)	NB-r (U)	NB-r (G)	SC	EuroSAT	Pets	Imagenet-1k	Flower	CUB
CLIP	-	-	-	-	-	-	-	41.8	45.0	23.2	35.3	64.0	88.1	96.1	92.8	97.8
SigLip	-	-	-	-	-	-	-	54.5	54.9	31.2	42.7	63.9	98.3	97.6	95.8	98.4
GPT-4o [†]	39.4	45.7	59.0	58.1	64.6	66.4	39.6	65.0	67.0	40.5	48.3	70.5	98.3	98.3	99.0	99.3
LLaVA-1.5	44.3	31.0	38.0	50.0	37.7	43.8	12.7	36.7	42.7	12.2	24.9	26.3	43.0	20.6	51.8	49.0
Instruct-BLIP	44.0	18.5	38.7	34.5	20.2	24.2	4.0	19.5	21.3	1.1	4.7	20.4	56.0	11.0	18.2	33.8
LLaVA-OV-7B	34.7	31.4	45.0	60.4	52.0	53.3	27.0	56.2	58.0	32.1	15.3	66.5	88.1	92.5	83.2	85.3
+4-shot-ICL	25.0	35.0	38.9	47.8	47.6	50.4	22.2	48.2	49.6	31.5	16.1	47.1	63.9	49.0	63.8	60.6
+MTV	37.3	32.9	44.5	61.1	56.2	58.0	30.7	58.1	59.5	33.4	28.7	65.5	88.5	<u>95.9</u>	83.2	85.6
+LoRA	<u>78.3</u>	<u>90.0</u>	<u>47.0</u>	<u>63.1</u>	<u>58.6</u>	<u>60.9</u>	<u>32.4</u>	<u>59.4</u>	<u>60.3</u>	<u>35.4</u>	<u>30.4</u>	<u>85.0</u>	<u>96.8</u>	<u>93.2</u>	<u>91.2</u>	<u>91.8</u>
+SAVs	80.8	94.3	51.8	66.1	60.3	62.3	35.1	72.7	73.0	53.1	37.6	86.7	97.0	97.5	99.6	97.5
	+46.1	+62.9	+6.8	+5.7	+8.3	+9.0	+8.1	+16.5	+15.0	+20.5	+22.3	+20.2	+8.9	+5.0	+16.4	+12.2
Qwen2-VL-7B	24.0	26.9	43.3	68.3	53.8	56.6	28.5	60.2	61.9	35.6	24.9	54.7	92.6	84.4	93.7	93.2
+4-shot-ICL	40.4	52.9	37.6	67.1	38.2	41.3	15.2	42.4	45.6	22.7	25.2	29.4	43.5	43.8	59.4	48.4
+MTV	32.3	21.9	41.9	<u>68.5</u>	54.8	57.3	<u>29.7</u>	63.5	64.0	37.0	<u>40.7</u>	52.3	91.7	91.2	94.3	94.1
+LoRA	<u>84.8</u>	<u>87.7</u>	<u>46.3</u>	70.8	<u>55.3</u>	<u>57.4</u>	28.8	<u>65.2</u>	<u>66.1</u>	<u>40.4</u>	38.4	<u>72.9</u>	98.4	<u>96.4</u>	<u>97.1</u>	<u>95.0</u>
+SAVs	85.1	96.0	47.2	68.3	57.6	60.9	32.3	70.0	71.0	42.5	47.5	79.9	<u>98.1</u>	97.6	99.8	98.7
	+61.1	+69.1	+3.9	+0	+3.8	+4.3	+3.8	+9.8	+9.1	+13.8	+22.6	+25.2	+5.5	+13.2	+6.1	+5.5

Table 1. **Results** evaluation on Safety, Visual Question Answering (VQA), Image-Text Retrieval (I-T Retrieval), and Image Classification benchmarks. The best result for each generative model is shown in **bold** and the second best in underline. We **gray** out additional baselines. We note that CLIP and SigLIP cannot be evaluating directly on tasks with interleaved image-text queries. [†] GPT-4o is a close-source model and as such, is shown just as an upperbound (since SAVs are not directly applicable). Key: NB - NaturalBench, SC - SUGARCREPE.

differences between two similar images, making NaturalBench especially challenging for any existing VL models. The class labels are $\mathcal{C} = \{“A”, “B”\}$. As suggested in the paper, we evaluate “question accuracy” which awards one point if a model correctly answers a question for both images, “image accuracy” which awards a point when a model answers both questions for an image, and finally “group accuracy” awards one point when a model correctly answers all four pairs. (3) Vizwiz [20] is VQA dataset designed to progress research in vision systems to assist blind and vision-impaired individuals. The dataset was collected by asking blind people to take pictures and record questions about the image. Although typically a generative task, we reformulate Vizwiz as first a classification task distinguishing answerable and unanswerable questions followed by standard response generation. The class labels are $\mathcal{C} = \{“answerable”, “unanswerable”\}$.

Image-Text Retrieval. NaturalBench-Retrieval [44] and SUGARCREPE [28] both measure fine-grained semantic understanding in image-text pairs, with class labels $\mathcal{C} = \{“Yes”, “No”\}$ indicating whether an image-text pair is correctly matched. While SUGARCREPE presents one image with two captions (original and altered), NaturalBench-Retrieval adds complexity by using two similar images with two corresponding captions, effectively eliminating lan-

guage bias and requiring models to capture more nuanced visual-semantic relationships.

Image Classification. Image classification tasks evaluate a model’s ability to categorize images into predefined classes. We evaluate on several standard classification benchmarks: EuroSAT [22] (satellite imagery for land use classification), Oxford-IIIT-Pets [75] (pet breed identification with visually similar categories), Flowers [71] (flower species recognition), Caltech Birds (CUB) [96] (fine-grained bird species classification), and ImageNet-1k [10] (general object recognition). For each dataset except ImageNet, we formulate the task as a 4-way multiple choice question with labels $\mathcal{C} = \{“Class 1”, “Class 2”, “Class 3”, “Class 4”\}$. Due to model oversaturation in simpler formulations, ImageNet-1k is formulated as a 16-way classification problem.

4.3. Baselines

For our results, we utilized SAVs with 20 examples per label. We compared our method with multiple SOTA baselines, including GPT-4o [73]. As a closed-source model, GPT-4o is presented simply as a strong closed-source baseline. Furthermore, we present the results of classic LMMs, LLaVA-1.5 and InstructBLIP, which have been previously applied to image classification [111]. Zero-shot (ZS) baselines are implemented by querying the model directly and generating a response. For image-text retrieval, ZS uses the

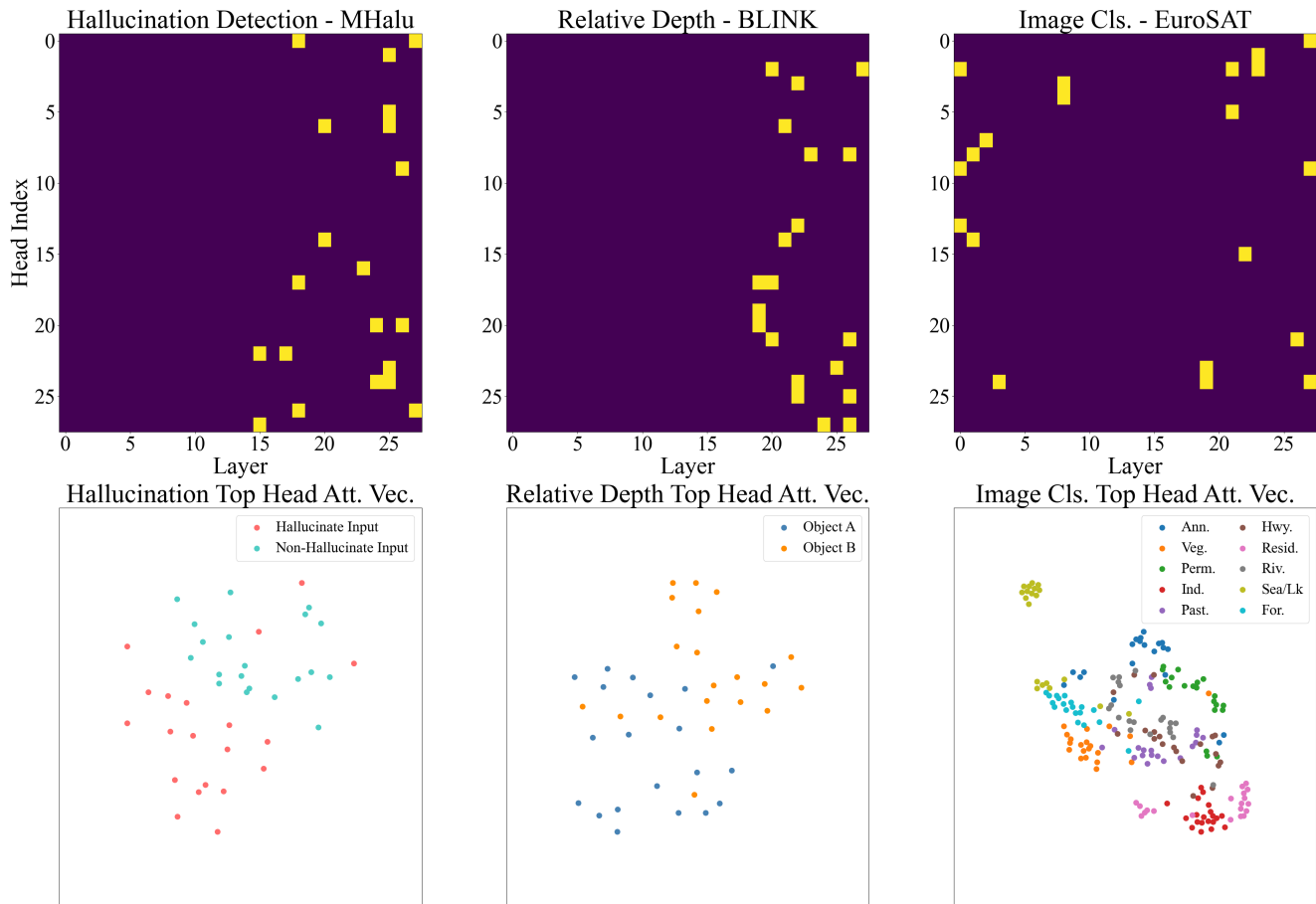


Figure 3. **Head Visualization.** On the top row, we show the top-20 attention head locations for a given task. All the heads are indexed by their (head, layer) position, and the selected heads are highlighted. On the bottom row, we visualize the attention vector of few-shot examples for the top head of the given class with t-SNE clustering [26]. Each point represents the embedding of an input sample.

SOTA generative scoring method VQAScore [57]. In addition to ZS, we also compare to several test-time and finetuning few-shot methods (all with the exact same sample complexity as SAVs). For instance, we compare to the current SOTA multimodal few-shot method, MTV [30] as well as doing 4-shot ICL and LoRA [29] finetuning for each task.

4.4. Results

Results are shown in Table 1. An advantage of our method is its adaptability to any VL classification task that has image, text, or interleaved image-text inputs. Thus, we apply SAVs to a wide range of tasks in safety, VQA, image-text retrieval and image classification. Furthermore, our approach even significantly improves over SOTA ZS, few-shot, and finetuning methods. One interesting observation is that few-shot ICL consistently deteriorates performance. We hypothesize that instruction tuning of LMMs disrupts the few-shot prompting capabilities gained during pretraining in favor of structured instructions as shown

in [14, 86]. On the other hand, SAVs overcome this by directly operating on the internal activations. Finally, not only is our method successful across a wide-range of tasks, but it also improves on challenging visual perception and compositional reasoning tasks (e.g., BLINK and NaturalBench) that all VL models struggle with. Please refer to our supplementary section in Section 7 for mor results.

4.5. Ablations

We perform a comprehensive ablation study of our method on MHalubench, NaturalBench, and EuroSAT (see Table 2). For more ablations, please refer to Section 7.1 in the Supp. For all ablations, we use LLaVA-OneVision-7B.

Varying number of examples. In Figure 4 (left), we examine the impact of varying the number of few-shot examples used in our method. Our primary results in Table 1 indicate that just 20 examples per label are necessary to yield state-of-the-art performance on a variety of VL classification tasks. This ablation shows that accuracy on these tasks

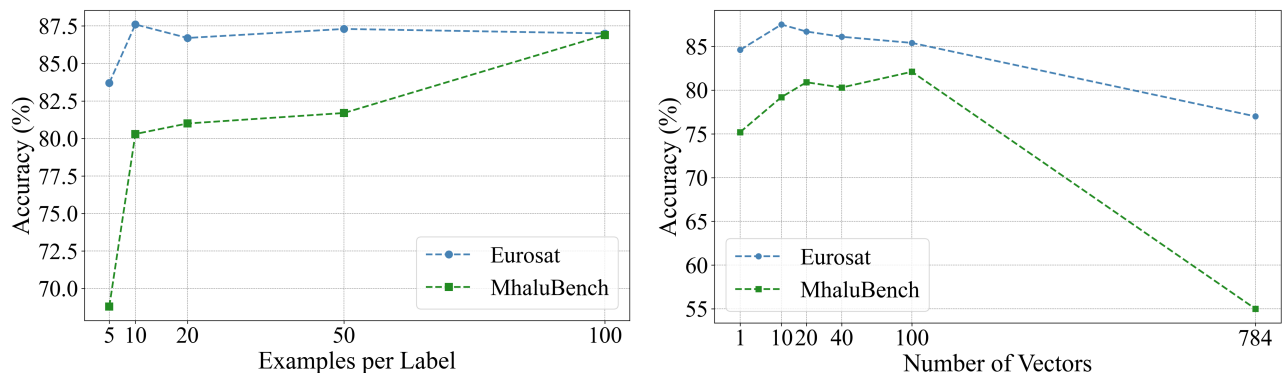


Figure 4. **Scaling Property of SAVs.** Performance of LLaVA-OneVision-7B + SAVs on varying number of few-shot examples per label (left). Performance of LLaVA-OneVision-7B + SAVs on varying numbers of attention vectors used (right).

can scale with increasing numbers of examples per label.

Varying number of attention vectors. In order to extract attention vectors, we select a very sparse set of heads from hundreds. We vary the number of attention vectors selected in Step 2 of our method and demonstrate that just 20 vectors are enough to realize nearly all of the classification accuracy of our method. Results are shown in Figure 4 (right).

SAVs are flexible to different evaluation strategies. In our method, we leverage class centroid classification as the evaluation method for selecting sparse features. We view this flexibility of the sparsification method to be a key feature of our work. As such, we compare our class centroid classification approach to k-nearest neighbors (KNN) and linear probing. For linear probing, we train a lightweight MLP module for 20 epochs using the top heads’ features. All methods make use of the same 20 examples per label for consistency. Our results in Table 3a show that class centroid classification outperforms both KNN classification and is comparable with linear probing.

Comparing heads vs. layers. Based on prior work and transformer-architecture intuition, we treat the attention vectors outputted by the heads as a viable set of features for classification tasks. We verify this intuition by comparing the performance of selecting 2 sparse layers to selecting sparse heads as feature maps for our tasks. As shown in Table 3b, head-based attention vectors outperform concatenated layer features on all three benchmarks.

4.6. Additional Experiments

In this subsection, we present experiments that demonstrate additional properties and capabilities of SAVs, beyond its use as features for VL classification tasks. Additional experiments can be found in Section 7.2 of the Supplementary. For all experiments, we use LLaVA-OneVision-7B.

Visualizing SAVs. SAVs are both an efficient and interpretable method for leveraging generative LMMs for VL

classification tasks. To emphasize this point, we show the selected heads for hallucination detection, relative depth, and image classification in the first row of Figure 3. The visualizations demonstrate both the sparsity and specificity of the SAVs that are used for each task. Unlike prompting and finetuning methods, our approach clearly outlines exactly where in the model’s activation space informative attention vector features are extracted from. We furthermore show that the features extracted from these heads are useful for the given task in the second row of the figure, where we visualize the features outputted by the top selected head for each few-shot sample via t-SNE [26]. The clear clustering of examples of the same label indicates that even with a single head, high-quality features are being selected as SAVs.

Evaluating SAVs generalizability. Here, we ask whether SAVs extracted from one task, can generalize to another similar task. We utilize SAV heads from VLGard to evaluate on MhaluBench and vice versa. We do a similar approach with LoRA, by applying LoRA finetuned on VLGard to MhaluBench. Interestingly, our results in Table 2a show that SAVs generalize between tasks, while LoRA weights, as expected, overfit to the finetuned task.

Comparing SAVs to CLIP/SigLIP on interleaved image-text tasks. As discussed in Section 1, SAVs are fully multi-modal features able to represent inputs that are image-only, text-only, and even interleaved image-text. This is something that is not possible to directly replicate with CLIP and SigLIP models which have separate image and text encoders. Nevertheless, we compare our method to both CLIP and SigLIP on tasks that require interleaved image-text inputs. While SAVs can do this natively, we enable this comparison by concatenating the separate image and text features of CLIP and SigLIP in order to evaluate on MhaluBench and NaturalBench. We find that our method vastly outperforms concatenated CLIP and SigLIP features on both benchmarks as shown in Figure 2b. This result

(a) Generalization			(b) Interleaved Tasks			(c) SAVs vs Few-Shot SigLIP		
	MHB	VLG		MHB	NB		NB	ES
Zero-Shot	34.7	31.4	CLIP	51.9	1.2	SigLIP Few-shot Mean	1.2	88.2
VLG LoRA	34.1	90.0	SigLIP	48.6	1.2	SigLIP Few-shot Cross-Modal	6.8	86.4
VLG SAV	55.3	94.3	SAVs	80.8	35.1	SAVs	35.1	86.7

Table 2. **Additional SAVs Experiments.** We (a) demonstrate the generalization of SAV heads to similar tasks, (b) show the effectiveness of SAV for tasks with interleaved image-text inputs, and (c) compare SAVs with few-shot formulations of CLIP and SigLIP.

(a) Classification Methods				(b) Sparse Configurations			
	MHB	NB	ES		MHB	NB	ES
Class Centroid	80.8	35.1	86.7	Sparse Heads	80.8	35.1	86.7
KNN	53.0	11.0	78.1	Sparse Layers	79.0	28.4	81.8
Linear Probe	82.5	32.9	83.1				

Table 3. **SAVs Ablations.** We perform several ablations to identify the important aspects of our method that contribute to its effectiveness. In particular, we compare the effectiveness of (a) different classification methods and (b) head feature sparsification versus layer feature sparsification. Note: MHB represents MHALuBench, NB represents NaturalBench Group Score, and ES represents EuroSAT. For more ablations, please refer to Section 7.1 in the Supplementary.

demonstrates the adaptability of our method to any VL classification regardless of the input’s modality.

Comparing SAVs to few-shot SigCLIP. As SAVs are extracted with few-shot examples, we compare our method to an analogous version of few-shot SigLIP. However, because SigLIP cannot be directly made few-shot, we adapt SigLIP as a few-shot class centroid classifier. One method used is the current SOTA few-shot classification method for encoder models Cross-Modal Adaptation [56]. We apply a second method where we aggregate CLIP/SigLIP embeddings into a mean embedding for each label. Then, just as in SAVs, we perform class centroid classification for each query using our set of mean SigLIP embeddings. We note that for image classification like EuroSAT, only image embeddings are needed, but for MHALuBench, multimodal embeddings are necessary. SAVs are inherently multimodal and so can be flexibly applied to both, but CLIP/SigLIP only have image-only or text-only embeddings. To overcome this, we use SigLIP image features for EuroSAT and concatenate image and text features for MHALuBench. Interestingly, while SigLIP is comparable to SAVs on EuroSAT, our method *vastly* outperforms SigLIP in the few-shot setting for MHALuBench, suggesting generalizability of our method for a variety of multimodal tasks that CLIP-like struggle with. Our results are shown in Table 2c.

5. Conclusion

Our research demonstrates the effectiveness of extracting Sparse Attention Vectors (SAVs) from the heads of an LMM and utilizing them directly for vision-language classification. Our method stands out by using only few-shot exam-

ples per label and only less than 1% of the heads to outperform zero-shot, few-shot, and fine-tuned baselines on a variety of image-text and image-only tasks. In addition, SAVs allows generative LMMs to close the gap with closed-source GPT-4o while also being an interpretable method that can generalize to similar tasks. Our ablations reveal the flexibility of using any classification method as a sparsification method for attention vectors and also shows that features are found as outputs of heads rather than layers. Overall, these results show that SAVs is a lightweight, performant, and generalizable method for extending generative LMMs’ multimodal classification abilities. We are encouraged by the outcomes, and anticipate many directions for future work. In addition to methodological improvements, we look forward to the application of SAVs as features for multimodal retrieval, data compression, or more generally as a distilled representation for downstream models.

6. Limitations

Sparse Attention Vectors are a significant step in generalizing the capabilities of generative LMMs to classification tasks. Nevertheless, it is valuable to consider certain limitations of our approach. SAVs are a method that requires access to the model’s internal architecture and so may not be directly applicable to closed-source models like GPT-4 [73] and Gemini [89, 90]. Additionally, some tasks like image-text retrieval [28, 91] can benefit from more fine-grained confidence metrics attached to each label than proportion of voting heads per label. These challenges prompt future work in these directions as well as exciting questions about how to use SAVs as feature embeddings for other tasks.

Acknowledgements.

We would like to thank Abrar Anwar and Tyler Bonnen for helpful feedback and discussions. This project was supported in part by DoD, including PTG and/or LwLL programs, as well as BAIR’s industrial alliance programs.

References

- [1] S. R. Bowman, A. Williams, N. Nangia. A broad-coverage challenge corpus for sentence understanding through inference. *ArXiv*, 2017. 2
- [2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *IEEE International Conference on Computer Vision*, pages 2425–2433, 2015. 1
- [3] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *ArXiv*, abs/2308.12966, 2023. 1
- [4] Pierre Baldi. Autoencoders, unsupervised learning, and deep architectures. In *ICML 2011 Unsupervised and Transfer Learning Workshop*, 2011. 2
- [5] Rohan Bavishi, Erich Elsen, Curtis Hawthorne, Maxwell Nye, Augustus Odena, Arushi Somani, and Sağnak Taşirlar. Introducing our multimodal models, 2023. 3
- [6] Matyas Bohacek and Michal Bravansky. When XGBoost outperforms GPT-4 on text classification: A case study. In *Trustworthy Natural Language Processing (TrustNLP) 4th Workshop*, pages 51–60, Mexico City, Mexico, 2024. Association for Computational Linguistics. 1, 2
- [7] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020. 2
- [8] Martin Juan Jos’e Bucher and Marco Martini. Fine-tuned ‘small’ llms (still) significantly outperform zero-shot generative ai models in text classification. *ArXiv*, abs/2406.08660, 2024. 1, 2
- [9] Xiang Chen, Chenxi Wang, Yida Xue, Ningyu Zhang, Xiaoyan Yang, Qiang Li, Yue Shen, Jinjie Gu, and Huajun Chen. Unified hallucination detection for multimodal large language models. *arXiv preprint arXiv:2402.03190*, 2024. 4, 3
- [10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5
- [11] Xiang Deng, Vasilisa Bashlovkina, Feng Han, Simon Baumgartner, and Michael Bendersky. Llm to the moon? reddit market sentiment analysis with large language models. *Companion Proceedings of the ACM Web Conference 2023*, 2023. 2
- [12] Karan Desai and Justin Johnson. Virtex: Learning visual representations from textual annotations. In *2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 11162–11173, 2021. 2
- [13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805, 2019. 1, 2
- [14] Sivan Doveh, Shaked Perek, M Jehanzeb Mirza, Wei Lin, Amit Alfassy, Assaf Arbelle, Shimon Ullman, and Leonid Karlinsky. Towards multimodal in-context learning for vision & language models. *ArXiv*, abs/2403.12736, 2024. 6
- [15] Alaaeldin El-Nouby, Natalia Neverova, Ivan Laptev, and Herv’e J’egou. Training vision transformers for image retrieval. *ArXiv*, abs/2102.05644, 2021. 1, 2
- [16] Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. *ArXiv*, abs/2402.01306, 2024. 2
- [17] Evelina Fedorenko, Michael K. Behr, and Nancy Kanwisher. Functional specificity for high-level linguistic processing in the human brain. *Proceedings of the National Academy of Sciences*, 108(39):16428–16433, 2011. 2
- [18] Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive. *arXiv preprint arXiv:2404.12390*, 2024. 2, 4, 3
- [19] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. *ArXiv*, abs/2104.08821, 2021. 2
- [20] Danna Gurari, Qing Li, Abigale Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. Vizwiz grand challenge: Answering visual questions from blind people. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3608–3617, 2018. 5
- [21] Donghoon Han, Eunhwan Park, Gisang Lee, Adam Lee, and Nojun Kwak. Merlin: Multimodal embedding refinement via llm-based iterative navigation for text-video retrieval-rerank pipeline. *arXiv preprint arXiv:2407.12508*, 2024. 2
- [22] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019. 5
- [23] Roei Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. *ArXiv*, abs/2310.15916, 2023. 2
- [24] Roei Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. *arXiv preprint arXiv:2310.15916*, 2023. 2
- [25] Roei Herzig, Alon Mendelson, Leonid Karlinsky, Assaf Arbelle, Rogerio Feris, Trevor Darrell, and Amir Globerson.

- Incorporating structured representations into pretrained vision & language models using scene graphs. In *2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023. 2
- [26] L. Hinton G, van der Maaten. Visualizing data using t-sne. *Journal of machine learning research*, 2008. 6, 7
- [27] Alberto Hojel, Yutong Bai, Trevor Darrell, Amir Globerson, and Amir Bar. Finding visual task vectors. In *European Conference on Computer Vision (ECCV)*, pages 257–273. Springer, 2025. 2
- [28] Cheng-Yu Hsieh, Jieyu Zhang, Zixian Ma, Aniruddha Kembhavi, and Ranjay Krishna. Sugarcrepe: Fixing hackable benchmarks for vision-language compositionality. *ArXiv*, abs/2306.14610, 2023. 5, 8
- [29] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 6
- [30] Brandon Huang, Chancharik Mitra, Assaf Arbelle, Leonid Karlinsky, Trevor Darrell, and Roei Herzig. Multimodal task vectors enable many-shot multimodal in-context learning. *arXiv preprint arXiv:2406.15334*, 2024. 2, 6, 3
- [31] Weiquan Huang, Aoqi Wu, Yifan Yang, Xufang Luo, Yuqing Yang, Liang Hu, Qi Dai, Xiyang Dai, Dongdong Chen, Chong Luo, et al. Llm2clip: Powerful language model unlock richer visual representation. *arXiv preprint arXiv:2411.04997*, 2024. 2
- [32] Drew A. Hudson and Christopher D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6693–6702, 2019. 1
- [33] Achiya Jerbi, Roei Herzig, Jonathan Berant, Gal Chechik, and Amir Globerson. Learning object detection from captions via textual scene attributes. *ArXiv*, abs/2009.14558, 2020. 2
- [34] Ting Jiang, Shaohan Huang, Zi qiang Zhang, Deqing Wang, Fuzhen Zhuang, Furu Wei, Haizhen Huang, Liangjie Zhang, and Qi Zhang. Promptbert: Improving bert sentence embeddings with prompts. In *Conference on Empirical Methods in Natural Language Processing*, 2022. 2
- [35] Ting Jiang, Shaohan Huang, Zhongzhi Luan, Deqing Wang, and Fuzhen Zhuang. Scaling sentence embeddings with large language models. *ArXiv*, abs/2307.16645, 2023. 2
- [36] Ting Jiang, Minghui Song, Zihan Zhang, Haizhen Huang, Weiwei Deng, Feng Sun, Qi Zhang, Deqing Wang, and Fuzhen Zhuang. E5-v: Universal embeddings with multimodal large language models. *ArXiv*, abs/2407.12580, 2024. 2
- [37] Nancy Kanwisher. Domain specificity in face perception. *Nature neuroscience*, 3(8):759–763, 2000. 2
- [38] Sahar Kazemzadeh, Vicente Ordonez, Marc andre Matten, and Tamara L. Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014. 1
- [39] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013. 2
- [40] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11513–11522, 2022. 2
- [41] Yibin Lei, Di Wu, Tianyi Zhou, Tao Shen, Yu Cao, Chongyang Tao, and Andrew Yates. Meta-task prompting elicits embeddings from large language models. In *Annual Meeting of the Association for Computational Linguistics*, 2024. 2
- [42] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021. 2
- [43] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *ArXiv*, abs/2307.16125, 2023. 1
- [44] Baiqi Li, Zhiqiu Lin, Wenxuan Peng, Jean de Dieu Nyandwi, Daniel Jiang, Zixian Ma, Simran Khanuja, Ranjay Krishna, Graham Neubig, and Deva Ramanan. Natural-bench: Evaluating vision-language models on natural adversarial samples. *arXiv preprint arXiv:2410.14669*, 2024. 2, 4, 5
- [45] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *ArXiv*, abs/2408.03326, 2024. 4
- [46] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv preprint arXiv:2201.12086*, 2022. 2
- [47] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023. 2
- [48] Shiyu Li, Yang Tang, Shizhe Chen, and Xi Chen. Conan-embedding: General text embedding with more and better negative samples. *ArXiv*, abs/2408.15710, 2024. 2
- [49] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, abs/2101.00190, 2021. 2
- [50] Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. Scaling language-image pre-training via masking. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23390–23400, 2022. 2
- [51] Ziyue Li and Tianyi Zhou. Your mixture-of-experts llm is secretly an embedding model for free. *arXiv preprint arXiv:2410.10814*, 2024. 2
- [52] Ziyue Li and Tianyi Zhou. Your mixture-of-experts llm is secretly an embedding model for free. *arXiv preprint arXiv:2410.10814*, 2024. 2
- [53] Jiacheng Lin, Kun Qian, Haoyu Han, Nurendra Choudhary, Tianxin Wei, Zhongruo Wang, Sahika Genc, Edward W

- Huang, Sheng Wang, Karthik Subbian, et al. Unleashing the power of llms as multi-modal encoders for text and graph-structured data. *arXiv preprint arXiv:2410.11235*, 2024. 2
- [54] Tsung-Yi Lin, M. Maire, Serge J. Belongie, James Hays, P. Perona, D. Ramanan, Piotr Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014. 1, 3
- [55] Zhiqiu Lin, Xinyue Chen, Deepak Pathak, Pengchuan Zhang, and Deva Ramanan. Revisiting the role of language priors in vision-language models. *arXiv preprint arXiv:2306.01879*, 2023. 2
- [56] Zhiqiu Lin, Samuel Yu, Zhiyi Kuang, Deepak Pathak, and Deva Ramanan. Multimodality helps unimodality: Cross-modal few-shot learning with multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19325–19337, 2023. 8
- [57] Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. In *European Conference on Computer Vision*, pages 366–384. Springer, 2024. 6
- [58] Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. In *European Conference on Computer Vision*, pages 366–384. Springer, 2025. 2
- [59] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023. 1
- [60] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 1, 3
- [61] Tian Yu Liu, Matthew Trager, Alessandro Achille, Pramuditha Perera, Luca Zancato, and Stefan O Soatto. Meaning representations from trajectories in autoregressive models. *ArXiv*, abs/2310.18348, 2023. 2
- [62] Romain Lopez, Pierre Boyeau, Nir Yosef, Michael Jordan, and Jeffrey Regier. Decision-making with auto-encoding variational bayes. *Advances in Neural Information Processing Systems*, 2020. 2
- [63] Romain Lopez, Pierre Boyeau, Nir Yosef, Michael Jordan, and Jeffrey Regier. Decision-making with auto-encoding variational bayes. *Advances in Neural Information Processing Systems*, 33:5081–5092, 2020. 2
- [64] Shiyin Lu, Yang Li, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, and Han-Jia Ye. Ovis: Structural embedding alignment for multimodal large language model. *ArXiv*, abs/2405.20797, 2024. 2
- [65] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana-Maria Camburu, Alan Loddon Yuille, and Kevin P. Murphy. Generation and comprehension of unambiguous object descriptions. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11–20, 2015. 1
- [66] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013. 2
- [67] Chancharik Mitra, Brandon Huang, Trevor Darrell, and Roei Herzig. Compositional chain of thought prompting for large multimodal models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2
- [68] Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. Mteb: Massive text embedding benchmark. In *Conference of the European Chapter of the Association for Computational Linguistics*, 2022. 2
- [69] Jianmo Ni, Gustavo Hernández Abrego, Noah Constant, Ji Ma, Keith B. Hall, Daniel Matthew Cer, and Yinfei Yang. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. *ArXiv*, abs/2108.08877, 2021. 2
- [70] Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. Large dual encoders are generalizable retrievers. *ArXiv*, abs/2112.07899, 2021. 2
- [71] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729, 2008. 5
- [72] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022. 2
- [73] OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023. 1, 5, 8, 3
- [74] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Francis Christiano, Jan Leike, and Ryan J. Lowe. Training language models to follow instructions with human feedback. *ArXiv*, abs/2203.02155, 2022. 2
- [75] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *IEEE/CVF conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012. 5, 6
- [76] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 4, 3
- [77] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 2
- [78] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763. PMLR, 2021. 2
- [79] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a

- reward model. *Advances in Neural Information Processing Systems*, 36, 2024. 2
- [80] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *ArXiv*, abs/2204.06125, 2022. 2
- [81] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019. 1, 2
- [82] David E Rumelhart, James L McClelland, PDP Research Group, et al. *Parallel distributed processing, volume 1: Explorations in the microstructure of cognition: Foundations*. The MIT press, 1986. 2
- [83] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015. 1, 2
- [84] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014. 2
- [85] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8309–8318, 2019. 3
- [86] Aaditya K. Singh, Stephanie C.Y. Chan, Ted Moskovitz, Erin Grant, Andrew M. Saxe, and Felix Hill. The transient nature of emergent in-context learning in transformers. In *Advances in neural information processing systems (NeurIPS)*, 2023. 6
- [87] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2013. 2
- [88] Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. Text classification via large language models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023. 2
- [89] Gemini Team. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *ArXiv*, abs/2403.05530, 2024. 8
- [90] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 8
- [91] Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5238–5248, 2022. 8
- [92] Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. Function vectors in large language models. *ArXiv*, abs/2310.15213, 2023. 2
- [93] Eric Todd, Millicent L Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. Function vectors in large language models. *arXiv preprint arXiv:2310.15213*, 2023. 2
- [94] Matthew A. Turk and Alex Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3:71–86, 1991. 1, 2
- [95] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. *ArXiv*, abs/1711.00937, 2017. 2
- [96] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge J. Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 5
- [97] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Improving text embeddings with large language models. *ArXiv*, abs/2401.00368, 2023. 2
- [98] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Ke-Yang Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *ArXiv*, abs/2409.12191, 2024. 4
- [99] Pengyu Wang, Dong Zhang, Linyang Li, Chenkun Tan, Xinghao Wang, Ke Ren, Botian Jiang, and Xipeng Qiu. Inferaligner: Inference-time alignment for harmlessness through cross-model guidance. *arXiv preprint arXiv:2401.11206*, 2024. 2
- [100] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024. 3
- [101] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Jilan Xu, Zun Wang, Yansong Shi, Tianxiang Jiang, Songze Li, Hongjie Zhang, Yifei Huang, Yu Qiao, Yali Wang, and Limin Wang. Internvideo2: Scaling video foundation models for multimodal video understanding. *ArXiv*, abs/2403.15377, 2024. 2
- [102] Zhiqiang Wang, Yiran Pang, and Yanbin Lin. Large language models are zero-shot text classifiers. *ArXiv*, abs/2312.01044, 2023. 2
- [103] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. *ArXiv*, abs/2109.01652, 2021. 2
- [104] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Huai hsin Chi, F. Xia, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903, 2022. 2
- [105] Dean Wyatte, Fatemeh Tahmasbi, Ming Li, and Thomas Markovich. Scaling laws for discriminative classification in large language models. *ArXiv*, abs/2405.15765, 2024. 2
- [106] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yi Zhou, Junyan Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, Chenliang Li, Yuanhong Xu, Hehong Chen, Junfeng

- Tian, Qiang Qi, Ji Zhang, and Feiyan Huang. mplug-owl: Modularization empowers large language models with multimodality. *ArXiv*, abs/2304.14178, 2023. 3
- [107] Peter Young, Alice Lai, Micah Hodosh, and J. Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014. 1
- [108] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11941–11952, 2023. 2
- [109] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023. 2
- [110] Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D. Manning, and C. Langlotz. Contrastive learning of medical visual representations from paired images and text. In *Machine Learning in Health Care*, 2020. 2
- [111] Yuhui Zhang, Alyssa Unell, Xiaohan Wang, Dhruba Ghosh, Yuchang Su, Ludwig Schmidt, and Serena Yeung-Levy. Why are visually-grounded language models bad at image classification? *ArXiv*, abs/2405.18415, 2024. 1, 2, 5
- [112] Haozhe Zhao, Zefan Cai, Shuzheng Si, Xiaojian Ma, Kaikai An, Liang Chen, Zixuan Liu, Sheng Wang, Wenjuan Han, and Baobao Chang. Mmicl: Empowering vision-language model with multi-modal in-context learning. *ArXiv*, abs/2309.07915, 2023. 2
- [113] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 3
- [114] Yongshuo Zong, Ondrej Bohdal, Tingyang Yu, Yongxin Yang, and Timothy Hospedales. Safety fine-tuning at (almost) no cost: A baseline for vision large language models. *arXiv preprint arXiv:2402.02207*, 2024. 2, 4